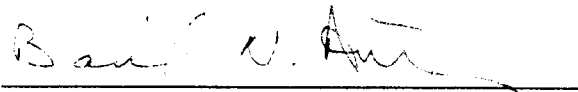


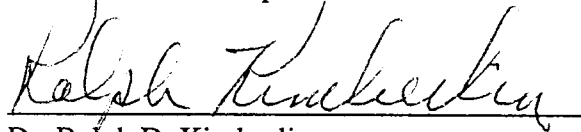
To the Graduate Council:

I am submitting herewith a thesis written by Travis M. Allen entitled "A Method for Optimizing a Helicopter Gun System Lead Computing Optical Sight Through Simulation." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Masters of Science, with a major in Aviation Systems.



Dr. Basil N. Antar, Major Professor

We have read this thesis and
recommend its acceptance:



Dr. Ralph D. Kimberlin



Dr. C. Ted Paludan

Accepted for the Council:



Associate Vice Chancellor
and Dean of The Graduate School

A METHOD FOR OPTIMIZING A
HELICOPTER GUN SYSTEM
LEAD COMPUTING OPTICAL SIGHT
THROUGH SIMULATION

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Travis M. Allen

August 1996

DISCLAIMER

The results contained within this thesis were developed at the Naval Air Warfare Center, Aircraft Division, Patuxent River, Maryland. The discussion of the data, conclusions and recommendations presented are the opinion of the author and should not be construed as an official position of the United States Department of Defense, the Naval Air Systems Command or the Naval Air Warfare Center.

DEDICATION

This thesis is dedicated to my wife, Elizabeth,
whose devotion and support made this project possible,
and which have enabled me to accomplish
much more than I could have done on my own;
to my parents who gave me so much of themselves.

ACKNOWLEDGMENTS

The author would like to acknowledge his brother, Timothy C. Allen, for his assistance and encouragement during the coding of the computer simulation programs.

ABSTRACT

This research explores the use of simulation to optimize a lead computing sight for a helicopter gun system. Basic research into air-to-air gunnery, air-to-air combat, and helicopter gun accuracy improvement programs involving lead computing sights was accomplished as well as a review of the existing F-18 "Hornet" lead computing sight system. A computer simulation was developed to provide a test bed for obtaining tracking accuracy data for a gun tracking task using three different pilot interfaces. Volunteer operators, both pilots and non-pilots, "flew" the simulation and real time data were recorded. The data were analyzed and used to develop tracking accuracy models which were in turn used in the development of a computer ballistics program using Monte Carlo simulation techniques. This limited scope simulation was used to evaluate the effects of the various kinds of pilot interfaces on the accuracy of the modeled gun direction system in combination with various control feedback loops. Probability of hit (P_H) data were generated which was utilized as a basis for a discussion of sight design and implementation. The method was extrapolated to use more capable aircraft specific high-fidelity flight simulators as data gathering devices. Finally, a recommendation was made for the mechanization of an optimized helicopter air-to-air gun and lead computing optical sighting system for the AH-1W "SuperCobra".

TABLE OF CONTENTS

CHAPTER	PAGE
1. INTRODUCTION AND PURPOSE	1
2. BACKGROUND	4
2.1. ARMING ATTACK HELICOPTERS FOR AIR-TO-AIR COMBAT	8
2.2. EFFECTIVENESS OF CURRENT ATTACK HELICOPTER GUN SYSTEMS	11
2.3. A HISTORICAL PERSPECTIVE	15
2.4. THE AH-64 INTEGRATED AAW (INTAAW) PROJECT	20
2.5. SUMMARY	23
3. CONFIGURATIONS FOR AN IMPROVED AH-1W "SUPERCOBRA" AIR-TO-AIR GUN SYSTEM	25
3.1. CURRENT AH-1W GUN SYSTEM IMPLEMENTATION	26
3.2. LEAD COMPUTING OPTICAL SIGHTS	37
3.3. PROPOSED LCOS AH-1W CONFIGURATIONS	43
4. GUN TRACKING SIMULATION	46
4.1. PREVIOUS SIMULATION EFFORTS	47
4.2. DESIGN GOALS	49
4.3. SOFTWARE FAMILIARIZATION AND PROGRAM BASICS	51
4.4. FLIGHT AND TARGET MOTION MODELS	57

4.5.	BALLISTICS COMPENSATION MODEL	60
4.6.	DATA RECORDING	64
4.7.	DATA COLLECTION	64
4.8.	DATA REDUCTION	66
4.9.	DATA ANALYSIS AND MODEL DEVELOPMENT FOR THE BALLISTICS SIMULATION	67
5.	BALLISTICS SIMULATION	77
5.1.	DATA ANALYSIS AND RESULTS	81
6.	CONCLUSIONS	90
6.1	EXTRAPOLATION OF THE METHOD	91
	BIBLIOGRAPHY	93
	REFERENCES CITED	95
	APPENDICES	
	APPENDIX A	100
	APPENDIX B	116
	APPENDIX C	132
	APPENDIX D	134
	APPENDIX E	181
	APPENDIX F	203
	VITA	205

LIST OF TABLES

TABLE	PAGE
I. Historical kill ratios: Fixed-wing vs. Helicopter	5
II. Air-to-air characteristics of helicopter and fixed-wing aircraft	7
III. AH-1W gun sighting system advantages and disadvantages	30
IV. Data Elements Recorded by the Gun Tracking Simulator	65
V. Raw Tracking Accuracy Score Data	68
VI. Statistical Description of Important Flight Parameter Data	76
VII. Ballistics Simulation Program Models	79
VIII. Comparison of Computed P_H Values for Three Simulated Gun Systems	82
IX. Effects of Target Size on P_H Values	87
X. Effect of Deleting Feedback Loops	89

LIST OF FIGURES

FIGURE	PAGE
1. Weapon Hierarchy for Engagement for Fixed and Rotary-Wing Aircraft . . .	10
2. Reticle Positioning for a 90° Deflection Shot on a 450 Knot Target at One Projectile Time-of-Flight (3410 Feet)	14
3. World War One Ring and Bead Sight Compared to AH-1W "SuperCobra" HUD Gun Sight Reticle	17
4. AH-1W M-197 Gun Installation	28
5. AH-1W M-65 Telescopic Sight Unit (TSU) and Front Cockpit Location . . .	31
6. AH-1W Heads-up Display (HUD) and Rear Cockpit Location	33
7. AH-1W Helmet Sight System (HSS) Components and Sighting Reticle . . .	34
8. Gun Direction System Schematic	36
9. Sample Fire Control Iterative Solution	39
10. Illustration of Fixed Reticle-Gun Compensation Model.	52
11. Illustration of CCIP Reticle-Fixed Gun Model	56
12. Graphical Comparison of Total Aiming Misalignment When Tracking a Moving Target and a Stationary Target	70
13. Controller Motion Time History	71
14. Aiming Misalignment versus Range to Target	73
15. Dispersion Diagram	84

16.	Gun Firing Geometry	121
17.	Effective Gun Firing Envelope	124
18.	Lead and Lag Pursuit Curves	126
19.	AH-1W Gun Reticle Diagram Showing Mil Sizing	129
20.	MiG-29 AH-1W HUD Rocket Reticle Depiction	131

LIST OF ABBREVIATIONS

AAA	Anti-Aircraft Artillery
AAW	Air-to-Air Warfare (or) Air-to-Air Weapon
APDS	Armor Piercing Discarding Sabot (ammunition)
AO	Angle Off the tail
ATGM	Anti-Tank Guided Missile
BRL	Ballistics Reference Line
CCIP	Constantly Computing Impact Point
CCRP	Constantly Computing Release Point
CCU	Coincidence Control Unit
CPG	Copilot/Gunner
DEP	Design Eye Point
DVO	Direct-View Optics
EOSS	Electro-Optical Simulation System
FFSP	Full Function Signal Processor
FLIR	Forward-Looking Infrared
FOV	Field Of View
GBL	Gun Bore Line
GDC	Guidance Development Center
GPS	Global Positioning System
HE	High Explosive

HMD	Helmet Mounted Display
HSS	Helmet Sight System
HUD	Heads-Up Display
IFCU	Interface Control Unit
IHADSS	Integrated Helmet And Display Sight System
INTAAW	Integrated Air-To-Air Weapons System
IRGM	Infrared Guided Missile
LCOS	Lead Computing Optical Sight
LOS	Line-Of-Sight
MAWTS-1	Marine Aviation Weapons and Tactics Squadron One
MCAS	Marine Corps Air Station
MFD	Multi-Function Display
MICOM	US Army Missile Command
NTS	Night Targeting System
NOE	Nap Of the Earth
NVG	Night Vision Goggle(s)
P_H	Probability of hit
P_K	Probability of kill
RADGUNS 1.9	Radar Directed Gun System Simulation v.1.9
SA	Situational Awareness
SAPHEI	Semi-Armor Piercing High Explosive Incendiary
TAA	Target Aspect Angle

TADS	Target Acquisition Designation Sight
TCA	Turret Control Assembly
TOF	Time-of-flight
TP-T	Target Practice - Tracer
TSU	Telescopic Sight Unit
TOW	Tube-launched Optically-tracked Wired-guided (Missile)
TVC	Television Camera
US	United States
USMC	United States Marine Corps
WDSS	Weapons Delivery System Simulation
WTI	Weapons and Tactics Instructor
WWI	World War One

CHAPTER 1

INTRODUCTION AND PURPOSE

An excellent weapon and luck had been on my side. To be successful the best fighter pilot needs both.

Lt. General Adolph Galland
General of the Fighter Arm, Luftwaffe, 1941-45
104 Victories. (Spick 1991)

Air-to-air combat is at once a simple and a complex thing. It is simple because it embodies the single-minded objective of destroying one's enemy. Whether in a one-on-one engagement or a mass of contesting aircraft formations, the goal is to single out an enemy aircraft and destroy it. It is complex because achieving this objective involves difficult physical realities such as aircraft energy states, three-dimensional relative motion, various aircraft and weapons systems characteristics, and projectile ballistics; as well as more ephemeral factors such as the pilot's situational awareness, mental and visual acuity; physical stamina, tactical proficiency, aeronautical skills, and the natural desire not to be killed in the process of destroying the enemy.

In the 50 years or so since air-to-air combat became a recognized part of war-making, much effort has been expended in maximizing the pilot's chance of success, both through strict pilot selection and training, and through development of increasingly superior aircraft and weapons systems. To achieve the highest probability of success, the capabilities of the

pilot and his aircraft must be complimentary and the weakness of one needs to be balanced by the strengths of the other.

Pilots are good at things that machines have not yet been able to master. Accurate identification of enemy aircraft, assessing relative strengths and weaknesses between one's own forces and the enemy's, aircraft control during aggressive maneuvering, application of tactical principles, and maintaining situational awareness in a rapidly changing fight are problems that a talented pilot can solve almost instantaneously, seemingly without effort.

On the other hand, machines outstrip pilots in more analytical pursuits; accurately estimating the range and range rate of a target, computing bullet drop for a given range, measuring and compensating for own aircraft state (pitch and roll attitude and aircraft sideslip) and rates (pitch, roll, yaw rate, and three-dimensional displacement), determining relative target motion, and calculating an appropriate lead and elevation sighting angle. Additionally, constitutional assurances notwithstanding, all pilots are not created equal. A few are aerial demigods who spring forth from the womb (or flight school at least), ready to enter the pantheon of Aces, while the majority are merely competent and glad of the "edge" that superior technology can provide.

Therefore, providing an optimized air-to-air aircraft weapons system which takes into account the abilities of the average operator is an important design goal for anyone involved in the development of the machine portion of the man/machine fighting unit. This thesis presents a method for optimizing such a system.

The first part of this report provides background information and reviews the basic theory of aerial gunnery as it applies to air-to-air combat with specific application for an

attack helicopter equipped with a forward firing or turreted cannon. Historical records of fixed-wing aircraft with similar flight performance are also reviewed to provide a grounding for general conclusions concerning the requirements for air-to-air gunnery from a rotary-wing platform. A discussion of the flight test of a prototype lead computing sighting system for the AH-64A helicopter is provided. The second part discusses possible sight and weapons systems configurations for a specific aircraft, the AH-1W "SuperCobra" and provides basic information on lead computing optical sight systems. The third part details the development of a computer based gun tracking simulator, its use in data collection to develop generalized tracking accuracy models, a discussion of the reduction of the data taken, and the results of that analysis. The fourth part of the report details the development of a simplified computer based ballistics simulation based on Monte Carlo modeling techniques which incorporates the previously derived tracking accuracy models, a discussion of its use in developing probability of hit (P_H) data, an analysis of this P_H data, and the development of specific conclusions based upon both simulations. Finally, the report describes a proposal for the use of this method incorporating more capable aircraft specific flight simulators as well a final proposal for an optimized helicopter air-to-air gun and lead computing optical sighting system for the AH-1W "SuperCobra".

CHAPTER 2

BACKGROUND

Armed helicopters have come to play a larger and larger role on the battlefield in the years since the Vietnam War. They have increased in number as well as in capability and now make up the majority of the close-air support and attack aviation assets available to the ground commander. Their increased presence on the battlefield, often times on both sides of a conflict, has greatly increased the probability of air-to-air engagements between helicopters and between helicopters and fixed-wing aircraft.

History provides numerous instances of helicopters being involved in air-to-air engagements with both fixed and rotary-wing aircraft. When an armed fixed-wing aircraft meets a rotary-wing aircraft (particularly an unarmed helicopter), it usually spells trouble for the helicopter pilot. Table I, adapted from Barnes (1991b) records the historical kill ratios between fixed-wing aircraft and helicopters. There is little to surprise the reader in the lopsided outcome, as most of the helicopters downed were transport aircraft or lightly armed scouts and utility craft. The five fixed-wing aircraft downed by helicopters provide a more interesting tale.

Three of the fixed-wing aircraft were brought down by gunfire. Both of the kills during the Vietnam war were achieved during the same action when an Air America UH-1 "Huey" intercepted three AN-2 "Colt" light-transports and shot down two utilizing crew-

Table I: Historical kill ratios: Fixed-wing vs. Helicopter.

War	Fighter Kills	Helicopter Kills	Kill Ratios
Vietnam War, 1965	3	2	1.5:1
Six Day War, 1967	1	0	1:0
Indo-Pakistani War, 1971	2	0	2:0
Yom Kippur War, 1973	51	0	51:0
Arab-Israeli Attrition, 1973-1982	2	0	2:0
Rhodesian-Botswana, 1979	0	1	0:1
Iran-Iraq Gulf War, 1980	133	2	66.5:1
Falkland War, 1982	4	0	4:0
Peace For Galilee, 1982	5	0	5:0
Angola, 1984	18	0	18:0
Contra War, 1984	1	0	1:0
Romanian Revolution, 1989	4	0	4:0
Kuwait-Iraq, 1990	1	0	1:0
The Gulf War, 1991	6	0	6:0
Totals	231	5	46.2:1

Source: Barnes (1991).

served door guns. Both the Iraqis and the Iranians claimed a jet fighter each during the Iran-Iraq War, through action by their respective attack helicopter forces. Perhaps most germane to the subject of this report, a Rhodesian SA-316B "Alouette III" downed a Botswanan Britten-Norman Defender (a civil transport twin turboprop aircraft outfitted with weapons stations and systems) from a hover with 20mm cannon fire. (Barnes 1991b).

Helicopter versus helicopter aerial combat is also well documented. Perhaps the most significant conflict to date as regards helicopter air-to-air combat was the Iran-Iraq war which was fought in the early 1980's. Radio communiques translated by the US Foreign Broadcast Information Service record 174 air-to-air engagements involving helicopters during the course of the conflict. Of these, 56 were helicopter-versus-helicopter engagements, and ten involved direct combat between Iranian attack helicopters (AH-1J "TOW Cobra") and Iraqi attack helicopters (Mi-24 "Hind-D"). In the final tally, ten "Hind-D" and six "Cobra" helicopters were downed in combat between the two helicopter types. Numerous other rotary-wing platforms were destroyed by Iranian and Iraqi attack helicopters, including the downing of six Iraqi transport helicopters by Iranian "Cobras" in one engagement in February of 1984. (Barnes 1991a).

A much larger record exists for fixed-wing air-to-air combat than for helicopters. Some elements of the air-to-air fight are transferable between these platforms (as will be discussed later), but a summary of some of the differences can be instructive. Table II, reproduced from Buresh , Parlier, and Wilson (1988), provides a comparison.

Table II: Air-to-air characteristics of helicopter and fixed-wing aircraft.

Task	Fixed-wing - High Altitude	Helicopter - Nap of the Earth (NOE)
Detection	•Occurs at long ranges. •Environment uncluttered. •Speed does not alter detectability. •Detection ranges are significantly greater than weapons ranges.	•Occurs at short ranges. •Environment very cluttered, EO/radar may not be useable. •Speed increase altitude, significantly increasing exposure. •Weapons ranges are greater than detection ranges.
Acquisition	•Maneuvers involved speed to jockey for firing position. •Speed dictates long turn times. •Weapons have limited azimuth capability, dictating aircraft pointing for weapons launch. •Turn rates of 15°/sec to 20 °/sec are maximum	•Given comparable weapons, the first one to shoot will win. •Rapid turns and quick shots will dominate. •Turret guns permit off-axis short range engagement. •Turn rates of 40°/sec to 60 °/sec are maximum
Tactical Maneuvering	•Turn radii of ½ mile are minimum. •Considerable vertical maneuvering is desirable.	•Turn radii of 100 feet are common, hover turns will be important. •Vertical maneuvering above NOE may not be permitted due to anti-aircraft threats.

Source: Buresh, Parlier, and Wilson (1988).

ARMING ATTACK HELICOPTERS FOR AIR-TO-AIR COMBAT

As helicopter air-to-air warfare moved from a theoretical possibility to a reality, efforts were made to rapidly integrate various air-to-air systems on armed helicopters and to develop tactics for the use of existing weapons systems for air-to-air engagements. The AIM-9 series "Sidewinder" infra-red guided missile (IRGM) is currently approved for use on the US Marine Corps AH-1W "SuperCobra" attack helicopter and has been fired from the Army's AH-64A "Apache". The "Stinger" man-portable air defense IRGM has been modified for use on various US Army helicopters (AH-64A, OH-58 "Kiowa") and the entire family of Russian attack helicopters (Mi-24/24/35 "Hind", Mi-28 "Havoc", Ka-50 "Hokum") have been modified or designed from the ground up with provisions for air-to-air IRGM's. (Lambert 1995).

Extensive testing has been performed on the use of existing gun and missile systems for air-to-air combat. Iraqi forces claimed an air-to-air kill of an Iranian F-4 "Phantom" by an Mi-24 "Hind-D" with an AT-2 "Swatter" anti-tank guided missile (ATGM). (Barnes 1991a). The AGM-114 "Hellfire" laser guided missile has shown great promise for air-to-air engagements, and in fact, in a recent mishap a Marine Corps F-18 "Hornet" was struck and nearly destroyed by a Hellfire missile fired from an AH-1W. The Hellfire's seeker head mistook the Hornet's laser designator for the target and followed the designation beam to an unfortunate end. (Anderson 1996). To a lesser extent, the BGM-71 series TOW ATGM is also a viable candidate for air-to-air work, especially against low-performance aircraft and helicopters. (DON 1993a).

In the close-in air-to-air realm which is typified by aggressive maneuvering and limited tracking time, the primary weapon for air-to-air engagements is considered to be the rapid firing aircraft mounted cannon. Aircraft cannons provide a rapid-firing, high-velocity, short time-of-flight explosive projectile which can bring down an enemy aircraft with a short burst. (Shaw 1985).

The various engagement ranges of these weapons has given rise to a weaponneering hierarchy shown in Figure 1 (adapted from DON (1993a)) in which a target is engaged with the longest range weapon commensurate with range of detection and identification. If all engagements were conducted at long range, it would appear that there would be little use for optimizing a helicopter's gun system for air-to-air engagements. However, in the low-level regime where the majority of helicopter flight operations take place, an enemy airplane or helicopter may not be detected until the target range is well inside minimum missile engagement range. Additionally, through the use of electronic and infra-red countermeasures, a missile's seeker head can be decoyed or "spoofed", rendering the longer range weapons useless. (Shaw 1985). The air-to-air combat regime that this report will primarily address will be the close-in maneuvering fight, though the other stages should be considered equally as important and worthy of further investigation.

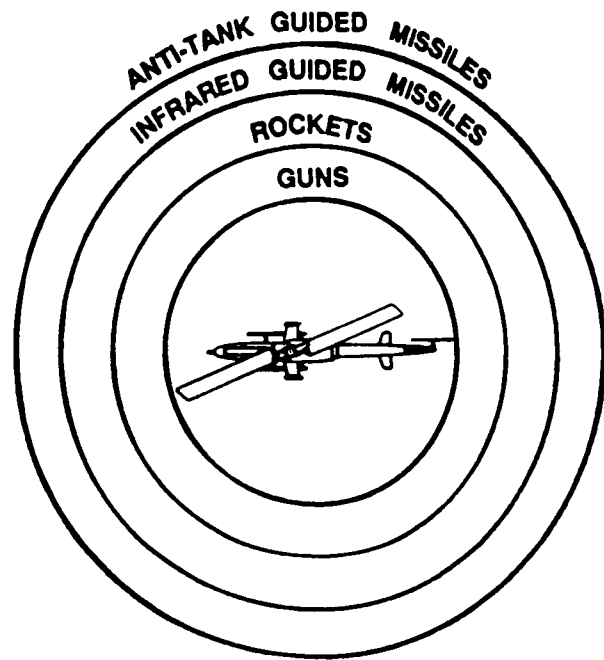
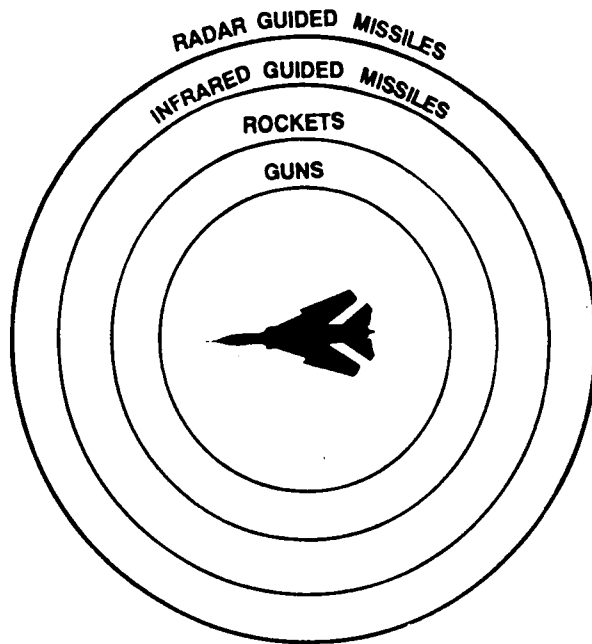


Figure 1: Weapon Hierarchy for Engagement for Fixed and Rotary-Wing Aircraft.

Adapted from: DON (1993a).

EFFECTIVENESS OF CURRENT ATTACK HELICOPTER GUN SYSTEMS

Clearly then, a robust gun capability is a requirement for an attack helicopter intended for use in air-to-air engagements. However, current gun systems are not very well optimized for air-to-ground work, let alone more demanding air-to-air engagements. As anecdotal evidence for the unsuitability of the current helicopter gun systems for air-to-air engagements, the experience of the author during the Spring 1990, Marine Corp's Weapons and Tactics Instructor (WTI) training course is provided.

The WTI course is an advanced training school where experienced pilots from all USMC aviation communities are brought together for extensive training in both combined operations and for "graduate level" training in their specific aircraft. The aim of the WTI course is to provide instructors to the Marine Corps operational squadrons, so that the WTI's can pass on the expertise and knowledge gained to the rest of their squadron.

During the air-to-air gunnery exercise for the attack helicopter students, four AH-1W helicopters loaded with 500 rounds of 20mm ammunition were arrayed to fire on a banner target towed by an A-4 "Skyhawk". The engagement was preplanned with the target range known and the lead and elevation angles computed before the flight. Range to target was limited to less than one projectile time-of-flight (TOF). Projectile TOF represents the distance that the projectile will travel over the referenced period of time, normally described in seconds or fractions of seconds (one TOF equals the distance a projectile will cover in one second). Range for air-to-air gun engagements are often referenced as a function of projectile TOF, and the current AH-1W gun system mechanization is considered ineffective for air-to-

air engagements out past one TOF. The tow speed varied between 190 and 220 knots and the tow aircraft flew a predictable straight line path essentially perpendicular to the helicopter's firing line. This target path admittedly maximized the target crossing rate and angle, but also maximized the target's aspect. Multiple tows were made to allow the maximum time for firing solution setup and to provide as many firing runs as the pilots of the AH-1W's desired. At the end of the exercise a grand total of 3 hits were recorded out of 2000 rounds fired.

There are many possible answers to the question as to why eight reasonably experienced "Cobra" pilots (including four instructor pilots) were only able to achieve a 0.15% hit rate. The firing rate of the AH-1W's cannon is only 650 rounds per minute, which will limit the number of rounds fired and the projectile stream density during an engagement. (DON 1993a). However, the number of engagements were essentially unlimited, and the flight path of the target was predictable and essentially unchanged from firing run to firing run. The 90° aspect angle provides the most difficult deflection shot, but the preplanned nature of the exercise and the unlimited number of engagements should have mitigated this to some extent. Therefore, the primary cause of this poor performance must lie elsewhere.

The AH-1W's sighting system for its 20mm cannon is little changed from the system that was developed during the Vietnam War for the earlier models of the AH-1 [see Appendix A for a general discussion of attack helicopter gun sighting systems]. The AH-1W has three sighting modes for its gun system. The most accurate is slaving the gun to the stabilized telescopic sight unit (TSU) used for guiding ATGM's, but the tight field-of-view (FOV) and slow slew rates are ill-suited for a close-in maneuvering fight. The helmet sight system (HSS) which slaves the gun to the pilot's line-of-sight (LOS) would seem to be a

superb method for firing at off-axis targets, but not only does firing off-axis greatly increase the difficulties in accurately estimating an appropriate lead and elevation angle, but the natural slippage of the pilot's helmet during head movement and the lack of sight stabilization yields very inaccurate results. Finally, the rear cockpit heads-up display (HUD) provides a wide FOV, but the reticle representation, despite its high-tech appearance, is in reality no more sophisticated than the ring and bead sights used at the turn of the century.

All of these firing modes have the additional disadvantage of requiring the pilot to mentally compute lead and elevation angles based on estimated range to the target, estimated target relative motion, and the pilot's perception of his own aircraft state and rates. The pilot must then manually offset the aiming reticle by an appropriate amount in the correct direction. (DON 1993a).

It is the author's contention that the reason the pilots fared so badly during this exercise was that they couldn't accurately aim to hit the target. This assertion may seem contradictory when one considers that the lead and elevation angles were computed prior to the exercise and were available to the pilots during the flight. However, it is one thing to know what the appropriate lead angle is, it is another to accurately make use of that information. Figure 2 is adapted from the AH-1 Tactical Manual (DON 1993a) and represents the relative positioning of the AH-1W HUD gun reticle and a 450 knot fixed-wing target for a 90° deflection shot with the target at a range of one bullet time-of-flight (TOF), or 3410 feet. As the reader can imagine, there is a high degree of difficulty in achieving correct reticle positioning for this firing solution due to the target size and the large reticle-to-target spacing. When the pilot is also faced with computing the lead and elevation angles "on

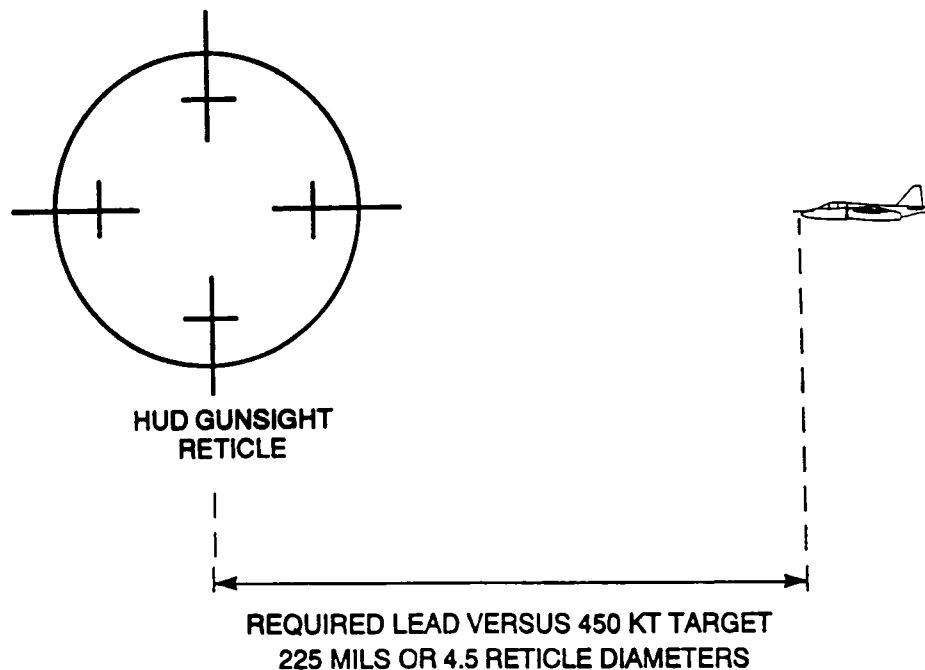


Figure 2: Reticle Positioning for a 90° Deflection Shot on a 450 Knot Target at One Projectile Time-of-Flight (2725 Feet).

Adapted from: DON (1993a).

Note: One projectile time-of-flight refers to the range the projectile will cover in one second of flight.

the fly” while simultaneously trying to maneuver his aircraft into firing parameters, obtaining even a few hits from a firing burst is merely a matter of luck.

A HISTORICAL PERSPECTIVE

The difficulty in accurately determining appropriate lead and elevation angles is not new to the AH-1W or the attack helicopter community in general. It is instructive to examine some of the problems experienced by scout and fighter pilots during World War I, when the first air-to-air combat engagements took place. This period of conflict is of particular interest to the attack helicopter pilot, because the performance of the aircraft used during this conflict most closely matches the forward-flight performance characteristics of the modern attack helicopter. The biplanes and triplanes flown by the Allied and German forces during the Great War operated in a speed range that varied between 60 and 110 knots as compared to the modern attack helicopter which operates in forward flight from 45 to 150 knots. Their power loading was higher than the attack helicopter (typically 12 to 20 lb/hp compared to 7 lb/hp for the AH-1W) which means that the attack helicopter would also have an advantage in climb performance and energy maintenance during maneuver, but the turn radii and altitudes operated in are similar. (Spick 1988).

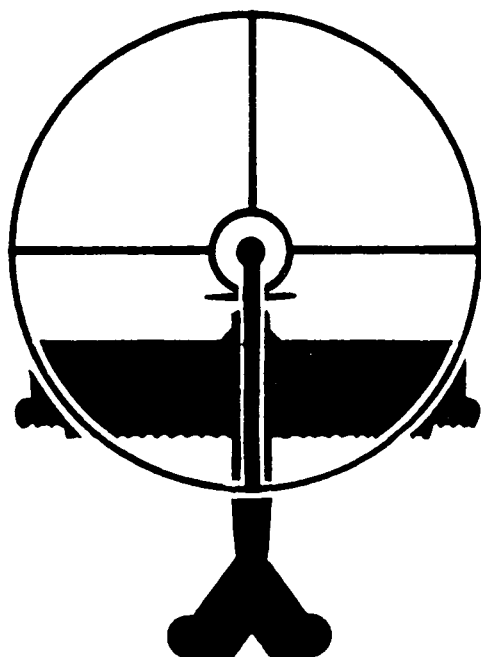
Additionally, the weaponry used during close-in engagements are similar. The AH-1W's 20mm cannon firing rate as previously noted is 650 round per minute, which compares with most of the forward firing machine guns used in World War I. The machine guns used on the more successful scout and fighter aircraft during this era required a propeller

synchronizing mechanism to allow the gun to be mounted on the aircraft fuselage in reaching distance of the pilot, and this mechanization reduced firing rates to 450 to 600 rounds per minute. (Spick 1988). Of course, the high-explosive 20mm projectile is much more destructive than the .30 cal bullets fired by most World War I machine guns, but then the unsophisticated aircraft structures of that time were considerably more delicate and easier to down than modern aircraft, so this difference is less significant.

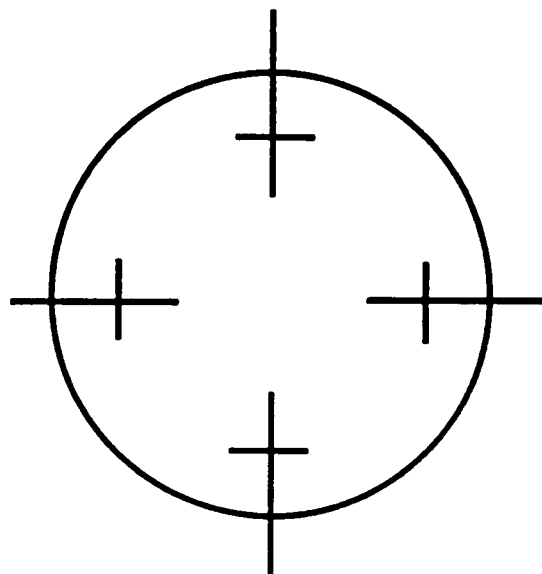
The modern attack helicopter's gun sighting systems have not changed much from those used in the First World War. The predominant sight was the ring and bead sight which was considered superior in a maneuvering fight, as it did not block out the pilot's field-of-view. The ring and bead sight was also used in estimating range to target by comparing the relative size of the target to the known size of the ring portion of the sight. Figure 3 presents a typical ring and bead sight display of the WWI era compared with the HUD gun reticle display from an AH-1W "SuperCobra". (Shaw 1985; DON 1993a). The telescopic sight (such as the Aldis sight) was marginally more effective at increasing engagement ranges when attacking a non-maneuvering pilot, but it was less than useful in a maneuvering melee, and was generally ignored when firing in the in-close fights, with the expected poor results. (Shaw 1985).

It may be illuminating to consider the gunnery accuracy performance of the pilots who flew these early fighter aircraft, and the factors that effected their results.

Leonard Rochford was a British pilot flying Sopwith Pups and Camels during WWI with No 3 Naval Squadron, later No 203 Squadron, RAF. He was credited with twelve



(a) Ring and Bead Sight



(b) AH-1W HUD Gun Reticle

Figure 3: World War One Ring and Bead Sight (a) Compared to AH-1W "SuperCobra" HUD Gun Sight Reticle (b).

Adapted from: (a) Shaw (1985).

(b) DON (1993a).

confirmed kills out of 74 engagements, during two tours on the Western Front. Spick (1988) describes Rochford's attendance at an air gunnery course at Berck sur Mer in February, 1918.

"Firing at a target floating in a pond, he scored seventeen hits from 200 rounds and fifteen hits from 120 rounds. He also fired 400 rounds at a drogue target towed by a BE 2c, scoring only seven inners and eighteen outers. This was about par for the course, and explains the poor results achieved [during combat] by the majority of pilots."

In fact, most pilots regularly sent large volumes of rounds downrange with little effect. This could partly be blamed on the intricacies of deflection shooting. From Spick (1988),

"The aircraft of the First World War were small by today's standards, and while a marksman on a ground range could expect to score a respectable number of hits on a similarly sized target at between 500 and 600 yards, using the machine-guns of the period, hitting a moving target from an unstable and vibrating aeroplane at less than half this distance was virtually impossible....For this reason crossing shots at high angles were rarely successful, even when used by the most gifted marksman, as the bullets tended to be spread along the length of the target, rather than grouped. The most reliable method was to get in close for a small or no-deflection shot."

This "stick your nose in the cockpit" approach typified most air-to-air engagements during WWI, and that indeed was how the highest scoring pilots racked up their impressive results. However, getting right on top of one's opponent was no guarantee of success. Once again from Spick (1988).

"Keith Caldwell, of No 60 and later commanding No 74 Squadrons, was one of the most aggressive men ever to strap an aeroplane onto his back. Although he survived the war with a tally of seventeen confirmed victories,

he was often seen hard on the tail of a German scout, firing for all he was worth at point-blank range without apparent effect, while Ball [British ace with 43 victories] himself recorded missing from a mere five yards on one occasion.”

The matter of range estimation was also significant [see Appendix B for a discussion of range estimation]. The ability to accurately estimate or measure range to target is crucial to success in air-to-air combat (for reasons to be discussed later) and most pilots are no good at range estimation by eye. From Spick (1988).

“Tracer was supposed to be an aid to aiming, and both sides used it extensively, but it could be very misleading. The burn time was variable in the extreme; sometimes it would extinguish itself quite quickly. If this happened before reaching the target, it would appear that the bullets were vanishing into the enemy aircraft, whereas in fact they were falling short. In a fast-moving manoeuvring combat, many pilots preferred to ignore their telescopic sight, and “hosepipe” tracer at their opponent. This was not very effective. The average pilot’s judgement of range was abysmal, and often when he thought he was firing from about 100 yards, the actual distance could be double, or even treble this. The high scorers tended to be good judges of distance.”

The limited duration of engagements and the short length of time available to track a target aircraft also worked against success in air-to-air contests. Once an enemy aircraft was detected (hopefully unawares), the longest tracking periods were found during the closing phase [see Appendix B for a discussion on the phases of air-to-air combat]. Once the battle was met and the element of surprise was lost, the maneuvering stage of the contest afforded good tracking opportunities for only a few seconds at a time. The pilot had to either be very good at estimating lead and elevation sighting angles, or he was forced to adopt the previously noted “hosepipe” or “snapshot” approach to firing [see Appendix B for a

discussion of air-to-air gunnery techniques]. This second method could be effective, but it had the disadvantage of "diluting" the attacker's firepower and dispersing the impact of his rounds. (Spick 1988).

This discussion of the problems encountered during the Great War of over 80 years ago shows that those of us involved in the development in helicopter gun systems in general and air-to-air gun systems in particular, have failed to learn from the mistakes of the past. The performance deficiencies of the pilots of WWI are still found in the pilots of today, despite stringent selection and vigorous training. Yet little serious work has been done to provide today's attack helicopter pilots with a system that complements their strengths and makes up for their natural weaknesses. One group that has made great efforts in this area however, is the McDonnell Douglas Helicopter Company.

THE AH-64 INTEGRATED AAW (INTAAW) PROJECT

McDonnell Douglas Helicopter Company has conducted several programs in recent years to provide an air-to-air capability for its attack helicopter product, the pre-eminent AH-64 "Apache". In 1991, McDonnell Douglas Helicopters and the US Army conducted a joint program to provide an improved air-to-air capability for the "Apache" gun system, as part of the Integrated Air-to-Air Weapons System (INTAAW) project. The test bed for the project was an AH-64A with a modified weapons processor (the aircraft's fire control computer). Aircraft state and rate information was fed to the weapons processor to stabilize the helicopter's 30mm chain gun. Range to the target was determined using the Target

Acquisition Designation Sight (TADS) laser rangefinder/designator, and target angular velocity was measured using the TADS angular drive velocity in elevation and azimuth. The TADS automatically tracked the target using the onboard contrast tracker in the television camera, or "TVC autotrack mode" after the system operator acquired the target and centered it in the sight's field-of-view.

The target platform was a remotely piloted large-scale model aircraft which was sixteen to eighteen feet long and which had a wing span of twenty feet. The modified "Apache" fired 30mm high explosive (HE) shells in short bursts of about 10 rounds at the maneuvering target aircraft. The division of labor in the cockpit was thus; the pilot flew the aircraft into release parameters, and the copilot/gunner (CPG) acquired the target aircraft with the TADS, activated the TVC autotrack mode, and initiated the firing burst. Numerous direct hits were recorded at distances out past 1000 meters. Furthermore, the improvements to the weapons processor also provided a significant increase in the air-to-ground accuracy of the "Apache" gun system. This very successful developmental program provided the basis for the improved "Longbow Apache" weapons processor algorithms. Interestingly, there are few publicly available reports describing the flight test portion of the INTAAW program and this account was obtained through an interview with Mr. Ed Wilson, the Senior Experimental Test Pilot at McDonnell Douglas Helicopters. (Wilson 1996).

The success of the INTAAW program was even more remarkable when one considers that the 30mm round fired by the "Apache" isn't particularly suited for air-to-air combat. Despite the significant destructive power of the 30mm projectile, which was developed for use against armored vehicles and main battle tanks, the round itself is relatively slow

compared to other rounds developed for air-to-air use (2598 ft/sec versus 3410 ft/sec for the PGU-28/B 20mm round). (Lennox 1995; Fulmer 1987). This slow speed causes an arcing trajectory which would ordinarily serve to limit the effective range of the 30mm in air-to-air combat to less than 1500 feet [see Appendix A for a more complete discussion of aerial gun ballistics], yet the INTAAW "Apache" was able to regularly hit a target three times smaller than the ones it is expected to face in combat, firing short bursts, at twice that distance. The superiority of the aircraft fire control system made up for the limitations of the basic gun round by relieving the pilots of the weapons pointing task. (Wilson 1996).

Unfortunately, due to the limited slew rates of turret mounted sighting systems and their narrow fields-of-view, these type sights are largely ineffective in the close-in maneuvering fight. For example, the AH-1 Tactical Manual advises that the crew using the TSU will have difficulty tracking targets with high crossing rates inside 1500 meters. (DON 1993a). Additionally, while at longer ranges the CPG might be able to operate the weapons system while the pilot stabilizes the aircraft in firing parameters, in the close-in fight the uncertain nature of maneuvering into a firing position for a quick shot requires that the pilot be the one who actually aims and fires the weapon. (Wilson 1989). The level of crew coordination required is analogous to asking the aircraft's crew to thread a needle with the pilot holding the needle and the CPG holding the thread; easily done if sitting on a park bench, significantly more difficult if riding in a roller coaster.

A similar air-to-air gun accuracy improvement program was undertaken in the late 1970's by the US Army and the Armament Systems Department of General Electric Corporation to develop an improved gun fire control system for the AH-1G/S "Cobra"

helicopter, though the author could find little information concerning the results of this effort. (GEC 1979).

SUMMARY

The difficulties inherent in obtaining an accurate aiming solution for an air-to-air engagement using an aircraft mounted gun should now be evident, both from a historical perspective and from the anecdotal description of more modern experience. We should now consider how an aircraft's weapons systems should be designed to maximize the pilot's chance for success.

First, we need to recognize why historic and modern pilots have such a problem achieving good results during air-to-air gun combat. Firing rates of the aircraft gun may be low, effect of the projectile on the target may be small, deflection shooting may be inherently difficult; but the crux of the matter is that if the system or the pilot can not accurately determine where to aim the aircraft weapon, then the percentage of hits and the concurrent chance for success will be unacceptably low. Yet, we have seen that a properly mechanized system, such as one for the INTAAW program, can yield spectacular results.

Some basic conclusions on systems requirements for the close-in maneuvering fight can be made. The system should make obtaining a firing solution as simple as possible. It shouldn't require excessive mental computation by the pilot at the same time he is expected to aggressively maneuver his aircraft. It should provide an interface that maximizes the pilot's awareness of what is happening around him (situational awareness), and it should

allow the pilot who is maneuvering the aircraft to be able to train and fire the weapon at the adversary.

These general goals can be implemented through a multitude of theoretical mechanizations. However, not all systems are readily adaptable to the small attack helicopter platform. Additionally, due to financial constraints, retrofitting an effective air-to-air system to an existing attack helicopter platform must utilize, as a matter of course, existing equipment and systems to the maximum extent possible. It is a truism that in large part, a system's mechanization defines the pilot's interface, and the modification of equipment intended for one use (area suppression) to another (precision air-to-air combat) will to a great extent dictate the capabilities of the final configuration. Therefore, it is instructive to select one attack helicopter and examine some possible configurations and modifications before we attempt to optimize a particular mechanization.

CHAPTER 3

CONFIGURATIONS FOR AN IMPROVED AH-1W "SUPERCOBRA" AIR-TO-AIR GUN SYSTEM

The AH-1W "SuperCobra" is the evolutionary product of a long-term modification and modernization program begun by its manufacturer, Bell Helicopter Textron Inc., during the Vietnam War. The original aircraft in the AH-1 "Cobra" family (the AH-1G) was derived from the airframe and dynamic components of the US Army's UH-1 series "Iroquois" utility helicopter first used in Vietnam. The AH-1G was developed to meet the requirement for an attack specific helicopter gunship, and was used by both the US Army and US Marine Corps in Vietnam. A Marine version of the AH-1G, the AH-1J "SeaCobra" was developed in 1970. The "SeaCobra" was equipped with a twin-engine propulsion package to increase survivability during operations at sea, and was retrofitted with Navy standard avionics and weapons control systems. The next generation of Marine AH-1's was the AH-1T, which was outfitted with a higher output engine package and improved main and tail rotor, and later with a precision anti-tank capability in the form of the TOW ATGM system. The AH-1W "SuperCobra" was designed to use the General Electric T-700 series engines with a commensurate increase in maximum gross weight to 14,750 lbs. The AH-1W's weapons systems were upgraded with a Kaiser Heads-up display (HUD), and with provisions for firing the AIM-9 Sidewinder IRGM, and the AGM-114 "Hellfire" ATGM. (O'Reilly 1987; Michell 1994).

The USMC has extensively explored the AH-1W's current air-to-air gun capability through the efforts of Marine Aviation Weapons and Tactics Squadron One (MAWTS-1) at MCAS Yuma, Arizona. Specific training flights to expose AH-1W pilots to air-to-air gunnery techniques are integrated into the Marine Corps' Training & Readiness syllabus for attack helicopter pilots, and graduate level exercises and training is conducted by MAWTS-1 during its WTI course discussed in Chapter 2. This level of effort is admirable, because it shows that the Marine Corps and the Marine attack helicopter community recognizes the importance of helicopter air-to-air warfare. The large body of training to date, however, has been complicated and less effective than it should have been, primarily due to the unsuitability of the AH-1W's gun control system for air-to-air combat.

CURRENT AH-1W GUN SYSTEM IMPLEMENTATION

M-197 Gun and A/A49E-7(V4) Turret Assembly

The M-197 20mm turreted gun has proven an effective area suppression, air-to-ground weapon since its initial installation on the AH-1J "SeaCobra". The M-197 gun is a derivative of the M61A1 six-barreled 20mm "Vulcan" cannon developed for use on helicopters, light fixed-wing aircraft, and small naval craft. The M-197 is a Gatling type cannon with three barrels capable of firing both linked and unlinked rounds when equipped with the appropriate ammunition feed assembly. The gun firing rate can be variably set from 600 rounds/minute to 3000 rounds per/minute with the appropriate drive and feeder motors installed. The M-197 fires standard 20 x 120 20mm ammunition such as the M50, PGU

28/30, and Vulcan Phalanx M-149 armor piercing discarding sabot (APDS) ammunition families. (DON 1993a; Lennox 1995).

As installed in the AH-1W, the M-197 gun is lightweight and provides a moderately high rate of fire for more than one minute's duration depending on the ammunition load carried (up to 750 rounds). The gun achieves its full firing rate of about 650 rounds/minute within 0.3 seconds. (DON 1993a; Lennox 1995). The M-197 gun is mounted in the A/A49E-7(V4) turret assembly. The turret provides for azimuth drive $\pm 110^\circ$ either side of the aircraft's longitudinal centerline, 50° depression, and up to 11.8° elevation. Elevation is limited as a function of gun azimuth to provide safe separation between the projectile path and the aircraft's main rotor. Turret slew rates exceed $80^\circ/\text{second}$ in azimuth and $60^\circ/\text{second}$ in elevation. (DON 1993b). The gun firing circuits incorporate a burst limiting function which provides controlled firing bursts of 16 ± 4 rounds when this firing mode is selected by the operator. The M-197 can also be fired continuously with a maximum firing burst of 450 rounds, after which a six-minute cooling period is required. (DON 1992). Basic gun firing accuracy is very good with a mil dispersion of 7.5 mils, into which the gun will deliver 80% of its rounds. This translates into 80% of a firing burst landing within a $7\frac{1}{2}$ meter circle at 1000 meters target range. (DON 1993a). Figure 4 extracted from the AH-1W Flight Manual (DON 1992) depicts the AH-1W M-197 installation.

Sighting System

The AH-1W uses three gun sights; a stabilized nose-mounted electro-optic sensor, a fixed HUD in the rear cockpit, and a helmet sight system (HSS) with accommodations in

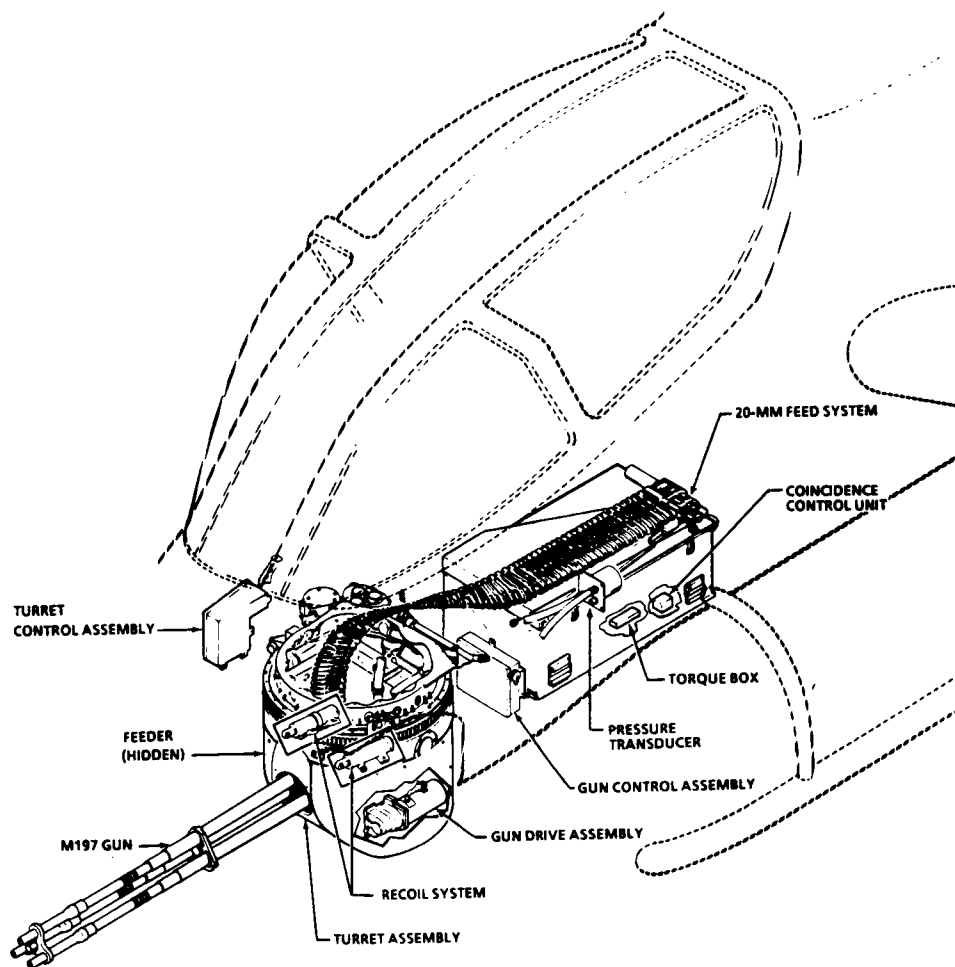


Figure 4: AH-1W M-197 Gun Installation.

Source: DON (1992).

both cockpits [see Appendix A for a discussion of the types of attack helicopter gun sighting systems]. The AH-1W's gun sighting system advantages and disadvantages are summarized in Table III.

The nose-mounted electro-optic sight available to the AH-1W is available in two configurations. The one in widest use is the M-65 TSU, which is an optical sight providing two powers of magnification (2X and 13X) and which is intended for daytime use only. (DON 1993a). The other sight is a modified M-65 TSU known as the Night Targeting System (NTS) which incorporates a FLIR and a TVC along with the original M-65 optics train. The FLIR and TVC provide monochrome output to a instrument panel mounted MFD for heads-up tracking, as well as to a small display mounted in the direct view optics tube originally used for heads-down tracking with the M-65 optics. The FLIR is intended for day/night use; the optics train and TVC are limited by capability to day use only. The NTS is currently being introduced into the USMC AH-1W fleet. (Walters and Brubacher 1992). Figure 5 adapted from the AH-1W Flight Manual (DON 1992) depicts the M-65 TSU and its mounting location in the AH-1W's front cockpit. The mounting location is representative for both the M-65 and the NTS.

The M-65 and the NTS can be used to fire the aircraft's M-197 gun utilizing any sensor mode and magnification with the gun slaved to the sensor LOS. Electro-mechanical resolvers provide sensor elevation and azimuth signals to the gun direction system. Range to the target is manually set utilizing a range potentiometer knob in the front cockpit, though the NTS does provide a range readout from its laser rangefinder. The range knob adjusts the gun's elevation relative to the aircraft ballistic reference line (BRL). (DON 1993a).

Table: III AH-1W gun sighting system advantages and disadvantages.

Sighting device	Gun aiming mode	Advantages	Disadvantages
Helmet Sight System (HSS)	Slaved to pilot's line-of-sight.	•Rapid target acquisition and aiming. •"Look and shoot" firing.	•Helmet slippage and aircraft vibrations reduce accuracy. •Pilot must estimate range and target motion. •Pilot must offset aim point to compensate for aircraft states and rates, and for mentally computed lead and elevation angle.
Heads-up Display (HUD)	Fixed forward, aligned with aircraft ballistics reference line (BRL).	•Open field-of view through HUD. •Stable display.	•Pilot must estimate range and target motion. •Pilot must offset aim point to compensate for aircraft states and rates, and for mentally computed lead and elevation angle.
Telescopic Sight Unit (TSU)	Slaved to TSU's line-of-sight.	•Telescopic sight allows engagement at extended ranges. •Sight mechanical stabilization reduces effects of aircraft vibration. •Laser rangefinder provides accurate range to target.	•Narrow field-of-view limits use in deflection shooting. •No automatic range input to turret control from laser rangefinder. •Pilot must estimate target motion. •Pilot must offset aim point to compensate for aircraft states and rates, and for mentally computed lead and elevation angle.

Source: DON (1993a).

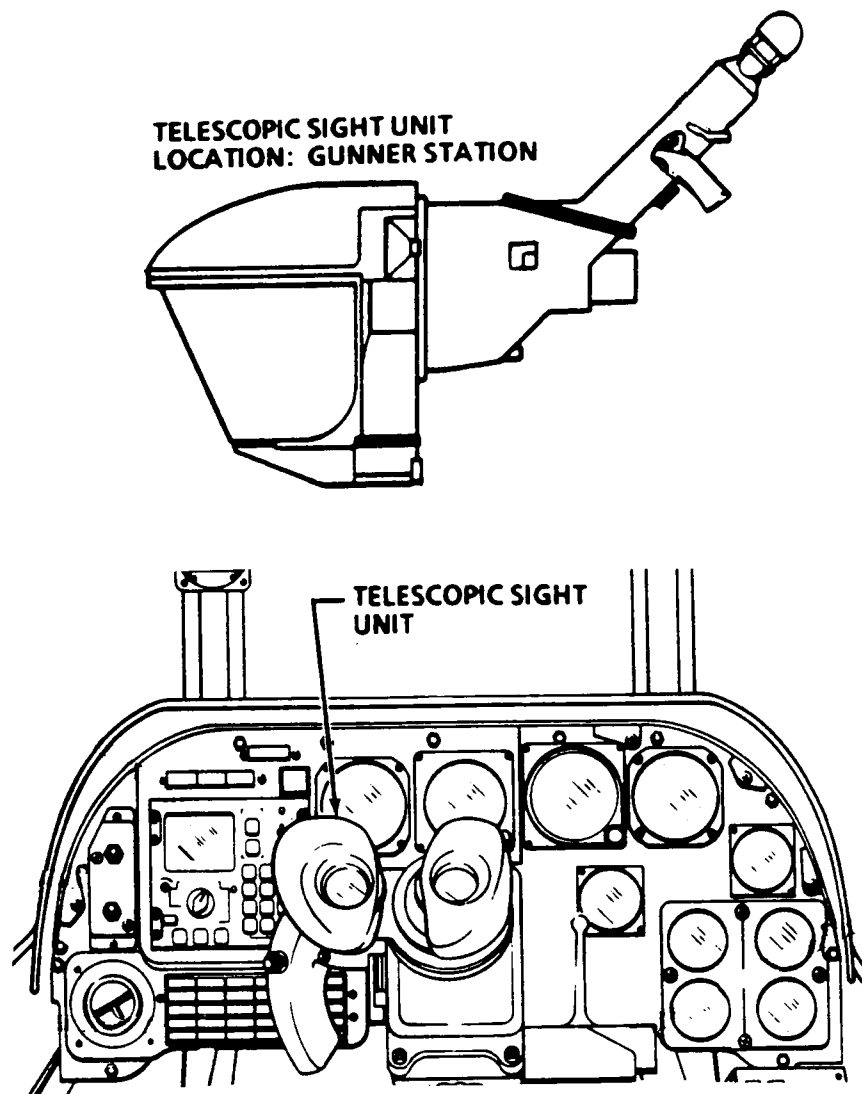


Figure 5: AH-1W M-65 Telescopic Sight Unit (TSU) and Front Cockpit Location.

Adapted from: DON (1992).

The rear cockpit HUD is mounted on top of the pilot's glare shield. The HUD display provides flight and aircraft performance information, as well as targeting displays for those of the aircraft's weapons systems that can be fired from the rear cockpit. The HUD display is day/night compatible, and is also compatible with the pilot's night vision goggles (NVG). When using the HUD as an aiming device for the M-197 gun, the gun is locked in a fixed-forward position, aligned with the aircraft's BRL (0° azimuth, 0° elevation). The reticle position is adjusted vertically on the display for range through the use of a manual range potentiometer control on the pilot's instrument panel. Reticle setting relative to the aircraft BRL is displayed in mils just to the right of the gun reticle on the HUD display. (DON 1992). Figure 6, adapted from the AH-1W Flight manual (DON 1992), depicts the AH-1W HUD with the gun reticle selected, and the HUD's mounting location in the rear cockpit.

The HSS provides a quick pointing capability for the M-197 gun by allowing the pilot or CPG to slave the gun to the operator's LOS. Figure 7, adapted from the AH-1W Flight Manual (DON 1992), depicts the various components of the HSS and the HSS sighting reticle displayed in the helmet eyepiece. Operator LOS is derived through the use of three electro-mechanical position resolvers incorporated in the operator's linkage arm. The output from the resolvers is used to compute helmet position in the cockpit and helmet LOS, which is transmitted to the gun direction system. Range to the target is manually set utilizing a range potentiometer control in the operator's cockpit, which adjusts the gun's elevation relative to the aircraft BRL. (DON 1991).

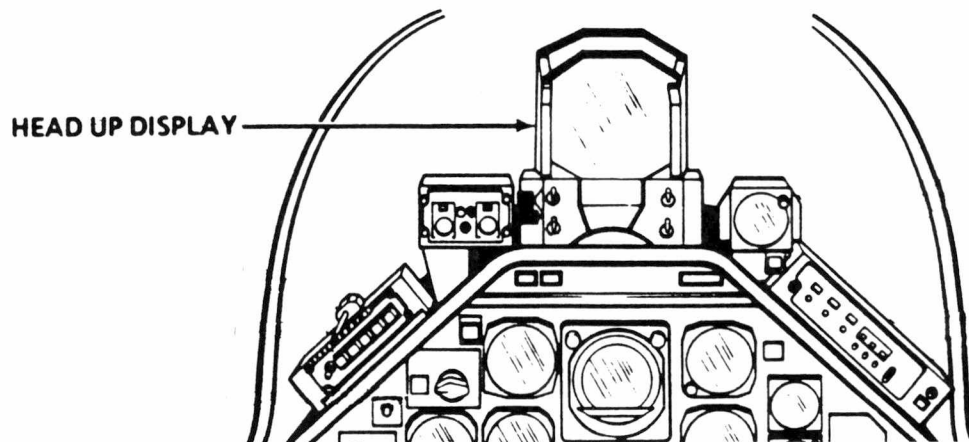
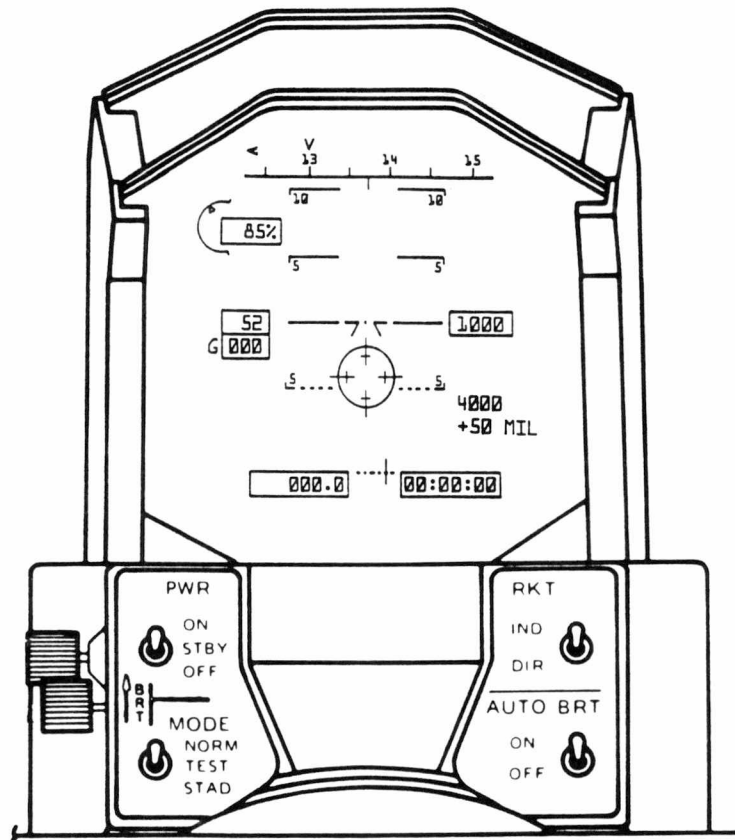
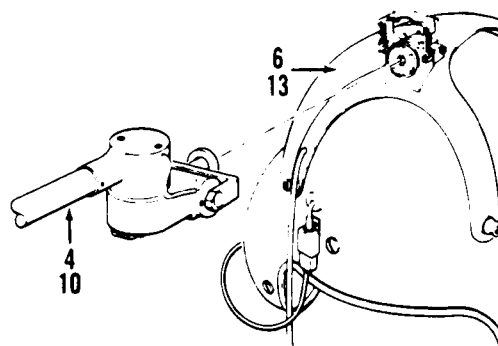
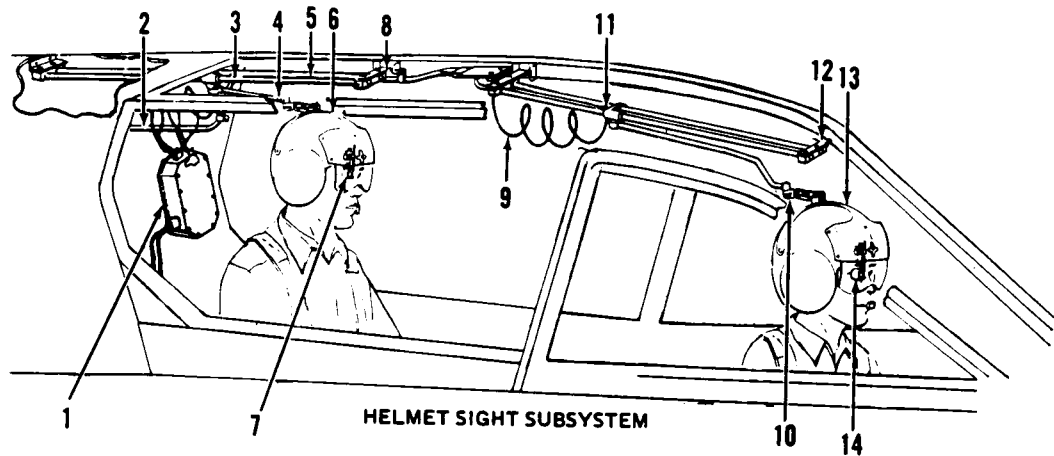


Figure 6: AH-1W Heads-up Display (HUD) and Rear Cockpit Location.

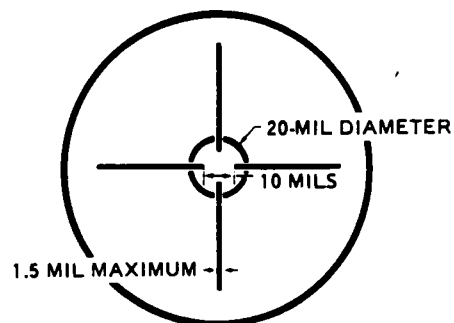
Adapted from: DON (1992).

Note: The HUD display depicted is not to scale and has been reduced in size to prevent obscuration by the HUD structure.



PILOT/GUNNER LINKAGE ARM
ATTACHMENT TO HELMET SIGHT

1. Electronic interface assembly
2. Gunner extension cable
3. Pilot linkage cable
4. Pilot linkage arm
5. Pilot linkage rails
6. Pilot helmet sight
7. Pilot eyepiece
8. Pilot linkage front support
9. Gunner linkage cable



PILOT/GUNNER EYEPIECE
RETICLE PATTERN

10. Gunner linkage arm
11. Gunner linkage rails
12. Gunner linkage front support
13. Gunner helmet sight
14. Gunner eyepiece

Figure 7: AH-1W Helmet Sight System (HSS) Components and Sighting Reticle.

Adapted from: DON (1992).

Gun Direction System

Gun direction is achieved through electronic signals to the turret control assembly (TCA). The TCA contains the turret control logic, and the azimuth and elevation servo amplifiers which control the drive speed and direction of the respective turret drive motors. The gun direction scheme utilizes turret azimuth and elevation position and rate servo feedback loops to rapidly align the gun with the LOS signals transmitted by the active aircraft sight. The LOS signal from the active sight is summed by the interface control unit (IFCU) with the position feedback signal transmitted by electro-mechanical resolvers in the turret azimuth and elevation drive systems. The resultant error signal is transmitted to the TCA which sends a signal to the azimuth and elevation drive motors to drive the turret until the error signal is reduced to a null. (DON 1991). Azimuth and elevation tachometer generators are mechanically coupled to the respective drive system. The output of these devices, known as tach feedback, provides rate control to maintain commanded drive rate, and reverse drive dynamic braking as the turret position converges with the commanded turret direction to prevent overshoot and turret oscillation. Dynamic braking is also used to halt turret movement prior to reaching a turret mechanical travel limit in order to prevent damage to the turret or airframe. (DON 1993b). The coincidence control unit (CCU) monitors the turret azimuth error signal. If the turret azimuth resolver lags the sight by more than 90° , a forcing function is applied to accelerate the rate of the turret in the direction of the sight LOS. (DON 1991). Figure 8, adapted from the AH-1W Armament Systems Principles of Operation Manual (DON 1991) depicts a schematic of the gun direction system.

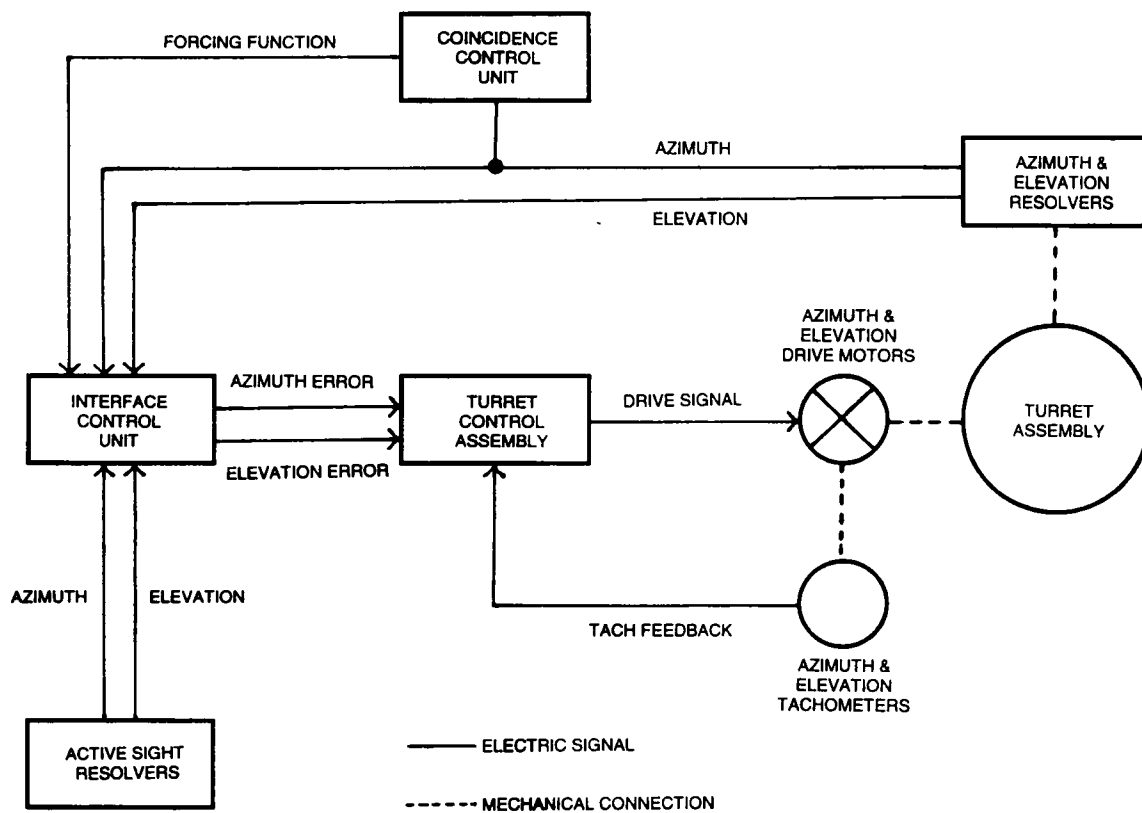


Figure 8: Gun Direction System Schematic.

Adapted from: DON (1991).

LEAD COMPUTING OPTICAL SIGHTS

There are many configurations of lead computing optical sights (LCOS), so before an implementation for the AH-1W is introduced, a short background discussion is warranted. LCOS are designed to relieve the pilot of the necessity of mentally calculating, or more correctly, estimating the lead and elevation angles required to hit a target during maneuvering flight. The basic components of an LCOS are a sight display unit, a gyroscopic sensing unit, and a fire control computer. (Shaw 1985). Additional components may include a range and range rate measuring device (radar, laser, or manual mil referenced measurement), and additional sensors that may provide various inputs of interest to the ballistics computation, such as airspeed, ambient conditions, gun bore line resolvers, or angle of attack feedback. Regardless of the mechanization, the LCOS will normally have to answer two questions in developing a ballistics solution: "Where will the target be at the end of the projectile time-of-flight?" and "For that location, what is the lead and elevation angle and the projectile time-of-flight?". The seeming paradox is that the answer to the first question cannot be determined without the answer to the second, and vice-versa.

Fire Control Basics

The fire control problem outlined above is universal to any weapons system firing at or from a moving platform. The answer to the paradox is that the fire control system must solve for relative target motion and the ballistics solution in a closed-loop, iterative fashion. Since at the outset of the firing computation, neither the future target location or the TOF are

known, a series of approximations must be made until the calculated error falls within acceptable limits. These approximations are performed in the following manner. Initial target location and relative motion data are determined. Initial TOF is computed based on the initial position of the target and internal and external ballistic parameters, and then future target position is calculated based on the initial TOF and an updated set of relative motion data. A new TOF is computed based on the updated future target position and updated ballistics parameters, and so on. Modern computers can step through these iterations quite rapidly, though in an extremely dynamic maneuvering fight, an acceptable firing solution may not be attainable due to rapidly changing ballistics considerations and target relative motion. (Frieden 1985; Shaw 1985). Figure 9 from Frieden (1985) illustrates the iterative process employed in fire control solutions, the example shown being for a ground based anti-aircraft system.

Types of Lead Computing Sights

Lead computing sights can be classified in several ways. For the purpose of this study, they will be examined from three aspects; display type, LOS rate reference, and display time reference. Display type refers to the manner in which the aiming solution is displayed to the pilot or systems operator. Modern fighter aircraft, whose guns are normally (but not always) fixed in an immobile mount, use a "disturbed-reticle" system. The disturbed reticle system is one in which the reticle moves on the sight display in order to induce an aiming correction by the pilot. For example, if the LCOS determined that increasing range to target required an increase in gun elevation, the system would move the reticle downward

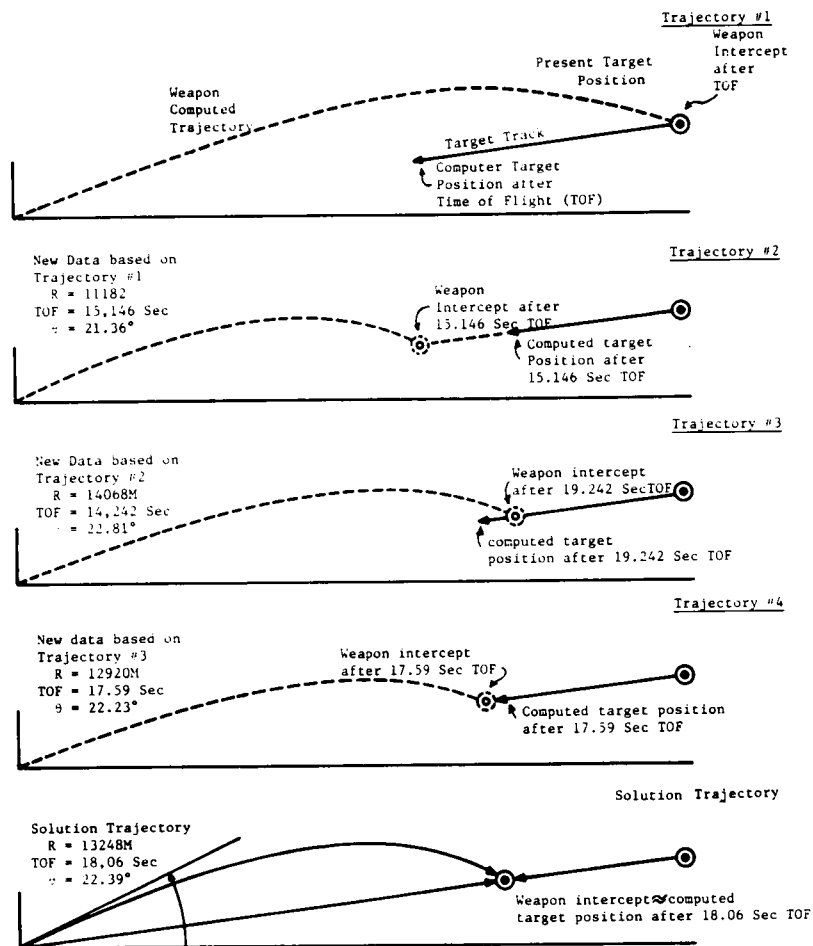


Figure 9: Sample Fire Control Iterative Solution.

Source: Frieden (1985).

on the display, inducing the pilot to raise the nose of the aircraft to maintain reticle-target alignment, adding the required elevation in the form of aircraft pitch. (Shaw 1985). One form of disturbed-reticle sight is known as a "constantly computing impact point" reticle, or CCIP. The CCIP system positions the reticle on the sight to depict where the projectile will be at the end of the TOF for a known or preset range to target. The reticle moves to compensate for external ballistics effects, own aircraft rate and state, and target range and relative motion (if known). The pilot flies the aircraft to overlay the target with the CCIP reticle and fires the weapon. Another mechanization that falls into the category of disturbed-reticle systems is the "constantly computing release point" reticle, or CCRP. This sight displays a target or "fly to" reticle based on sensor data of target location and (possibly) a prediction of future target location, as well as other ballistics considerations. A second reticle is displayed which is normally fixed. The pilot "flies" the fixed reticle to overlay the "fly to" reticle and then fires the weapon. Both the CCIP and CCRP systems are used in air-to-ground applications as well as air-to-air.

The other category of LCOS based on display type is a "fixed" reticle system. Sights of these type provide an indication of own aircraft's plane of motion (the plane in which the projectiles will fire) and often use a preset mil reference for applying lead angle. An indication of the projectile stream may be incorporated, and is sometimes referred to as the "bullet fall line". An interesting range adjusting display may be incorporated, known as a "bullet funnel". The pilot will set a default wingspan for target aircraft in the LCOS computer, and the sight generates a funnel shaped reticle on the display. The bullet funnel is sized so that when the pilot aligns the sight such that the wingtips of the target aircraft just

touch the inside edges of the funnel display, then the appropriate elevation angle will be set. As might be expected, the funnel is wider at the top, since the farther away the target is, the smaller it will appear. A more distant/smaller target fits further down in the funnel (more elevation induced) and a larger/closer target fits nearer to the top of the funnel (less elevation induced). This is an automated version of the manual mil range referencing technique described in Appendix A under *Range Estimation*.

For the LCOS to be able to determine the future position of the target, it must know the rate of the aircraft to target LOS (simply LOS rate), and the range and range rate of the target. The LOS rate may be directly measured by a sensor, such a radar or electro-optical sensor such as the AH-64 TADS, or it may be derived by assuming that own aircraft body rates match the target LOS rate. This assumption is valid only if the pilot can zero the relative motion of the target in the sight field-of-view. Sights that use a radar to track target LOS rate are often referred to as "radar directed" sights, and examples include the F-18 "Hornet" and F-15 "Eagle" LCOS.

The final categorization of the LCOS has to do with the time reference for the sight display. The display may be a "predictor" or "director" sight which shows where the target will be located in the sight field-of view after projectile TOF, or it may be "historical" or "real-time". The historical-type LCOS computes the projectile trajectory, but does not display it until after the TOF associated with the target's present range. If the reticle is superimposed over the target, then the rounds should be impacting the target. The historical sight requires less computational power than the predictor sight since it does not require that the target's future position be calculated, only the projectile's position based on the

calculated trajectory and own aircraft motion during the projectile TOF. However, the pilot must remember to open fire an appropriate TOF before the reticle appears to overlay the target. Additionally, the system can be more difficult to track with, since there is a time lag between the movement of one's own aircraft and the change in the reticle position on the sight. Therefore, fine-tuning aim position can be frustrating. (Shaw 1985).

Attack Helicopter Lead Computing Optical Sights

As noted, LCOS come in numerous variations, only some of which are applicable for use on an attack helicopter. For example, an accurate fire control radar similar to those found on fixed-wing fighter aircraft has not yet been fitted to a production attack helicopter, so a radar directed LCOS would be difficult to integrate. The attack helicopter equipped with an LCOS should have as much in common with a ground based anti-aircraft artillery (AAA) piece as a fixed-wing fighter aircraft, which is not necessarily a bad thing when one considers the lethality of some modern day AAA systems.

One mechanization of an attack helicopter LCOS which bears much in common with ground and ship based AAA cannons makes use of the added flexibility derived from a turreted gun. Instead of making adjustments to the aiming reticle position on the sight display, it is possible to transmit the aiming corrections to the gun in the form of elevation and azimuth commands. The sight reticle is truly "fixed" and the pilot need only to keep the reticle on target, allowing the LCOS to aim the gun in the appropriate direction. This mechanization is well suited for helicopters, since the same ballistics computations can be used for multiple sighting modes (chin mounted sight, HUD, and helmet sight) and it works

well with the onboard ranging sensor most likely to be carried by a helicopter, the laser rangefinder boresighted to the sighting system LOS. This type of LCOS was utilized in the INTAAW project described in Chapter 2.

PROPOSED LCOS AH-1W CONFIGURATIONS

Any manner of known LCOS could be retrofitted to the AH-1W "SuperCobra", if cost were of no concern. But with the current budgetary realities that shape defense acquisition strategies, it is necessary to bound these proposals by requiring the use of as much of the current AH-1W gun system as possible. This study proposes two possible LCOS configurations which were used in the tracking simulation portion of the data collection and analysis effort, with mechanization variations introduced during the ballistics simulation effort. The third configuration examined was a simulation of the current AH-1W gun system.

To limit the scope of the simulation effort, the only sighting mode examined during the gun tracking simulation effort was the use of the rear cockpit HUD. The possible benefits of the proposed configurations to the other sighting system modes will be discussed in the final chapter of the report.

The first configuration examined was a fixed reticle sight with feedback to the gun direction system for lead and elevation angle adjustment. This configuration was similar to the effort undertaken in the AH-64 INTAAW program described in Chapter 2. The sight LOS reference to the LCOS was constant, and would be assumed to be the BRL of the aircraft in the physical configuration. Target LOS rate was derived from time history data during the

ballistics simulation effort, based on recorded own aircraft rates. Range and range rate information in this configuration could be provided through the use of a laser rangefinder with range rate derived through range differentiated over time. The laser utilized could either be the one contained in the NTS, with the sight automatically slaved to the aircraft BRL, or it could be a stand-alone unit installed on the fuselage of the aircraft and boresighted to the aircraft BRL. A stand-alone laser rangefinder would not be prohibitively expensive and would also be effective for ranging for air-to-ground gun and rocket delivery. Additionally, modification of the NTS system to provide laser range during rear cockpit air-to-air engagements would be very complicated. Actual implementation of this configuration would require at a minimum; installation of a stand alone laser rangefinder or modification to the NTS system as noted above, software and hardware modifications to the TCA, the IFCU, and the full function signal processor (FFSP) which provides the HUD display symbology, as well as provision for a measurement system for aircraft state and rates.

The second configuration was a CCIP reticle system, without range or range rate feedback. The gun mode was fixed-forward, boresighted to the aircraft BRL, as is the case in the current AH-1W HUD gun firing mode. The range setting was assumed to be constant at one TOF to simplify the simulation effort, though in the physical configuration the range potentiometer could be retained and a discrete range could be manually set by the pilot. The disturbed-reticle moved to compensate for own aircraft state and rates. Since the assumption was that there was no range or range rate information beyond the constant range set at one TOF, no target future position prediction was provided and the operator had to estimate appropriate lead and elevation angles. The LCOS was mechanized as a predictor sight

(representing point of impact for a round fired at the time of reticle presentation), though it could also be possible to mechanize the sight as a historical LCOS if desired. Actual implementation of this configuration would require at a minimum, software and hardware modifications to the FFSP as well as the provision for some measurement system for aircraft state and rates.

The final configuration represented the current fixed guns firing mode of the AH-1W. The range setting was once again be assumed to be constant at one TOF, though the previous paragraph's comments on this issue apply here as well. This configuration was utilized to provide a baseline for comparison, as well as a model to examine the effects of changing the gun firing rate during the ballistics simulation effort.

These configurations were not chosen to provide an absolute recommendation for an LCOS for the AH-1W, as the quality of the simulation, especially the tracking simulation, would not support this. They were representative however, of two readily differentiable types of LCOS, which served to demonstrate the simulation effort described. The differences in the relative performance of these two LCOS configurations provided some thought provoking results.

CHAPTER 4

GUN TRACKING SIMULATION

The goal of this project was to show the utility of simulation in optimizing a helicopter LCOS and gun system, so as a practical matter it was necessary to develop a computer gun tracking simulation. Without access to a high speed work station, and possessing no expertise in programming for those platforms, the author settled on the use of a desktop personal computer (PC) running the Microsoft Windows® environment. The use of the Windows® environment off-loaded the requirements for input device drivers and management, file interface, and system interrupts to the operating system and allowed concentration on the development of the algorithms which were the heart of the simulation. Appendix C provides a brief description of the system hardware and software used during this project's simulation efforts.

The gun tracking simulation was coded in Microsoft Visual Basic® which was chosen for its ease of use and short familiarization period. The program code is presented in Appendix D. Unfortunately, Visual Basic® has limited graphics generation and handling capabilities which reduced the simulation to a low order of fidelity. Additionally, due to the relative slowness of programs written in Visual Basic® and running under Windows®, some compromises were made in the development of the flight model to preserve system throughput. These compromises will be discussed later in this chapter.

The initial stage of the gun tracking simulation effort consisted of background

research into related simulation efforts for the purpose of gaining insight into the development of the simulation program, development of program design goals, Visual Basic® programming familiarization, and finally the development of the basic logic structure for the simulation.

The next stage of the simulation development effort involved the development of the different mathematical models and algorithms which drove the system and data output, and the actual coding of the program. The last area to be discussed in this chapter is the data collection, reduction, and analysis effort. This effort culminated in the development of the models which were used in the ballistics simulation program.

PREVIOUS SIMULATION EFFORTS

Several simulation programs undertaken by government and business organizations determined to be similar to the effort described in this report, were reviewed prior to commencing work on the gun tracking simulation program.

Martin Marietta Aerospace created a Guidance Development Center (GDC) in the 1960's as a precision test facility for the testing of a wide range of guidance seeker systems. The GDC utilized a physical three-dimensional terrain model over which an electro-optical device traveled. The electro-optical device was mounted in a gimbaled flight table which, when coupled with the device transport mechanism provided six degrees-of-freedom. The movement of the electro-optical sensor was controlled by a computer which could vary motion rates and accelerations to simulate a variety of sensor platforms and missile types.

The terrain model could be lit with a variety of electromagnetic transmitters allowing testing of both visual and infrared sensors. Light emitting diodes placed on the terrain model simulated laser designation spots for testing of laser-homing guidance devices. The GDC provided "man-in-the-loop" target acquisition and guidance to test for tracking ability and miss distances. (Haselwood and Monroe 1973).

In a follow-on program in the early 1970's, Martin Marietta developed the Electro-Optical Simulation System (EOSS) for the Army's Missile Command (MICOM) as part of the Advanced Simulation Facility at Redstone Arsenal, Alabama. The EOSS was similar in concept to the GDC, though the EOSS boasted improved flexibility and capability over the older GDC. An important addition to the simulation concept was the addition of the Weapons Delivery System Simulation (WDSS) subsystem which added a six degree-of-freedom flight simulator based on the UH-1 cabin as an input device to the EOSS. Projected displays in the flight simulator module provided real-time feedback to the simulator operator and allowed the pilot to "fly" in reference to the EOSS terrain model. The stated goal in the integration of the WDSS subsystem to the EOSS was to test the total effectiveness of prototype weapons systems prior to extensive integration and flight test programs. Martin Marietta had determined during the use of the GDC, that "the heavy interplay between the man and the machine" required man-in-the-loop simulation for effective system design. (Haselwood and Monroe 1973).

McDonnell Douglas Helicopter Company's INTAAW program, described earlier in this report, also relied heavily on simulation during its design process. The test aircraft was heavily instrumented and significant data were recorded real-time during several "canned"

air-to-ground and air-to-air flight test scenarios. This data were fed into a complete systems simulation which included all system error models, and which supported the flight test effort. Numerous parameters were recorded including aircraft state and rates, air data measurements, and electro-optic sight measurements. This simulation predicted probability of hit performance throughout the simulation scenario sequence. (Osder and Mossman 1992).

A recent computer simulation developed by the US Government's Survivability/Vulnerability Information Analysis Center has been designed to simulate target detection, tracking, and shooting performance of AAA weapons systems against a passive aerial target. This program, the Radar Directed Gun System Simulation (RADGUNS 1.9) models gun acquisition and tracking systems, target flight path, crew performance, and various radar system parameters and environmental effects. The models of radar performance can be run with a Monte Carlo simulation; randomizing clutter, radar multi-path, and glint (environmental effects effecting radar performance). The program produces P_H , P_K , expected number of hits, and detailed studies of system performance. (SURVIAC 1995).

DESIGN GOALS

The gun tracking simulation program was designed to meet a set of goals which provided direction and structure to the programming effort:

The program had to be simple to design, code, and operate. The simulation was viewed as a research tool rather than a device designed to accurately represent a specific

aircraft/gun system. This approach allowed flexibility in developing the flight model and reduced the programming effort to a manageable level.

The program was designed for operation on a single computer. No effort was expended in trying to make the program compatible with various computing platforms, though some level of compatibility was assured through the use of a Windows® based programming environment. The upshot of this goal is that while the program will generally run without failure on any Windows® based machine, the rate of control response will normally vary from computer to computer. The primary input device was designed to be a commercial gaming joystick, though a level of keyboard control was incorporated to aid in the development process.

The program was designed to operate at "best speed". That is, the computing loops were not timed, but were allowed to operate as fast as the computer could process the required calculations. System control response was set through the use of discrete constants which governed apparent pitch and roll rates and damping. To ensure that processing speed did not vary among the different LCOS models, all required computations for each model were executed during each processing iteration, irrespective of the active simulation model. While this had the effect of slowing the overall processing speed of the program, it ensured that no hardware bias would effect the accuracy of the data output. Tracking simulation run times were limited by maintaining a relatively constant target closure rate, and ending the simulation run when the target reached a defined minimum distance from the simulated firing platform.

The program only modeled a helicopter's pitch and roll control. Power settings were

assumed to be constant and airspeed was allowed to vary according to aircraft pitch and roll attitude within predefined limits. Aircraft sideslip and fuselage angle-of-attack were assumed to always be null. These assumptions were required due to the lack of appropriate input devices for power and yaw control.

SOFTWARE FAMILIARIZATION AND PROGRAM BASICS

Due to the innate limitations of the Visual Basic® programming language, it was determined that the gun tracking simulation program's graphical environment would need to be extremely rudimentary if the program was to be able to run at a speed that would allow direct input from the operator and timely visual feedback from the computer. A simple world-view was developed consisting of a horizon line and several perspective lines, all of which moved relative to the operator's input. The basics of visually referenced flight require that the pilot/operator be able to discern aircraft pitch and roll angle through reference to the outside world, with the primary visual reference being the horizon line. To provide a visual sensation of movement, the perspective lines were incorporated in such a fashion so that they scrolled left or right in response to the operator's commanded roll angle. The perspective lines were referenced to the horizon line through the use of trigonometric transforms which enabled their use in operator orientation. Figure 10 is a screen printout of one of the simulation models and it shows a static representation of the horizon and perspective lines.

Additional operator reference was provided through the use of digital readout windows which gave numerical representation of computed pitch and roll angles, as well as

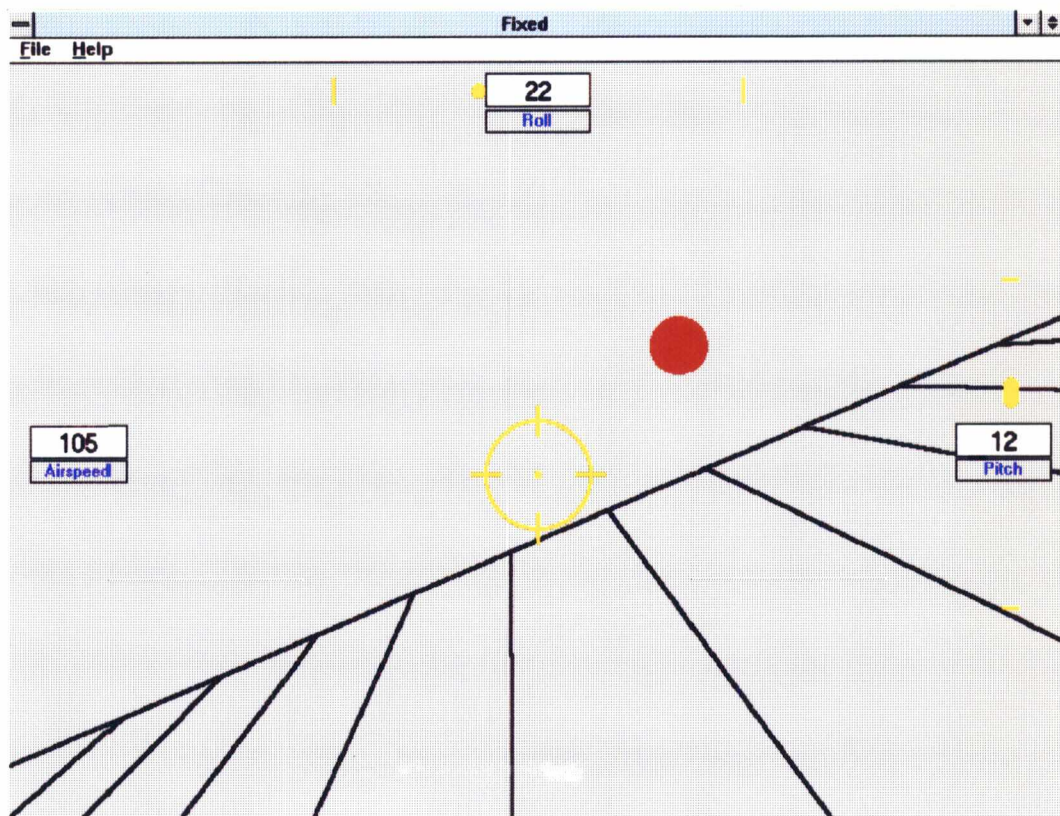


Figure 10: Illustration of Fixed Reticle-Gun Compensation Model.

Notes: The illustration is a screen printout of the gun tracking simulation during a sample run. The reticle remains stationary in the center of the computer screen, and the target and attitude reference lines move independently to provide a visual sensation of flight. Roll and pitch rate indicators can be seen as yellow bars to the left of the roll digital window and above the pitch digital window. Rate limit marks to the left and right, and above and below these windows respectively, give a visual indication of the amount of commanded pitch and roll rates.

a window which read out computed system airspeed. Pitch and roll rate cues were provided through the use of thermometer-like graphical readouts which increased in length as pitch or roll rate increased. These graphical representations of pitch and roll rate can be seen in Figure 10 adjacent to the pitch and roll angle digital windows. A rate limit line provided a reference so that the operator could estimate what percentage of maximum allowable pitch and roll rate was being generated.

Several different target representations were attempted, but the slowness with which the Visual Basic® systems handles graphical objects, such as Bitmap objects, and the limits on computational speed for determining target aspect and sizing lead the author to use a simple filled circle as a target. The size of the target circle varied with computed range to the target, and thus provided a visual indication of target distance and closure rate. The target size was modeled to equate to a circle ten feet in radius. Figure 10 provides a depiction of the target graphic.

The reticle representations were based on the AH-1W's HUD rocket reticle, and the reticle was sized such that the reticle and the target would be the same size at the completion of the simulation run. The size of the reticle graphic on the computer monitor screen was designed to approximate the visual size of the AH-1W's HUD rocket reticle when viewed from a normal pilot flight head position.

The modeled LCOS and gun system configurations were implemented so they operated sequentially. During training modes (discussed below), the operator had the option of selecting a specific model for practice. During data collection runs, the models were presented in order, with the data being written real time to a data file. Each model used the

same algorithms for control response and ballistic computation, with the only variations being the method with which the reticle was presented, and what data were written to the data file.

The first model presented was a fixed sighting reticle simulating all required aiming compensation being made at the gun (fixed reticle/gun compensation). The tracking method used by the operator was to track the center of the target graphic with the center of the reticle graphic. Figure 10 presents an illustration of this first model in use.

The second model simulated a fixed reticle-fixed forward firing gun system (fixed reticle/fixed gun). The reticle representation was the same as for the first model, but the operator was required to estimate and manually control reticle lead for target motion compensation, reticle elevation for bullet drop over range, and reticle positioning compensation for own aircraft state and rates. This model simulated the current fixed forward air-to-air gun firing mode of the AH-1W.

The final model represented a CCIP targeting system for a fixed forward firing gun (CCIP reticle/fixed gun). The operator was required to estimate and manually control reticle lead for target motion compensation and reticle elevation for bullet drop over range. The CCIP reticle was continuously repositioned on the screen to compensate for own aircraft state and rates to provide cuing for aiming compensation, and a BRL graphic was incorporated underlying the CCIP reticle in a contrasting color to provide a reference to reticle movement. CCIP motion was modeled for the range equating to the distance covered by a projectile during one second of flight time (one TOF). Therefore, at distances to target greater (or less) than one TOF, the operator had to add more (or less) aiming compensation

than was indicated by reticle motion. Figure 11 is a screen printout of a CCIP model training run and it shows a static representation of the CCIP reticle and BRL graphic.

To aid in operator familiarization, the program was designed to operate in three modes; two training modes and a data collection mode. The first training mode was designed to provide a visual aim point reference for the second and third simulation models. A white dot was drawn on the screen to provide visual reference for the proper reticle-to-target positioning. This aim point was positioned using the algorithms developed for target motion compensation and bullet drop over range. Additionally, the aim point moved to adjust for own aircraft rate and states during the fixed gun firing mode. No aim point was provided for the fixed reticle/gun feedback model, as the operator was merely required to track the center of the target representation. The graphical aim point representation is depicted in Figure 11, and can be seen just above and to the left of the center of the target graphic.

In both training modes, if the operator positioned the center of the aiming reticle within a target radius of the aim point (graphically depicted or not), then the target changed color from red to blue. This change is apparent when comparing Figures 10 and 11. During testing of the simulation, several operators requested a scoring readout, so that they could have a gauge of the quality of their tracking runs. A scoring system was incorporated using the same algorithm that drove the change in target color and which provided a score based on the percentage of time that the operator was able to keep the center of the reticle within one target radius of the computed aim point. This raw accuracy score proved useful both in training and in maintaining the test operator's interest during the repetitive tracking runs.

When operating in the data collection mode, no aim point was presented and the

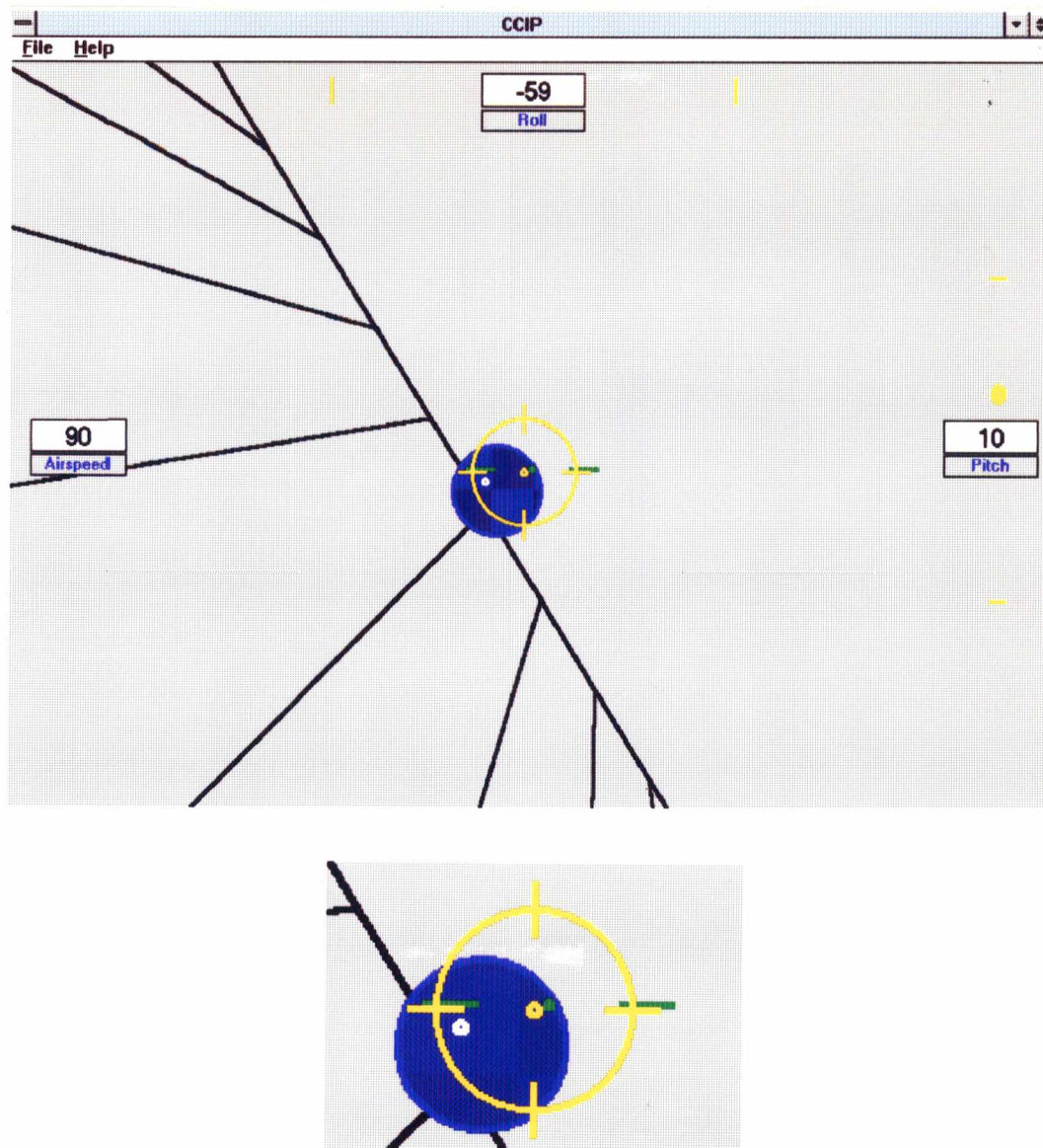


Figure 11: Illustration of CCIP Reticle-Fixed Gun Model.

Notes: The illustration is a screen printout of the gun tracking simulation during a training mode run of the CCIP model. The inset is an enlargement of the center portion of the full screen picture. The reticle can be seen to be displaced below and to the left of the green BRL representation. The BRL representation is stationary and provides a reference to estimate reticle movement. The white aim point indicator visible during the first training mode is displayed just below the left reticle tick mark. The target is blue, indicating that the reticle center is within one target radius of the computed aim point, a feature of both of the program training modes.

target color feedback algorithm was suppressed, so that the operator had to estimate the correct aiming point without assistance from the program.

Training for system operation was accomplished in the following fashion. Several runs were made in the first training mode to provide familiarization in system response to operator input (ie. flight familiarization) as well as to provide training in appropriate lead, bullet drop, and other aiming compensations for the CCIP and fixed gun modes. After an acceptable level of proficiency was achieved (as determined by the test subject), runs were made in the second training mode. No aim point was provided (except for the target itself), but the target color feedback mechanism reinforced correct reticle-to-target orientation. When a reasonable raw accuracy score was achieved, the test subject was deemed ready to proceed to data collection runs.

FLIGHT AND TARGET MOTION MODELS

The initial attempts to develop a realistic flight and target motion involved the development of a classical three-dimensional coordinate system within which the simulated firing platform and target would maneuver in relation to each other. These efforts quickly ran aground due to the slowness of programs written in the Visual Basic® language and operating in the Windows® environment. The use of conventional spacial tracking for simulated firing platform and target required numerous trigonometric transforms which slowed the processing speed of the resulting program to a crawl.

As an alternate approach, the flight model and the target motion models were

decoupled and referenced to the firing platform. Using this technique, the author was able to develop a simulation that provided a reasonable level of processing speed.

The flight model is based upon generation of own aircraft pitch and roll rates referenced to controller position. The flight control system in the AH-1W is essentially a rate controller, though the pilot does not command rate directly, but rather develops pitch and roll rates through orientation of the aircraft lift vector. The simulation program commands pitch and roll rate as a factor of percentage of controller displacement from center, multiplied by a pre-defined maximum pitch and roll rate. The maximum rate is defined in terms of maximum angular rate per second, and the commanded rate developed from controller displacement was normalized for each processing iteration by multiplying the gross rate command (percent of controller displacement multiplied by maximum defined rate) by the length of the processing loop in fractions of a second.

To provide more realism to the flight model, pitch and roll rates were reduced by a damping factor during each processing iteration. To simulate the natural speed and turn stability of the AH-1W, roll and pitch attitude was also reduced during each processing loop. The resulting pitch and roll rate commands were integrated by the flight model to develop pitch and roll attitude, and those angles were processed to displace the horizon and perspective lines displayed on the computer screen.

A module was incorporated into the flight model to compute the firing platform's airspeed based on deviation from a nominal airspeed caused by pitch and roll angle. The airspeed was bounded to maintain forward flight characteristics and a relatively constant target closure rate.

After the basic structure of the flight model was established, numerous test runs were made by the author and volunteer operators. Based on operator comments, the maximum roll and pitch rates and the damping and stability factors were adjusted to maximize controllability while maintaining the desired level of maneuverability. The final control coefficients were generally successful in providing a realistic flight simulation within the limits of the computer's ability to generate the graphical display environment, though the model seems a bit sensitive in the roll axis and most operators generated some operator induced roll oscillation, until they became familiar with the flight characteristics of the system.

The target motion model was designed with two modules. The first provided representative target relative motion based on roll and pitch angles computed by the flight model. The second was designed to generate a moderate amount of evasive maneuvering by the simulated target. The relative motion module drove the target graphic up and down on the screen with reference to the firing platform's pitch angle, and left and right on the screen with reference to roll angle. No actual displacement was computed based on separate flight paths, and in this respect the target motion model is highly unrepresentative of actual air-to-air flight engagements. Mitigating this lack of realism to some extent was the general lack of relative motion during most tracking runs. In the final analysis, the primary purpose of the relative motion module of the target motion model was to allow the operator to be able to null out the target's evasive maneuvering during tracking. Disregarding the normal increase in range found when maneuvering away from a target also had the benefit of maintaining a relatively constant target closure rate, which helped in obtaining near constant simulation run

times. Target closure rate was simulated by decrementing the range to target based on run time and own aircraft velocity generated by the flight model. Closure rate was visually depicted by changing the size of the target graphic based on computed range to target.

The evasive maneuvering module of the target motion model was designed to provide target motion to add gain to the operator's tracking task. The target evasive maneuvering was generated by adjusting lateral and vertical movement direction and rate based on run time and distance to target. If the program was allowed to run without operator input, the target would seem to take a spiral path out from its original location. During actual tracking runs however, this natural flight path was nulled through operator inputs and the target relative motion module.

BALLISTICS COMPENSATION MODEL

Ballistics calculations were made as part of the gun tracking simulation in order to calculate appropriate aim point for the purpose of providing feedback to the operator during training, for CCIP reticle positioning, and for computation of the amount of reticle misalignment during data collection runs. The various ballistic factors computed were a subset of the larger number of ballistics factors which are more fully explained in Appendix A.

Prior to developing the ballistics compensation models, a review of the software lines of code for the F-18 LCOS system was accomplished. While not directly related to the software coding accomplished in the development of this simulation, the review of the F-18

code provided additional information into the method for computing the factors effecting the ballistics computations for an LCOS. In particular, the F-18's LCOS system's use of filtered aircraft body rates to estimate line-of-sight to the target motion during some of its firing modes and variation of range-to-target measurement device during different firing modes were replicated in this program. (MDA n.d.).

Ballistics compensation computed by the gun tracking simulation was determined using the following models and algorithms:

Bullet Drop Over Range and Projectile TOF

Bullet drop over range and projectile TOF were modeled from the ballistics tables for the PGU-28/B 20mm projectile found in the AH-1 Tactical Manual (DON 1993a). The tables list projectile TOF, downrange travel (ft), horizontal velocity (ft/sec), and altitude drop (ft) for four different aircraft airspeeds (0 though 150 knots in 50 knot increments). The values from these tables were entered in a Microsoft Excel® spreadsheet and the points were curve-fitted by the Excel® program. The resulting fourth order equations were entered in a subroutine in the gun tracking simulation and were accessed to compute TOF in seconds as a factor of range to target, and bullet drop in feet as a factor of TOF and aircraft airspeed.

Roll Angle Compensation

Compensation for aircraft roll attitude was modeled based on an adjustment of elevation and lateral lead angle in mils multiplied by the sin of the flight model computed

roll angle. The computation reduced the computed elevation lead angle proportional to roll angle, and increased lateral lead angle in the direction of roll, proportional to roll angle.

Pitch Angle Compensation

Compensation for pitch angle was accomplished through the use of the appropriate formula found in the reference *Rifle Shooting* (THE 1962). The formula provides the ratio of the actual range to target divided by the range to target adjusted for elevation angle effect (see Appendix A for a more complete discussion of this phenomenon). The formula was entered into a subroutine in the gun tracking simulation inputting pitch angle and outputting the previously mentioned ratio. Vertical lead was adjusted by decreasing the calculated bullet drop distance based on a reduction in range to target computed in the subroutine.

Turn Rate Compensation

A turn rate model was developed by digitizing a graphical representation of turn rate, referenced to angle of bank and airspeed, found in the AH-1W Flight Manual (DON 1992). Data points were extracted from the graph for various turn rates (in °/second) referenced to specific airspeeds and roll angles. These points were entered in a Microsoft Excel® spreadsheet and the points were curve-fitted by the Excel® program. The resulting second order equations were then combined with the known distance from the AH-1W's center of rotation to gun muzzle to provide an equation which computed lateral velocity of the projectile based on the roll angle and airspeed computed by the flight model. The lateral distance covered by the projectile due to turn rate was computed based on the computed

lateral projectile velocity and TOF to the target. Lateral lead angle was adjusted to account for the computed distance.

Pitch Rate Compensation

Pitch rate computed by the flight model was combined with the known distance from the AH-1W's center of rotation to gun muzzle to provide an equation which produced vertical velocity of the projectile as a function of pitch rate. The vertical distance covered by the projectile due to pitch rate was computed based on the computed vertical projectile velocity and TOF to the target. Vertical lead angle was adjusted to account for the computed vertical distance of travel.

Target Motion Compensation

While not a ballistics computation *per se*, target motion compensation was computed and factored in to the elevation and lateral lead angles. Target motion direction and rates were provided by the target evasive maneuvering module of the target motion model. When combined with the TOF to the target (based on range to target), target motion provided an adjustment to the computed aim point which was used for the purpose of providing feedback to the operator during training and for computation of the amount of reticle misalignment during data collection runs.

DATA RECORDING

Provisions for data output and recording were made in the gun tracking simulation. The data parameters were selected to provide both time history readouts of significant measurements, and to provide a basis for the models which made up the heart of the ballistics simulation program, which is the subject of the next chapter. Data were recorded to computer memory real-time during data collection runs as comma delimited text files. One line of data were recorded during each processing loop. After the data collection run, the data file was saved to a computer floppy disk. The selection of the comma delimited text file format was made to ensure compatibility with the data analysis software and graphing software used during the data processing phase of the gun tracking simulation effort. The data elements recorded are listed in Table IV.

DATA COLLECTION

Data were collected in a series of tracking simulation runs over the course of several weeks. Volunteer operators were drawn from the pilots and engineers assigned to the US Naval Rotary Wing Aircraft Test Squadron at Patuxent River, Maryland. The total number of runs and participants were limited in order to better manage the large data manipulation effort that followed the data collection period. Operator familiarization and training and subsequent data collection runs took over an hour per operator. Once trained and familiar with the flight characteristics of the simulation, more than one data run might be recorded

Table IV: Data Elements Recorded by the Gun Tracking Simulator.

Data Element	Units	Remarks
Incremented Run Time	Seconds	Total run time from beginning of data collection run
Processing Loop Time	Seconds	Time for individual processing loop iteration
Target Distance Bin	Feet	Distance to target rounded up to the next highest hundred foot increment
Target Distance	Feet	Distance to target
Target Radius	Mils	Angular size of target (a factor of target size and distance to target)
Lateral Miss Distance	Mils	Lateral angle between reticle center and computed aim point
Vertical Miss Distance	Mils	Vertical angle between reticle center and computed aim point
Pitch Angle	Degrees	Pitch angle computed by flight model
Pitch Rate	Degrees/ Second	Pitch rate computed by flight model
Roll Angle	Degrees	Roll angle computed by flight model
Roll Rate	Degrees/ Second	Roll rate computed by flight model
Airspeed	Knots	Airspeed computed by flight model
Lateral Joystick Position	Per Cent	Joystick position measured as a percentage of total travel referenced from center position
Fore and Aft Joystick Position	Per Cent	Joystick position measured as a percentage of total travel referenced from center position

by an operator, though some operators only provided one data run. The author, due to the large experience base garnered in developing the gun tracking simulation, provided several data runs. In all, 6 volunteer operators provided a total of 15 data runs during the data collection period. Some additional data collection runs were made by the author for the purpose of collecting data in non-standard operating modes. For example, several data runs were made with the target evasive maneuvering module of the target motion model suppressed, in order to analyze the effects of target evasive maneuvering on operator tracking accuracy.

DATA REDUCTION

The data were reduced in order to develop the models which drove the ballistics simulation program, and to analyze the tracking simulation output to gain insight into the variation in tracking performance among the different gun tracking simulation models. The data were combined in a variety of data files and were processed using Microsoft Excel® and Jantel Scientific SigmaPlot®. Data reduction techniques varied according to the model to be constructed and the type of analysis performed. Statistical modeling was performed by grouping data parameters and using the statistical analysis tools provided with Microsoft Excel®. Typical statistical analysis performed was the computation of mean data values, determination of maximum and minimum values in a data set, and computation of standard deviation. Graphical analysis was accomplished using Jantel Scientific SigmaPlot® in order to visually assess the central tendencies and shapes of the data sets.

DATA ANALYSIS AND MODEL DEVELOPMENT FOR THE BALLISTICS SIMULATION

The data were analyzed for the purpose of developing the required models for the ballistics simulation, and to assist in developing conclusions concerning the efficacy of the simulation method and the modeled gun systems.

The raw tracking accuracy scores for each data run were extracted from the data files and subjected to statistical analysis. Table V lists the results of this analysis. These scores represented the percentage of time that the operator was able to position the center of the sighting reticle within one target radius of the required aim point. The raw accuracy scores are not the best indicator of overall system performance, but they provided a qualitative feel for the difference in operator tracking accuracy caused by the characteristics of the modeled gun systems. Operator comments during data runs indicated that the proper aim point was difficult to determine even after much practice with the simulator's training modes, and that the easiest tracking task involved maintaining the reticle on the center of mass of the target in the first modeled system; fixed reticle/gun compensation. The general difficulty most pilots experience in estimating lead for target motion and elevation for projectile drop is well documented and was discussed earlier in this report. The raw tracking accuracy scores generated during the data runs are in keeping with real world experience.

The effect of target evasive maneuvering on tracking accuracy was examined by comparing the total misalignment of the sighting reticle during four data runs against a target with the evasive maneuvering motion module engaged and disengaged. The tracking task was accomplished using the first gun system model, fixed reticle/gun compensation, and the

Table V: Raw Tracking Accuracy Score Data.

	Configuration 1	Configuration 2	Configuration 3
Mean	81.4%	28.4%	27.6%
Standard Deviation	14.4	12.8	13.5

Notes: 1) Configuration 1 simulated a fixed sighting reticle with all ballistic compensation being made at the turreted gun. Tracking method was to maintain the reticle on the center of the target.

2) Configuration 2 simulated a fixed sighting reticle with all ballistic compensation being made by the operator. Tracking method was to position the reticle to account for target motion, ballistic factors, and own aircraft state and rates.

3) Configuration 3 simulated a CCIP sighting reticle which moved to provide aiming cuing for own aircraft states and rate compensation. Ballistic compensation and compensation for target motion was made by the operator. Tracking method was to position the reticle to account for target motion and ballistics factors.

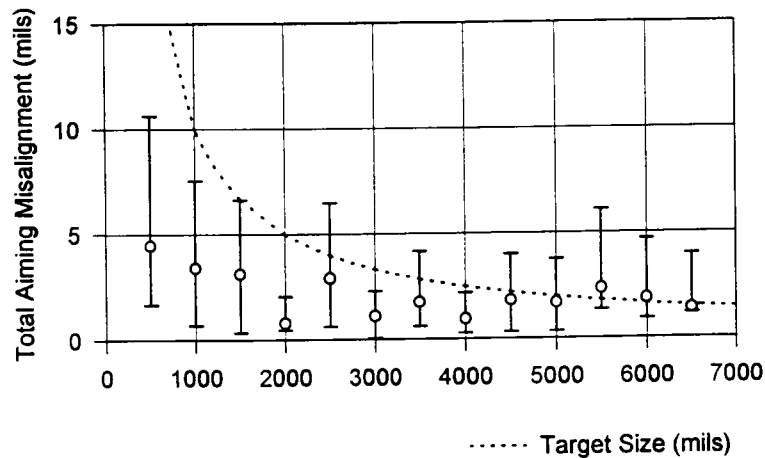
4) Scores reflect the percentage of time that the operator was able to position the center of the sighting reticle within one target radius of the required aim point. Raw scores were extracted from the data files created during the 15 data runs.

tracking method was to keep the center of the reticle on the center of the target graphic. The tracking task with the evasive maneuvering motion module disengaged was very low gain, and misalignment of the sighting reticle was minimized. Figure 12 is a graph depicting the results of this comparison. The error bars shown depict the maximum and minimum misalignment values found at each range, the data symbol represents the mean misalignment for each range. The curved lines on the graph represent the relative size of the target graphic. Wherever a data point or error bar exceeds the target size curve, the operator allowed the reticle to drift off of the target graphic.

Time histories of controller motion during data runs were examined to gain insight into relative operator effort required by each gun system model. Figures 13a and 13b depict these time histories. There is no significant difference between the three modeled configurations in the fore and aft axis (Figure 13a), but in the lateral axis (Figure 13b), controller activity was almost twice as great at the longer ranges associated with the beginning of data runs. The effects of the higher rates of target motion associated with the close-in ranges found at the end of data runs can be seen as greater levels of controller displacement later in run time on both sets of plots.

Data sets of tracking accuracy data were graphically analyzed to assist in the development of the mathematical models to be used in the ballistics simulation program. Figures 14a, and 14b present a portion of the respective data sets for lateral and vertical reticle misalignment. Due to limitations of the graphing program, the entire data sets cannot be presented. The data sets were loaded into an Excel[®] spreadsheet and various curve-fit

Moving Target



Stationary Target

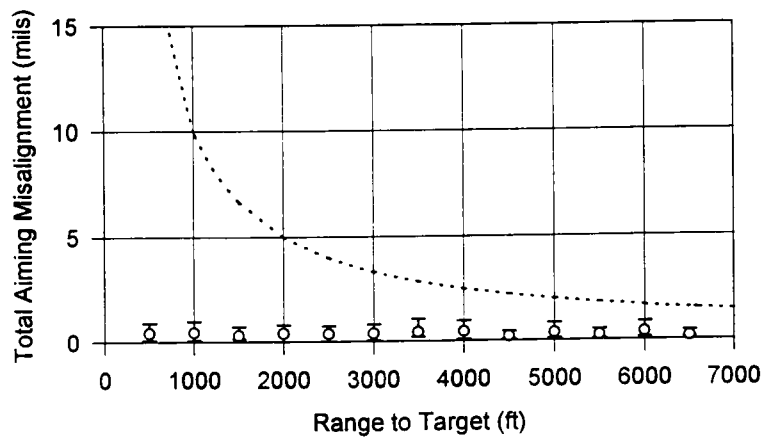
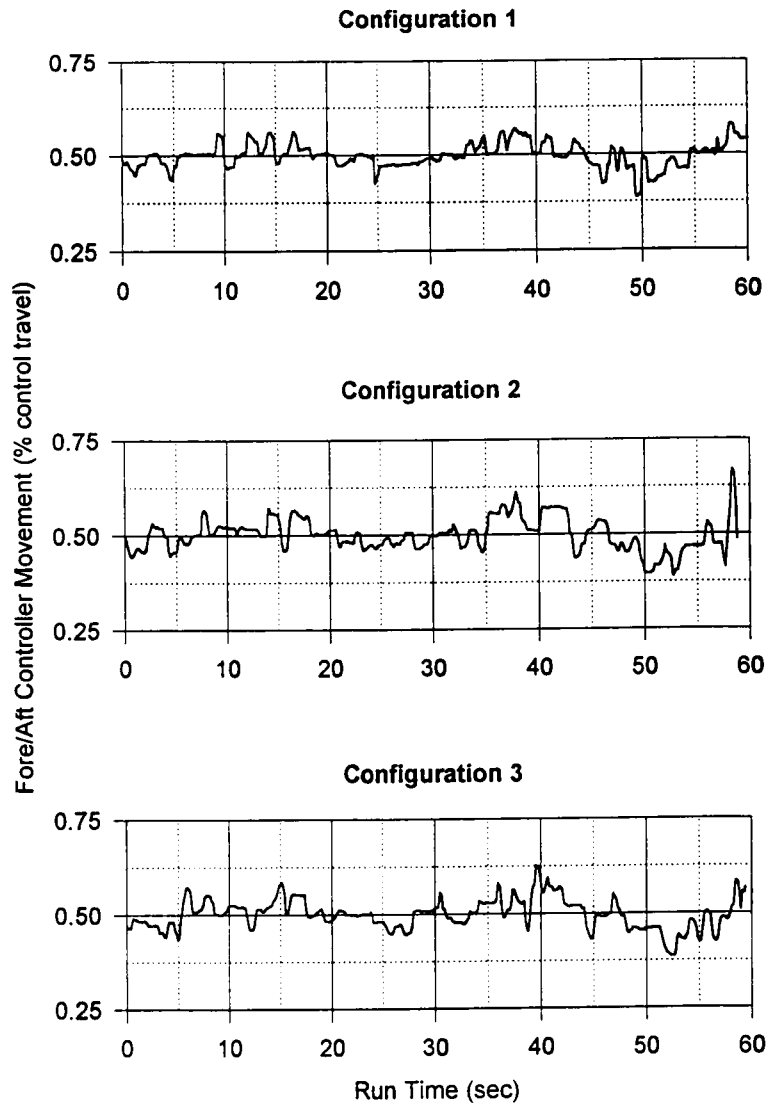


Figure 12: Graphical Comparison of Total Aiming Misalignment When Tracking a Moving Target and a Stationary Target.

Notes: The data runs were made using the first gun system model, which simulated a fixed sighting reticle with all ballistic compensation being made at the turreted gun. Tracking method was to maintain the reticle on the center of the target. The error bars represent the maximum and minimum values in the data sets for the ranges indicated. The data symbol represents the mean of the data set for each range indicated. The dotted line represents the relative size of the target in mils. Where an error bar or data symbol exceed the target sizing line, the operator allowed the reticle to drift off target.

Typical Controller Motion Time History

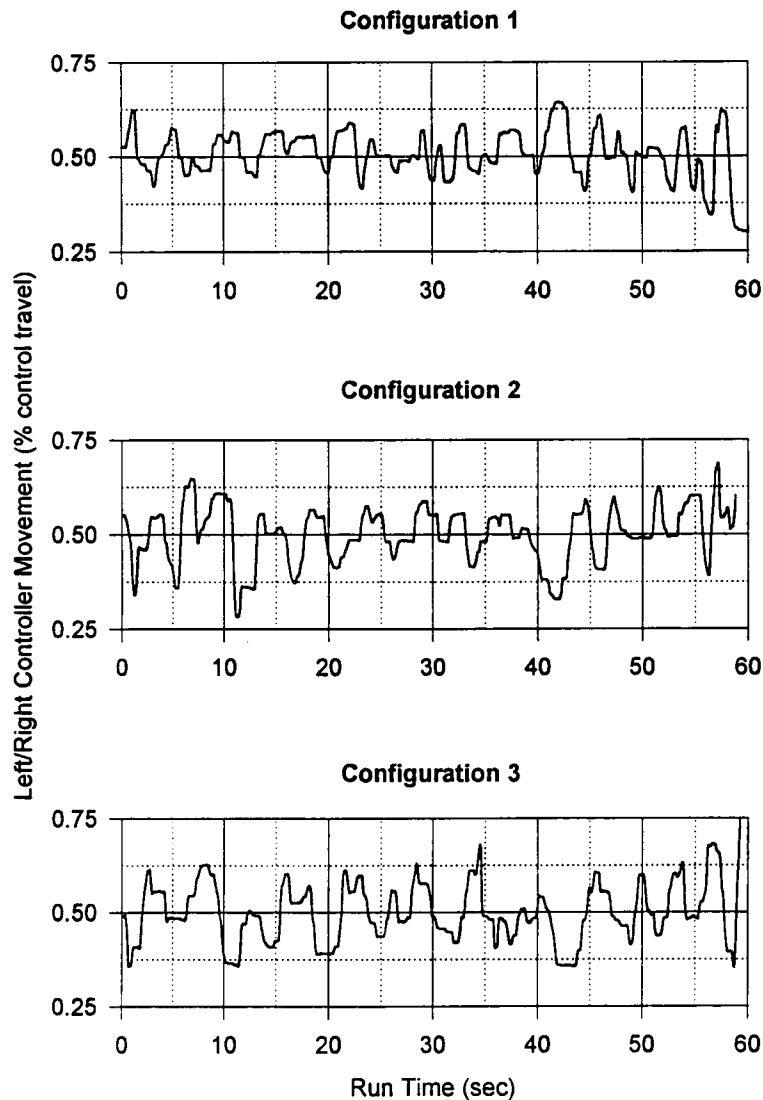


(a) Fore/Aft Motion

Figure 13: Controller Motion Time History.

Notes: The variation in run time between the time histories is the result of a change in computed airspeed during the run effecting the target closure rate. The increase in controller activity near the end of the run is a result of increased target relative motion rate as the range to target decreased.

Typical Controller Motion Time History

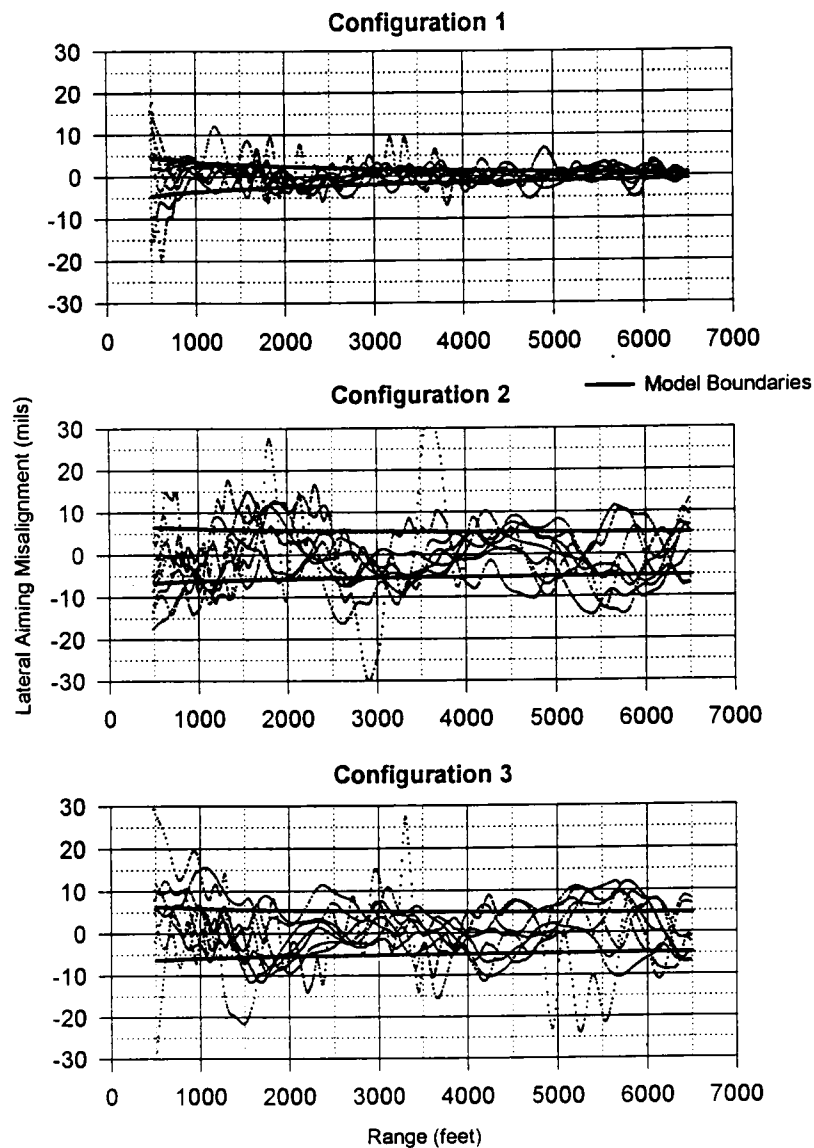


(b) Lateral Motion

Figure 13:

(Continued).

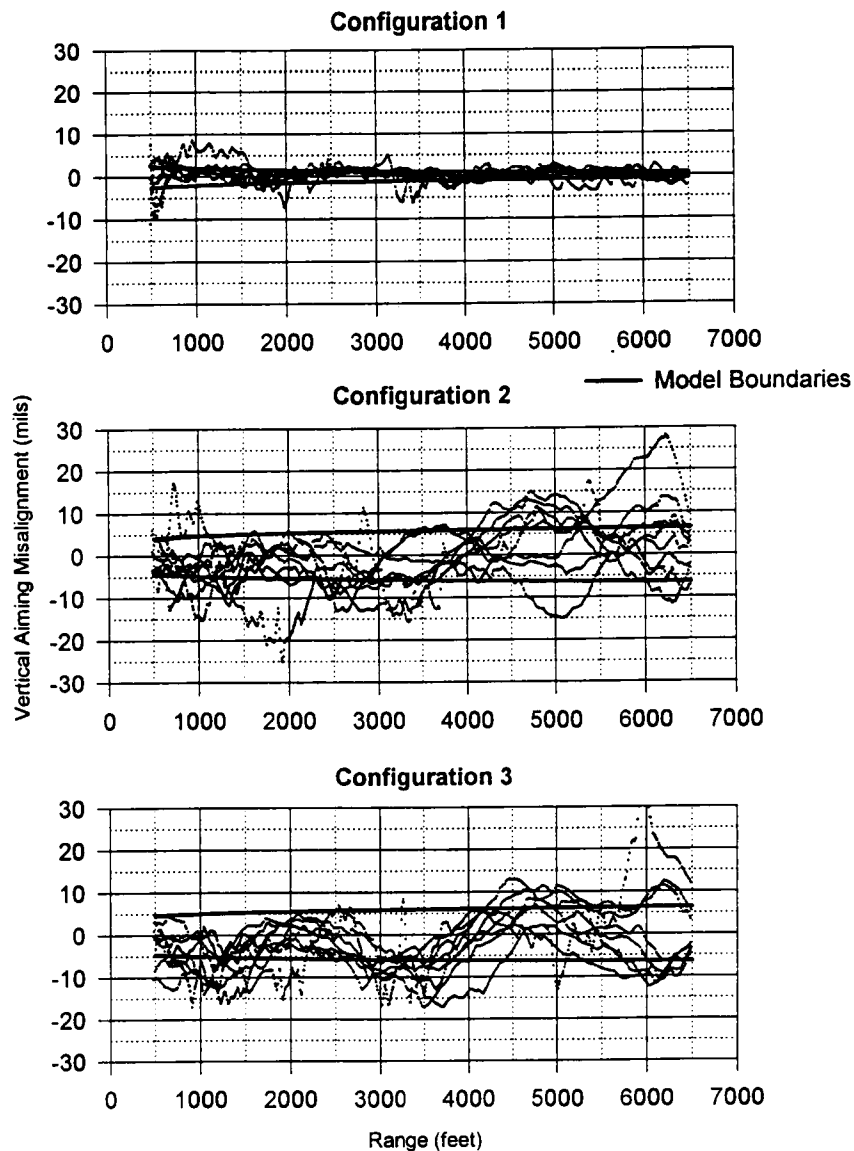
Notes: The variation in run time between the time histories is the result of a change in computed airspeed during the run effecting the target closure rate. The increase in controller activity near the end of the run is a result of increased target relative motion rate as the range to target decreased.



(a) Lateral Misalignment

Figure 14: Aiming Misalignment versus Range to Target.

Notes: These graphical representations present only a portion of the total data sets collected during the data collection effort. The heavy lines on the graphs represent the logarithmic models developed to simulate tracking error in the ballistics simulation program



(b) Vertical Misalignment

Figure 14: (Continued).

Notes: These graphical representations present only a portion of the total data sets collected during the data collection effort. The heavy lines on the graphs represent the logarithmic models developed to simulate tracking error in the ballistics simulation program

solutions were tried. The model curves were made using a logarithmic curve-fit algorithm, which most closely matched the central tendencies of the tracking accuracy data sets.

Models of important flight parameters were also constructed from data gathered during tracking simulation data runs. After viewing the graphical representations of pitch, pitch rate, and roll plotted against range, it was determined that these parameters could be modeled through simple statistical descriptions. The mean and standard deviation were determined for these parameters for use in the ballistics simulation. Table VI presents these values.

The gun tracking simulation effort provided valuable insight into the problem of achieving satisfactory air-to-air gunnery accuracy. The simulation confirmed the difficulty most pilots have in accurately estimating lead and elevation angles during gun tracking, and was invaluable in developing the ballistics simulation models.

Table VI: Statistical Description of Important Flight Parameter Data.

Gun System Model	Statistical Measure	Pitch (deg)	Pitch Rate ($^{\circ}$ /sec)	Roll (deg)
1	Mean	2.02	0.74	5.11
	Standard Deviation	1.74	0.89	7.54
2	Mean	2.41	0.88	5.38
	Standard Deviation	2.13	1.04	4.96
3	Mean	2.57	0.90	5.29
	Standard Deviation	2.30	1.00	4.63

Notes: 1) Configuration 1 simulated a fixed sighting reticle with all ballistic compensation being made at the turreted gun. Tracking method was to maintain the reticle on the center of the target.

2) Configuration 2 simulated a fixed sighting reticle with all ballistic compensation being made by the operator. Tracking method was to position the reticle to account for target motion, ballistic factors, and own aircraft state and rates.

3) Configuration 3 simulated a CCIP sighting reticle which moved to provide aiming cuing for own aircraft states and rate compensation. Ballistic compensation and compensation for target motion was made by the operator. Tracking method was to position the reticle to account for target motion and ballistics factors.

CHAPTER 5

BALLISTICS SIMULATION

The ballistics simulation developed in support of this study was designed to be simple in concept and operation. It made extensive use of the code from the gun tracking simulation whenever possible and was constructed in a modular fashion so that a subroutine or function call could be used by several parts of the program.

To reduce the amount of analysis required to support the development of the P_H values, the Monte Carlo method of simulation was chosen. The Monte Carlo simulation technique involves the development of a mathematical representation of a system, and the use of random variables in that model. Shapiro and Gross (1981) list the following steps for a simulation program built around the Monte Carlo gaming concept.

- Determine the functional and/or mathematical model that relates system output to its elements or inputs.
- Assign to each design or input variable a probability distribution.
- Generate random values for each of the input variables and using these with the functional model compute the value of the output.
- Repeat the above step many times to get a representative system output data set.
- Summarize the data set.

The first step was primarily accomplished during the construction of the gun tracking program, and during the data analysis and model construction portion of the gun tracking simulation effort. Table VII outlines the models developed and the distributions selected for each.

With the mathematical models in hand, the next step was to ascertain the desired data output. Four areas were chosen for examination. The first was a direct comparison of the P_H values of the three gun system models for a set of range increments. The second was the creation of graphical dispersion diagrams of the three system models at significant range points. The third was an examination of the effects of changing target size on the first gun system model, the fixed reticle/gun compensation model. The final area was an examination of the effect of not compensating for various ballistic factor when using the first gun system model.

The basic operating logic of the program involved creation of pseudo-random variables, inputting them into different mathematical models, generating a simulated shot, and determining if the shot would have hit the target. This was repeated many times with a count kept of the number of hits and the total number of iterations. The P_H values presented are merely the number of recorded hits divided by the number of total iterations.

The requirement to generate pseudo-random variables was more difficult than originally expected. Visual Basic® can only generate pseudo-random variables in a uniform distribution, so a subroutine converting this uniform distribution to a normal distribution had to be added to the program. The gaussian number generator was based on the polar form of the Box-Muller transform, with sample code obtained from Carter (1996).

Table VII:

Ballistics Simulation Program Models.

Model	Distribution	Remarks
Gun Dispersion	Normal	Gun dispersion model based on 80% of rounds impacting in a 7.5 mil circle.
Tracking Accuracy	Logarithmic/ Normal	Values based on a logarithmic math model. Values vary normally about the model within computed error bands.
Pitch	Normal	Varies about a mean computed during the gun tracking simulation effort with a standard deviation computed from the simulation data.
Pitch Rate	Normal	Varies about a mean computed during the gun tracking simulation effort with a standard deviation computed from the simulation data.
Roll	Normal	Varies about a mean computed during the gun tracking simulation effort with a standard deviation computed from the simulation data.
Speed	Uniform	Varies uniformly from 90 to 150 knots.
Turn Rate	N/A	Computed from Roll and Speed values. Algorithm from gun tracking simulation.
Projectile TOF	N/A	Algorithm from gun tracking simulation.
Projectile Drop	N/A	Algorithm from gun tracking simulation.
Acute Angle Effect	N/A	Algorithm from gun tracking simulation.

The program was coded in Microsoft Visual Basic®. The computer code is presented in Appendix E. The data files were written during run time to comma delimited files in computer memory. The files were downloaded to computer floppy disks, and the data were reduced using Microsoft Excel® and Jantel Scientific SigmaPlot®.

Because of the simplicity of the program, there is a limit to the amount of direct relationship that can be made from the computer's data output. There was no direct convergence testing, and the effects on the number of iterations run on the accuracy of the P_H were examined by making several trial runs with a varying number of iterations until apparent convergence was achieved. In this fashion, the P_H values were considered to have converged to no more than 1 percentage point. Each module that produced P_H data ran for 10,000 iterations. Computational run times exceed ten minutes on a 75 MHZ Intel Pentium® computer.

The quality of the simulation was also limited by some of the assumptions made during the gun tracking simulation efforts. Of particular concern is the assumption of constant power settings and null sideslip and fuselage angle-of-attack. These factors can be expected to provide significant scatter during actual air-to-air firing. Still, the ability to quantify *relative* performance of the different gun systems, and the *relative* effects of target size and feedback loops on the first gun system model provided some interesting results.

DATA ANALYSIS AND RESULTS

Comparison of Gun System Model Performance

The P_H performance of the three gun system models was computed at 500 foot range increments from 6500 feet to 500 feet. The computations were made based on pseudo-random variable input into the dispersion and tracking accuracy models. The results are presented in Table VIII. The first gun system, the fixed reticle/gun compensation model, significantly outperformed the other two systems at ranges outside of 500 feet. This should not be surprising, as the extreme difficulty experienced by the test operators when attempting to position the reticle on the desired aim point using the second and third modeled gun systems, assured that the tracking accuracy model for those configurations would generate large miss distances. It was not until the relative target size became large that the second and third models were able to generate an acceptable P_H . This mirrors the real life examples that were discussed earlier in this report and repeats the "stick your nose in the cockpit" approach used so often in the past. What is surprising are the very good P_H values found for the first gun system model well outside of what is currently considered to be the maximum effective air-to-air gun range of the modeled projectile. These figures should be considered optimistic, yet worthy of more intensive examination.

Dispersion Diagrams

To support the previous module, as well as to provide another way to examine relative performance of the modeled gun systems, dispersion diagrams were generated for

Table VIII: Comparison of Computed P_H Values for Three Simulated Gun Systems.

Range	Configuration 1	Configuration 2	Configuration 3
6500	.40	.02	.02
6000	.42	.02	.03
5500	.45	.03	.03
5000	.47	.03	.03
4500	.50	.05	.05
4000	.53	.06	.06
3500	.58	.08	.08
3000	.65	.11	.12
2500	.72	.18	.18
2000	.81	.26	.28
1500	.90	.43	.44
1000	.98	.77	.76
500	1.00	.99	.99
Average	.65	.23	.24

Notes: 1) Configuration 1 simulated a fixed sighting reticle with all ballistic compensation being made at the turreted gun. Tracking method was to maintain the reticle on the center of the target.

2) Configuration 2 simulated a fixed sighting reticle with all ballistic compensation being made by the operator. Tracking method was to position the reticle to account for target motion, ballistic factors, and own aircraft state and rates.

3) Configuration 3 simulated a CCIP sighting reticle which moved to provide aiming cuing for own aircraft states and rate compensation. Ballistic compensation and compensation for target motion was made by the operator. Tracking method was to position the reticle to account for target motion and ballistics factors.

4) The simulation computed the P_H values for firings against a simulated target 20 feet in diameter.

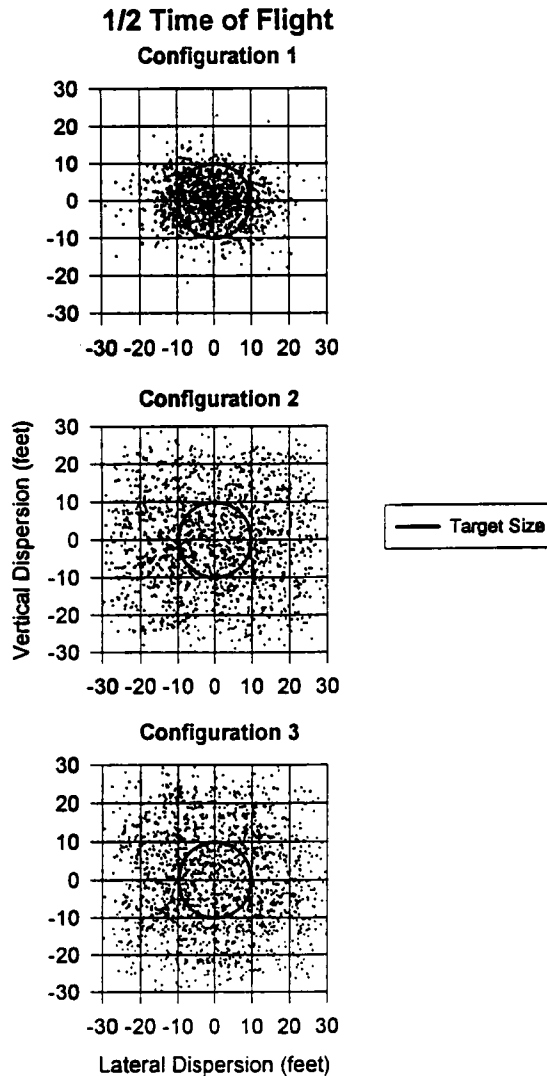
1/2, 1, and 2 TOF of the modeled projectile. These diagrams present 2000 pseudo-random shots with a 20 foot target reference circle displayed for comparison. Fall of each pseudo-random shot was generated with the gun dispersion and tracking accuracy models. The tight clustering of the first gun system model, and the weak grouping of the other two models bears out the tabular data presented in Table VIII. The diagrams are presented in Figures 15a, 15b, and 15c respectively. There is little apparent difference in the data between the fixed reticle/fixed gun system and the CCIP reticle system. This may be a result of the quality of the simulation, so the question as to their relative strengths must go unanswered.

Effects of Target Size

The effects of target size on the performance of the first gun system model, the fixed reticle/gun compensation system, were examined by varying the target size and computing P_H values for 500 foot range increments from 6500 feet to 500 feet. The results are non-linear, and validate the wisdom in picking as small an aircraft as possible for one's self, and as large an opponent as one can find. The results are presented in Table IX.

Effects of Deleting Feedback Loops

The effect of not compensating for different ballistic effects was examined by computing a base P_H for the first modeled gun system and then adding in error values based on the system feedback that was being deleted. These pseudo-random error quantities were computed in the distributions determined from the gun tracking simulation, and also in a normal distribution across their maximum allowable range. This second analysis was



(a) Range Equal to $\frac{1}{2}$ TOF (1433 feet)

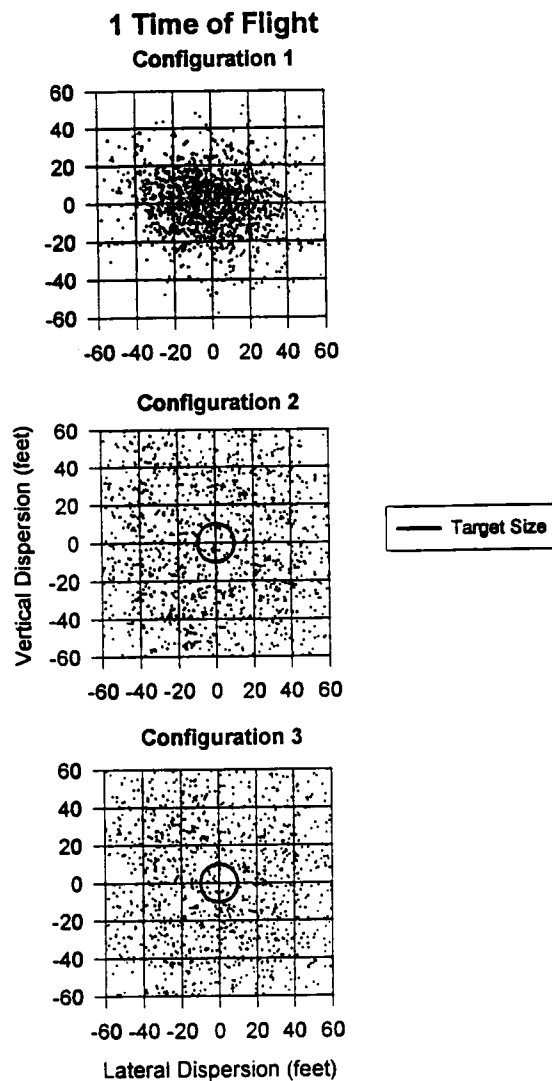
Figure 15: Dispersion Diagram.

Notes: 1) Configuration 1 simulated a fixed sighting reticle with all ballistic compensation being made at the turreted gun.

2) Configuration 2 simulated a fixed sighting reticle with all ballistic compensation being made by the operator.

3) Configuration 3 simulated a CCIP sighting reticle which moved to provide aiming cuing for own aircraft states and rate compensation. Ballistic compensation and compensation for target motion was made by the operator.

4) The simulation computed the P_H values for firings against a simulated target 20 feet in diameter.

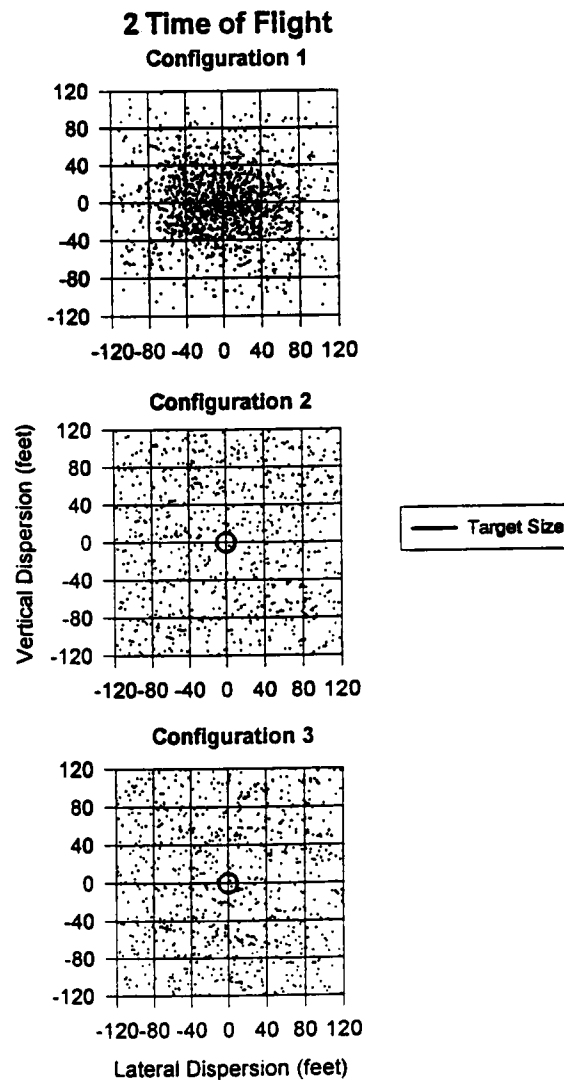


(b) Range Equal to 1 TOF (2725 feet)

Figure 15:

(Continued).

- Notes: 1) Configuration 1 simulated a fixed sighting reticle with all ballistic compensation being made at the turreted gun.
- 2) Configuration 2 simulated a fixed sighting reticle with all ballistic compensation being made by the operator.
- 3) Configuration 3 simulated a CCIP sighting reticle which moved to provide aiming cuing for own aircraft states and rate compensation. Ballistic compensation and compensation for target motion was made by the operator.
- 4) The simulation computed the P_H values for firings against a simulated target 20 feet in diameter.



(c) Range Equal to 2 TOF (4453 feet)

Figure 15: (Continued).

Notes: 1) Configuration 1 simulated a fixed sighting reticle with all ballistic compensation being made at the turreted gun.

2) Configuration 2 simulated a fixed sighting reticle with all ballistic compensation being made by the operator.

3) Configuration 3 simulated a CCIP sighting reticle which moved to provide aiming cuing for own aircraft states and rate compensation. Ballistic compensation and compensation for target motion was made by the operator.

4) The simulation computed the P_H values for firings against a simulated target 20 feet in diameter.

Table IX:

Effects of Target Size on P_H Values.

Range	Target Diameter (feet)				
	5	10	20	40	80
6500	.03	.13	.40	.73	.96
6000	.03	.13	.42	.75	.97
5500	.04	.15	.45	.79	.97
5000	.04	.16	.47	.82	.98
4500	.05	.18	.50	.85	.99
4000	.04	.19	.53	.89	.99
3500	.05	.21	.58	.91	.99
3000	.06	.23	.65	.95	1.00
2500	.08	.29	.72	.97	1.00
2000	.09	.35	.81	.99	1.00
1500	.14	.47	.90	.99	1.00
1000	.23	.65	.98	1.00	1.00
500	.53	.95	1.00	1.00	1.00

Notes: 1) Program simulated a fixed sighting reticle with all ballistic compensation being made at the turreted gun.

conducted because of the rather low mean and standard deviation values generated for these variables during the gun tracking simulation runs. The results are presented in Table X. Note the importance of compensating for roll to obtain effective results at long ranges. In a few cases, it will be noted that a slightly higher P_H was achieved with a feedback loop deleted than with it engaged. These minor variations are the result of normal scatter in the data and should not be considered significant.

It is interesting to note that even though deleting some of the feedback loops caused significant reductions in P_H , the resulting values were still significantly better than those for the system implementations where the operator was required to mentally estimate and manually compensate for a lead point. This shows the importance of operator tracking accuracy as a factor in overall system performance.

Table X: Effect of Deleting Feedback Loops.

Range	Feedback Loop Deleted From Ballistics Computation											
	None		Speed		Pitch		Roll		Pitch Rate		Turn Rate	
6500	.40	.40	.30	.31	.40	.21	.23	.06	.40	.20	.40	.39
6000	.42	.42	.36	.37	.43	.27	.28	.09	.43	.22	.43	.42
5500	.45	.45	.41	.41	.45	.33	.35	.12	.45	.27	.45	.44
5000	.47	.47	.45	.45	.48	.40	.41	.18	.48	.31	.47	.48
4500	.50	.50	.49	.48	.51	.45	.47	.26	.51	.36	.51	.50
4000	.53	.53	.53	.52	.55	.51	.53	.36	.55	.42	.56	.54
3500	.58	.58	.58	.58	.59	.58	.58	.48	.59	.49	.59	.59
3000	.65	.65	.63	.64	.64	.60	.64	.57	.64	.57	.64	.67
2500	.72	.72	.72	.72	.71	.70	.72	.70	.72	.67	.72	.72
2000	.81	.81	.81	.80	.81	.80	.81	.80	.81	.78	.81	.80
1500	.90	.90	.90	.90	.90	.90	.90	.90	.90	.89	.90	.90
1000	.98	.98	.98	.98	.98	.98	.98	.98	.98	.98	.98	.98
500	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Notes: 1) The first P_H in each block represents the value computed for the distribution indicated by the data from the gun tracking program. The second P_H in each block represents the value computed at a normal distribution over the maximum allowable range for the flight parameter.

2) Program simulated a fixed sighting reticle with all ballistic compensation being made at the turreted gun.

3) The program simulated firing at a target 20 feet in diameter.

4) Minor variations in P_H may be disregarded as normal scatter from the Monte Carlo generation technique.

CHAPTER 6

CONCLUSIONS

The numerical results obtained from the use of the simulation programs developed for this study are not in and of themselves significant. The models developed, while more than adequate for the purposes of this *demonstration of a method*, are too rudimentary to make exact decisions concerning the design and mechanization of a helicopter LCOS and gun system. However, the qualitative information obtained is valid and should be considered when contemplating such a developmental program. The most important points are:

- A system which requires the operator to mentally estimate lead angle and elevation, and then manually adjust aim point to make these adjustments will suffer from poor tracking accuracy. The most significant degradations in P_H encountered during this effort were due to poor operator tracking accuracy.
- The optimum tracking method is to allow the operator to track the center of the target with the reticle. This tracking method insures the highest tracking accuracy.
- Some feedback loops are more important than others. A simulation effort such as the one demonstrated here can enable a system designer to balance cost of integration

against the benefit gained from incorporating compensation for the parameter in question.

- When designing an LCOS and gun system, it is important to quantify and use operator tracking accuracy as a system variable, as it appears that the pilot's ability to track the target properly outweighs all other system parameters during air-to-air gunnery.

EXTRAPOLATION OF THE METHOD

This method of using simulation to optimize an LCOS and gun system for an attack helicopter is highly dependent on the quality of the simulation models. The optimum method of simulation would be to instrument an actual aircraft and record real world data, as was done with the McDonnell Douglas INTAAW program. Few program managers can afford the luxury of accomplishing a full scale integration of a range of possible implementations and then comparing them to find the most effective mechanization. By using a high fidelity simulator however, both risk and cost could be mitigated and a series of systems could be prototyped.

Perhaps an even more cost effective approach would be to perform a series of software based, low fidelity simulations before proceeding to the higher fidelity flight simulators, as the latter still requires extensive hardware modification and integration. Various LCOS mechanizations could be simulated and evaluated, in much the same way as

was done in this study, and the least effective configurations eliminated prior to an expensive integration effort. By taking this building block approach, an effective helicopter air-to-air gun system could be designed and fielded for the lowest possible cost.

BIBLIOGRAPHY

BIBLIOGRAPHY

This bibliography lists the references consulted during the course of this study, but not directly cited in the body of the report. References directly cited in the body are listed in the section which follows.

Cao, Yihua. *Kinematic Inverse Solution of Helicopter Turreted Gun*. Nanjing, P. R. China: East China Institute of Technology, 1992.

Department of the Army. *Operator's Manual for Army AH-64A Helicopter*. Washington, D.C.: Department of the Army, 1984.

Friedland, Bernard. *Control System Design, An Introduction to State-Space Methods*. New York, New York: McGraw-Hill Inc., 1986.

Grossman, John, ed. *The Chicago Manual of Style*, 14th ed. Chicago, Illinois: University of Chicago Press, 1993.

Gu, You-Liang, Loh, Robert N. K., Coleman, Norman and Lin, Ching-Fang. *Control of Weapons Pointing Systems Based on Robotic Formulation*. Evanston, Illinois: American Automatic Control Council, 1992.

Guide to the Preparation of Theses and Dissertations, 7th ed. Knoxville, Tennessee: The Graduate School, The University of Tennessee, 1991.

Gurewicz, Nathan, and Gurewicz Ori. *Teach Yourself Visual Basic® in 21 Days*. Indianapolis, Indiana: Sams Publishing, 1995.

Howard, Chris. *Programming Games for Beginners, Visual Basic® for Windows® for Fun and Profit*. Indianapolis, Indiana: Sams Publishing, 1993.

Morton, Mark H. And Toossi, Mostafa. *Dispersion Estimation by Monte Carlo Simulation for an Elastic Gun Tube/Projectile System*. Mesa, Arizona: McDonnell Douglas Helicopter Company, 1991.

Peters, J., Hashemo-Kia, M. and Toossi, M.. *Analytical Simulation of a Helicopter Mounted Turreted Gun System with Active Control*. Mesa, Arizona: McDonnell Douglas Helicopter Company, 1992.

Visual Basic® Programming System for Windows® Ver. 3.0. Microsoft Corporation, Redmond, Washington, 1993.

Wang, Wallace. *Visual Basic® 3 for Dummies®*. Foster City, California: IDG Books, 1994.

REFERENCES CITED

REFERENCES CITED

- Anderson, D.A. Capt. USMC. 1996. Marine Aviation Weapons and Tactics Squadron One (MAWTS-1) Instructor Pilot. Telephone conversation with the author, 3 June.
- Barnes, Scott. 1991a. Hind vs Cobra. *Defence Helicopter* 10 (February-March): 40-43.
- Barnes, Scott. 1991b. Easy Kills? *Defence Helicopter* 10 (August-September): 14-16, 48.
- Buresh, Jon A., Parlier, Charles A. and Wilson, Edward. 1988. *Air-To-Air Combat Development of the AH-64A Apache*, Mesa, Arizona: McDonnell Douglas Helicopter Company.
- Carter, Everett. 1996. Generating Gaussian Random Numbers. Monterey, California, 1996 [cited 25 June 1996]. Available from <http://www.taygeta.com/random/gaussian.html>; INTERNET.
- Crescent. 1990. QuickPac Scientific for Windows®. Ridgefield, Connecticut: Crescent Software.
- Domelen, Dave. 1996. The Coriolis effect. Columbus, Ohio, 1996 [cited 2 June 1996]. Available from <http://www.physics.ohio-state.edu/~dvandon/edu/coriolis.html>; INTERNET.
- DON (Department of the Navy). 1991. *Principles of Operation, AH-1W Armament System*. Washington, D.C.: Department of the Navy.
- . 1992. *AH-1W Flight Manual*. Washington, D.C.: Department of the Navy.
- . 1993a. *AH-1 Tactical Manual*. Washington, D.C.: Department of the Navy.
- . 1993b. *20mm Point and Suppressive Fire System A/A49E-7(V4) Technical Manual*. Washington, D.C.: Department of the Navy.
- Frieden, David R. LCDR USN, ed. 1985. *Principles of Naval Weapons Systems*. Annapolis, Maryland: Naval Institute Press.
- Fulmer, Richard D. 1987. *Navy/Marine Corps 20mm Improved Ammunition*. China Lake, California: Naval Weapons Center.
- Haselwood, Scott and Monroe, Ronald. 1973. *GDC/EOSS: Real-Time Visual and Motion Simulators for Evaluation of Fire Control and Electro-Optical Guidance Systems*, Orlando, Florida: Martin Marietta Corporation.

- Lambert, Mark, ed. 1995. *Jane's All the World's Aircraft 1994-1995*. Alexandria, Virginia: Jane's Information Group.
- Lennox, Duncan, ed. 1995. *Jane's Air Launched Weapons 1994-1995*. Alexandria, Virginia: Jane's Information Group.
- GEC (General Electric Company). 1979. *Required Hardware Changes to the 20mm AH-1S Turret System to Provide 1500 Shots/Minute Firing Rate*. Burlington, Vermont: General Electric Company.
- Mahaffey, M. D. Capt. USMC. [1984?]. Air-to-air gunnery for the AH-1 Cobra, Marine Aviation Weapons and Tactics Squadron One (MAWTS-1) Instructional Text. Yuma, Arizona. Duplicated.
- MDA (McDonnell Douglas Aerospace). n.d.. F-18 Software Specification PPS(91C) Rev A, St. Louis, Missouri: McDonnell Douglas Aerospace. Duplicated.
- Michell, Simon, ed. 1994. *Jane's Civil and Military Aircraft Upgrades 1994-1995*, Alexandria, Virginia: Jane's Information Group.
- O'Reilly, Peter J. 1987. *Marinized Apache, An Engineering Challenge*, Mesa, Arizona: McDonnell Douglas Helicopter Company.
- Osder, Stephen and Mossman, Davis. 1992. *Avionics Architecture for Performing Fire Control Research in AH-64 Aircraft*, Mesa, Arizona: McDonnell Douglas Helicopter Company.
- Shapiro, Samuel S. and Gross, Alan J. 1981. *Statistical Modeling Techniques*, New York, New York: Marcel Dekker, Inc.
- Shaw, Robert L. 1985. *Fighter Combat, Tactics and Maneuvering*, Annapolis, Maryland: Naval Institute Press.
- Spick, Mike. 1988. *The Ace Factor, Air Combat & the Role of Situational Awareness*, Annapolis: Naval Institute Press.
- SURVIAC (Survivability/ Vulnerability Information Analysis Center). 1995. MODEL SPOTLIGHT: RADGUNS 1.9 - Radar Directed Gun System Simulation. *SURVIAC Bulletin* (July) 1-2.
- THE (The Hunter's Encyclopedia). 1962. *Rifle Shooting*. New York, New York: Collier Books.

Walters, G. Maj. USMC, Brubacher W. 1992. AH-1W Night Targeting System Theory and Operational Function, Naval Air Warfare Center Instructional Text. Patuxent River, Maryland. Duplicated.

Wilson, Edward. 1989. *Target Acquisition in Helicopter Air-to-Air Combat*, Mesa, Arizona: McDonnell Douglas Helicopter Company.

Wilson, Edward. 1996, McDonnell Douglas Helicopter Company, Senior Experimental Test Pilot, Telephone conversation with the author, April 19.

APPENDICES

APPENDIX A

HELICOPTER GUN SYSTEM ACCURACY

Delivery of accurate air delivered gunfire is complicated by the three-dimensional motion of the gun equipped aircraft. The relatively simple method of directing gunfire from a stationary ground-based gun against a stationary target by adjusting gun azimuth for direction and bore elevation for gun-to-target range (and fine tuning both for wind effect) becomes drastically more difficult when the gun system is attached to an aircraft maneuvering aggressively in three dimensions to achieve a firing solution. If the target is maneuvering as well, consistently accurate firing solution computation will probably be beyond the pilot's mental abilities.

There are numerous weapons systems characteristics and physical phenomenon which act in conjunction to effect helicopter gun system accuracy. The totality of these factors could be the subject of a separate thesis and a full discussion is beyond the scope of this effort. A basic understanding of the subject is necessary however, to comprehend the magnitude of problems associated with air-to-air gunnery as it pertains to the attack helicopter, and therefore a brief discussion is warranted.

This appendix addresses several areas of general interest to designers of helicopter gun systems; weapons systems characteristics, interior ballistics, exterior ballistics, firing platform effects, and range and target motion effects. Much is left unexplored, particularly the very interesting area of terminal ballistics.

WEAPONS SYSTEMS CHARACTERISTICS

Sighting Systems

Sighting system characteristics have a significant effect on the total system accuracy of the gun. Sighting systems introduce a level of error through their own installation. This error is normally thought of as "boresight error", or the difference between the visual line-of-sight (LOS) of the sight and the LOS transmitted to the gun system. This error is usually measured in milliradians ($1^\circ = 17.5$ milliradians) and commonly referred to in "mils". (Shaw 1985). Three different sighting systems are in common use for helicopter gun systems.

The most accurate is a stabilized turreted sight which is normally used for both delivery of precision guided ATGM's and for gun direction. These sights may be nose mounted, or mounted on an extension of the helicopters main rotor mast (mast mounted). Examples are the AH-64 "Apache" Target Acquisition Designation Sight (TADS) and the AH-1W "SuperCobra" Night Target System (NTS) Telescopic Sight Unit (TSU). These types of sights provide a stabilized sight with a variety of sensor systems (camera, direct-view optics [DVO], and forward-looking infrared [FLIR]) operating at several selectable magnifications. The pilot interface is normally either a multi-function display (MFD) or an optics train. In some mechanizations, a laser rangefinder/designator is integrated into the sight which can provide accurate range to target information for gun direction computations. These sights can provide an air-to-air and air-to-ground capability to the maximum effective range of the helicopter's gun. However, sight slew rates and the normally narrow field-of-

view can limit the sight's utility in air-to-air combat at medium to close-in ranges. Typical boresight errors are less than one mil.

A fixed gun/rocket sight or more advanced Heads-Up Display (HUD) may be incorporated, normally in the rear cockpit of tandem seat helicopter. This sight configuration provides a sighting reference that is as stable and vibration free as the basic airframe and which has the advantage of being boresighted to the aircraft's ballistics reference line (BRL) which provides an accurate delivery capability for fixed-forward firing guns and wing mounted forward firing missiles and rockets. Typical boresight errors for fixed sights are one to two mils. Sighting with a fixed sight generally provides a wide field-of-view (enhancing pilot situational awareness) and a familiar tracking technique (aircraft pointing). The pilot will naturally tend to concentrate his attention primarily in the direction of his flight path, so providing a sight which resides in the middle of this area of regard reduces the pilot's mental workload. The fixed sight can provide a good air-to-air and air-to-ground aiming capability at close to middle ranges, but is limited at longer ranges by the lack of magnification which can inhibit accurate reticle placement.

Finally, a helmet mounted sighting reticle such as the "SuperCobra" Helmet Sight System (HSS), or preferably a sighting reticle integrated in a more capable helmet mounted display (HMD), such as the "Apache" Integrated Helmet and Display Sight System (IHADSS), combined with a helmet/head tracking system may be used. These systems automatically slave the aircraft turreted gun system to the pilot's LOS and provide a rapid aiming capability. Pointing accuracy of these helmet sighting systems is limited by the system's ability to resolve the pilot's LOS and the primary factor in this equation is the

ability of the head tracker to accurately track head motion in the six axes of freedom (displacement: up/down, left/right, back/front; rotation: up/down, left/right, and side-to-side tilt). Helmet mounted sighting system boresight accuracy varies depending on the type of helmet/head tracker used, but can be as much as five to ten mils for mechanical trackers and as much as ten to fifteen mils for optical/magnetic trackers. The generally loose fit of most pilot's helmets also degrades system accuracy during maneuvering flight due to helmet slippage. Helmet mounted sights also suffer in performance from a lack of magnification which inhibits accurate reticle placement and a lack of stabilization. A pilot's head moves in flight more than he realizes (this movement being masked by the excellent natural stabilization of the eye) and unwanted head motion caused by aircraft vibration and vigorous aircraft maneuvering can significantly effect helmet sighting system accuracy. Finally, most pilots find using their head as an aiming device while flying the aircraft to be very unnatural, as it keeps them from maintaining a good scan (ie. they are unable to look around for obstacles or other aircraft) and often times requires that they not look in the direction of their flight path which can be very unsettling, especially during low altitude flight. This generally makes the helmet sighting system of more use to the copilot/gunner than the pilot at the controls. The numerous limitations associated with the helmet mounted sight systems make them most appropriate during very close-in maneuvering air-to-air combat, where fast pointing is required and system accuracy is less important due to the larger relative size of the target.

Gun Systems

Gun systems installed on attack helicopters generally fall into two categories. Fixed-forward firing guns can be mounted on the aircraft fuselage as is done in some versions of the Russian Mi-24 "Hind" or in wing mounted gun pods. Guns may also be mounted in flexible firing mounts such as the chin mounted turrets found on the various "Cobra" and "Apache" variants which can provide a wide range of gun travel ($\pm 110^\circ$ in azimuth and $\pm 30^\circ$ to 60° in elevation), or in limited travel fuselage mounts such as on the Russian Ka-50 "Hokum" which can be depressed 30° and traversed $\pm 6^\circ$ (Lambert 1995).

Gun system accuracy is normally considered in terms of two factors; firing accuracy measured in mils (referred to as mil dispersion) and pointing accuracy measured in mils or degrees, which is a function of turret drive accuracy for flexible guns and boresight accuracy for fixed-forward firing guns. Mil dispersion can be effected by bore condition, gun construction tolerances, mount construction tolerances, aircraft vibrations transmitted to the gun, gun mount and airframe flex during firing, and variations in individual projectile performance with typical accuracies falling between five and ten mils in air-to-air gun applications (DON 1993a). Fixed-forward firing pointing or boresight error can be as low as one mil, and flexible gun accuracy generally falls between a few mils to 2° .

The maximum range that a projectile can be accurately fired and still reach the target with sufficient kinetic energy to have the desired effect is known as the maximum effective range. The limiting factor for the particular gun system may be the accuracy of the sighting system or the ballistic characteristics of the projectile. A common rule of thumb for determining the maximum effective range of a gun against air-to-ground targets is to multiply

the diameter of the projectile measured in millimeters by 100 and then read the range in meters. For example, a 20mm shell would be considered effective out to 2000 meters (20mm x 100 = 2000 meters). (DON 1993a). In most current helicopter gun systems, the maximum effective range of a gun projectile against an air-to-air target is based on two factors; its gravity drop over range and its variation in speed over range. This maximum effective air-to-air range is described in terms of the distance that the round travels in a second, or its time of flight (TOF). Aircraft without automatic lead and elevation computation systems known as lead computing sights (LCOS), have a maximum effective air-to-air range of $\frac{1}{2}$ to 1 TOF. (DON 1993a).

Gun lethality is a factor of the gun system accuracy, firing rate, and projectile characteristics. These factors are intertwined and normally cannot be considered separately. Consider that most aircraft can readily absorb numerous hits from small arms (.30 cal and smaller) in non-critical areas, while a single hit from a large caliber cannon (75mm or larger) firing a high explosive projectile should be sufficient to down all but the most heavily armored helicopter regardless of the point of impact. Higher firing rates are desirable, in that they can produce more hits during the shorter tracking times associated with air-to-air combat in the close-in arena, but if overall gun/sight system accuracy is low the pilot may find that he is merely missing at a higher rate. (DON 1993a). Firing rates are often intentionally reduced in flexible gun installations due to the helicopter's limits on tail rotor power and to reduce induced airframe vibration. The high recoil associated with rapid firing cannons mounted in nose turrets can overcome the helicopter's ability to maintain a constant heading during off-axis hover firing.

INTERIOR BALLISTICS

Interior ballistics deals with the factors effecting the motion of a projectile prior to leaving the launching device, which for our purposes is the gun barrel. Another definition of interior ballistics would be that it deals with the factors arising from the interaction of the gun barrel and chamber, the burning propellant, and the projectile. Variations in projectile performance caused by interior ballistics effects are of much significance to overall gun system performance, but may be difficult to compensate for during air-to-air combat. Gun firing solutions in air-to-air applications are normally based upon a standard or nominal projectile performance model. Normal variations from the standard model make up part of the system error considered in gun mil dispersion (see above). However, significant deviation from the projectile performance model due to the effects of interior ballistics factors can render gun direction solutions useless. Muzzle velocity is the result of interior ballistics and is of primary interest in air-to-air gun applications. (DON 1993a).

Gun Bore Condition

The condition of the gun bore will have a significant effect on air-to-air gun system performance in two regards. First, excessive gun bore erosion and wear can allow propellant gases to escape around the projectile during firing, reducing projectile muzzle velocity. This effect is commonly accounted for when firing large naval guns or artillery pieces, but is not normally considered when performing ballistics computations for air-to-air gunnery. Secondly, if the condition of the gun bore is bad enough, the projectile may either be subject

to an undesirable lateral force from escaping gas as the projectile leaves the end of the barrel, or it may even tilt slightly in the bore. This condition can lead to excessive and unaccounted gyroscopic effects much like the ones discussed in the following section on exterior ballistics. (Frieden 1985).

Propellant Performance

All current air-to-air guns use fixed ammunition, where a predetermined amount of propellant is contained in the shell casing by the projectile. Variation in propellant performance will cause a commensurate variation in chamber pressure, which will effect the muzzle velocity of the projectile. There is some variation of propellant performance from shell to shell, either from variation in the amount of propellant loaded or from variation in burn performance between different manufacturing lots of propellant powder. This variation can be a major component in the variation in muzzle velocity caused by interior ballistics effects. Propellant performance can also be effected by moisture absorbed during storage, and by propellant temperature at the time of ignition. Of these effects, only propellant temperature can be accounted for in gun direction computations and then only if a temperature sensor is located in the ammunition compartment of the aircraft. (DON 1993a).

Projectile Weight

Projectile weight is also subject to variations induced during the manufacturing process. Manufacturing variation is not accounted for in air-to-air applications. Projectile weight can also vary predictably from variant to variant within a cartridge family. For

example, within the PGU family of 20mm cartridges, the nominal weights of the various projectiles varies from 1580 grains for the Semi-armor Piercing High Explosive (SAPHEI) projectile to 1554 grains for the Target Practice-Tracer (TP-T) projectile. (Fulmer 1987). Some families of projectiles are designed so that this variation in projectile weight is accounted for in the propellant load, yielding a nearly constant muzzle velocity.

Muzzle Velocity

Muzzle velocity isn't a factor of interior ballistics *per se*, but is rather the result of ammunition design (propellant charge and projectile weight), gun characteristics (barrel length, tube condition, and rifling design), and ambient conditions (temperature, pressure, and humidity). To a great extent, muzzle velocity determines the maximum range and ballistic trajectory of the projectile. (DON 1993a). A higher muzzle velocity is desired due to the associated increase in range, decrease in TOF, and reduction in ballistic arc (see below). Gun systems are normally approved to fire several different families of cartridges, each of which has different muzzle velocity. These various muzzle velocities must be considered when computing firing solutions. (DON 1993a).

EXTERIOR BALLISTICS

Exterior ballistics deals with the factors which effect the motion of the projectile as it moves along its trajectory. Most of these effects are quantifiable and should be accounted for in ballistics computations.

Air Resistance (Drag)

Since gun projectiles by design do not intentionally generate lift, the majority of the drag effect on a projectile is the result of friction between the air and the projectile, or parasitic drag. A factor of velocity, projectile cross section, projectile shape and center of gravity, projectile angle-of-attack, and density altitude; drag is the counter force to muzzle velocity, which limits the maximum range of the projectile. The change in projectile velocity over range caused by air resistance is the primary limitation of maximum effective air-to-air gun range in applications where the pilot must mentally compute elevation and lead angle. (DON 1993a). Maximum effective range may be extended if this effect is accounted for in the firing solution.

Projectile Drift (Horizontal Plane Gyroscopic Effects)

The rifling in US guns imparts a clockwise spin to projectiles (when viewed from the rear). This spin creates a gyroscopic stabilization which tends to keep the axis of rotation pointed in a constant direction. As the projectile travels through its trajectory, its direction of motion shifts tangent to the ballistic curve. As the angle between the axis of rotation and the direction of motion increases, the center of pressure from air resistance shifts to the underside of the projectile, pushing the nose of the projectile up. This upward force from air resistance causes a gyroscopic precession to the right (when viewed from above). This precession causes a drift to the right after firing which increases in magnitude over range. (Frieden 1985; DON 1993a). Projectile drift requires an aiming correction to the left proportional to the range to the target.

Projectile Jump (Vertical Plane Gyroscopic Effect)

When a projectile fired in a direction other than directly into the relative wind, the crosswind causes an initial yaw which causes a proportional jump in the flight path of the projectile due to gyroscopic precession (see the above discussion of projectile drift). (DON 1993a). Firing to the right causes a downward jump and to the left an upward jump. Projectile jump is a factor of airspeed and gun deflection and it requires an aiming correction opposite of the effect. This effect may not be perfectly vertical, as vertical wind components from rotor downwash in a hover and during climbing or descending flight may cause the same effect.

Magnus Effect

Magnus effect refers to the lift generated by a spinning projectile when presented with a crosswind, caused by the differences in relative wind at the top and the bottom of the projectile. Projectiles presented with a left crosswind experience upward lift and with a right crosswind, downward lift. (Frieden 1985). Magnus effect is a factor of projectile rotational velocity, crosswind speed, and time of flight (range to target) and requires an aiming correction opposite of the effect. This effect may not be perfectly lateral, as vertical wind components from rotor downwash in a hover and during climbing or descending flight may cause the same effect.

Gravity Drop

Gravity drop is the effect of gravity which causes a projectile to drop below its original direction of flight. It is a factor of time of flight and to a lesser effect vertical distance from the earth's center. Gravity drop requires an increase in gun elevation proportional to range to target. (DON 1993a).

Firing at Acute Angles

When firing at a target whose elevation differs greatly from the firing platform, a correction will need to be made in the gun's elevation angle. Firing at a large elevation or depression (more than 5° to 10°) will cause an effective change in the ballistic arc of the projectile. This change in projectile trajectory will result in the round flying high for both elevation and depression angles, as compared to the ballistic arc of a projectile fired at a target at the same target range but coaltitude with the firing platform. To visualize this phenomenon, imagine firing a weapon at a target that is 85° above the gun's horizontal reference plane, with a weapon that has a 5° elevation angle set above the line-of-sight (LOS) of the weapon sighting system. The effective elevation of the weapon barrel would be 90° and under windless conditions, the projectile would go straight up and pass over the target (ie. above the sight-to-target LOS). Examining conditions more likely to be encountered during helicopter aerial gun fire, the correction for a 30° dive would be to use a sight setting for a horizontal range equal to about 87% of the actual slant range to the target. For a 45° dive, the correction increases to a sight setting equivalent to that for a target at 71% of the actual range to target. Interestingly, the amount of correction varies for equal

degrees of elevation and depression even though the overall effect is the same. This phenomenon requires a decrease in gun elevation proportional to the elevation or depression angle to the target. (THE 1962).

Relative Wind (Wind Drift)

Wind drift is the movement of the projectile caused by the force of the relative wind acting against the projectile's exposed flat plate area. Depending on the direction of the relative wind, the drift may manifest in range variation, azimuth variation, or both. A factor of wind strength and time of flight, wind drift includes the effect of rotor downwash. This is a relatively minor effect which can nonetheless become significant over longer ranges. (DON 1993a). Wind drift requires an aiming correction upwind, if wind direction and strength can be determined.

Coriolis Effect

Coriolis effect is the result of variation in the rotational speed of the earth as a projectile moves across lines of latitude (north or south), and centripetal acceleration as the projectile moves across lines of longitude (east or west). The north/south projectile appears to curve in the direction of the earth's rotation (eastward) as it moves towards a pole, and away from the direction of the earth's rotation (westward) as it moves towards the equator. The eastbound projectile will tend to curve towards the equator, while the westbound projectile will tend to curve toward the nearest pole. (Frieden 1985). Over the short flight distances found in air-to-air gun applications, the Coriolis effect causes only a few inches of

deviation (Domelen 1996), which can normally be disregarded. The effect is very quantifiable however, and with an accurate aircraft location (from a global positioning satellite [GPS] receiver perhaps) and an accurate firing direction, it could be accounted for in ballistics computations.

FIRING PLATFORM EFFECTS

Aircraft Orientation

Aircraft orientation to the gravity vector affects the result of other gun azimuth and elevation corrections. For example, when firing while in an angle of bank, gun elevation is reduced and a portion of the gun's elevation is converted to an increase in gun azimuth in the direction of the bank angle. Likewise, gun azimuth is reduced and a portion of the gun's azimuth is converted to a change in gun elevation. (DON 1993a). Similarly, variation in aircraft pitch attitude will cause a change in the ballistic trajectory of a projectile (see the preceding discussion on firing at acute angles). (THE 1962). Unless aircraft orientation is accounted for, any deviation from straight and level flight will reduce overall system accuracy.

Aircraft Body Motion

Aircraft rate and acceleration in its six axes of motion (pitch, roll, yaw, vertical, longitudinal, and lateral) will impart a proportional motion to the projectile which will affect the projectile's trajectory. For example, a pitch up during firing will impart a vertical rate to

the projectile which, unless compensated for can cause the projectile to pass over the target. Note that longitudinal aircraft motion normally reflects aircraft airspeed, disregarding any sideslip. (DON 1993a). Aircraft body motion requires an aiming correction to counteract the effect.

RANGE AND TARGET MOTION EFFECTS

Range to Target

Increasing range to target requires an increase in gun elevation to counteract gravity drop. The accuracy of range information is one of the most important factors in accurate projectile delivery. (DON 1993a).

Range Rate (Closure Rate)

The rate of change in the aircraft to target range is known as range rate or closure rate. When combined with the initial range to target and the projectile time of flight, the range to target at projectile impact can be computed and the appropriate range correction made. In a head-on encounter with a high performance fixed-wing aircraft, range (closure) rate can exceed 900 feet per second which can cause serious range errors without appropriate compensation. (DON 1993a).

Target Angular Velocity

Target angular velocity is the prime factor in computing lead angle. It is a factor of target speed, crossing angle, range to target, and range rate. The gun must be aimed ahead of the target's flight path so that the projectile meets the target at the end of the time of flight. (Frieden 1985).

APPENDIX B

AIR-TO-AIR COMBAT PRINCIPLES

Air-to-air combat is a subject which is as much opinion, doctrine, dogma, and art as it is science. Great differences have always existed over the "right" way to conduct aerial combat, and this report will not endeavor to put forward a specific endorsement of any school of tactics. However, practitioners of air-to-air combat work within a fairly uniform set of parameters and measures when discussing the subject and it is important to be familiar with these concepts. For example, while there is much disagreement over what is more important, aircraft performance or aircraft maneuverability, most students of aerial combat will at least agree that the notions of performance and maneuverability exist and that they each play no small role in air-to-air combat.

While there are many important references concerning air-to-air combat, two found in this report's reference list are of particular interest; Robert L. Shaw's definitive work, *FIGHTER COMBAT, Tactics and Maneuvering*, and Mike Spick's treatise, *THE ACE FACTOR, Air Combat and the Role of Situational Awareness*. Both of these books are required reading at the US Marine Corps' Weapons and Tactics Instructor course.

PHASES OF AIR-TO-AIR COMBAT

Air-to-air combat is usually divided into five stages. They tend to overlap and do not always flow in the order presented (the aerial combatant may back up a step before proceeding onward or may be thrust immediately into a latter stage by being caught unawares), but most aerial engagements pass at least fleetingly through each stage. (Spick 1988).

Detection and Identification

This is the first, and perhaps the most important stage. The pilot who first detects his enemy has the upper hand in a combat situation. He may maneuver to keep from being detected himself, he may be able to engage the enemy unawares, and perhaps most importantly, if he determines that his chances of success are slim, he may choose to not attack at all and so withdraw to find a more inviting situation. (Spick 1988). Detection for most helicopter engagements is visual, though the pilot's attention may be directed towards his adversary by a ground controller. (Wilson 1989). Advancements in helicopter borne radar systems such as may be found on the AH-64D "Longbow Apache" show great promise in aiding the pilot in picking out his targets before they can pick him.

Once a contact is detected, it must be identified as friend or foe. Long-range sensors can aid the pilot in determining the type and number of the enemy aircraft. (Wilson 1989). It is at this point that the pilot will decide to attack or retreat.

Closing

The closing phase involves diminishing the range to target until an attack can be made. With the advent of long-range precision guided missiles such as those discussed in the body of this report, this stage may be bypassed in favor of an immediate attack. In this case the winner will usually be the pilot who shoots first, which is why, once again, the first stage of aerial combat is so important. If the pilot decides to close with the enemy aircraft, it will be done in such a way as to maximize his chance of success. This primarily consists of maneuvering to approach undetected if possible and to position himself for the most advantageous shot possible. (Spick 1988). Weapons systems characteristics will dictate to a great extent the manner in which the pilot closes for attack. If the enemy detects the attacking pilot, then he must quickly decide whether to continue to close or to disengage (phase 5).

Attack

Once the attacking pilot has closed within weapons release range, it is time to press home the attack. The target is acquired in the weapons sighting system, the aircraft is flown into weapons release parameters and the attack is made. Historical records show that 80% of all successful attacks occur during the initial attack against an enemy who never detected his attacker. If the attack is unsuccessful however, the enemy pilot will almost assuredly become aware of the situation, so the pilot may once again decide to disengage (if possible). On the other hand, he may reattack using the same weapon, or he may close to use a weapon more suited for the particular engagement. (Spick 1988).

Maneuver

Maneuver is the phase that most readily comes to mind when considering aerial combat, and it has become firmly entrenched in the popular media as the “dogfight”. The maneuver stage occurs when both sides of the conflict are aware of the other’s intent, and are both maneuvering to obtain a good firing solution on the other. Pilots trained in air-to-air combat learn specific maneuvers to gain the advantage over an enemy pilot, and to counter any maneuvers that the enemy pilot may attempt to gain the advantage over him. Aircraft performance and maneuverability are important during this stage, as is aircraft weapons capability. The aircraft with the larger firing envelope for its weapons has a decided advantage over one that requires a narrow set of parameters be met for successful weapons release. A pilot’s superior ability to maintain a grasp on the rapidly changing situation around him (situational awareness or SA), may enable him to counter his enemy’s superior airmanship or aircraft. (Spick 1988).

Disengagement

At some point, the pilot will elect to disengage. It may be uneventful, as when all enemy aircraft have been downed and the victorious pilot returns to home base. Or it may be that the pilot determines that his chances of success are rapidly dwindling to the non-existent, and he picks an advantageous moment to make a break for it and escape an unfortunate end. In this age of guided missiles and long-range weapons, it takes a bit of judicious timing to successfully disengage from a superior enemy force. (Spick 1988).

AERIAL GUNNERY GEOMETRY AND TERMS

In air-to-air combat, the relative motion between aircraft and their individual flight paths and velocities gives rise to a variety of geometric relationships. The terms used in describing these relationships have developed over a period of time, but are in general usage among aerial combat trained pilots, at least in the US, Canada, and Great Britain.

Not all of these terms need be committed to memory, but a basic familiarity with the shorthand that the combat pilot speaks will greatly aid in discussing air-to-air gunnery. The portion of the target presented to the attacking aircraft (or "shooter") is the target's aspect, and the angle between the longitudinal axis of the target (projected rearward) and the LOS to the shooter is the "aspect angle", or target aspect angle (TAA). A target facing towards the shooter has a TAA of 0° , and facing away a TAA of 180° . (Shaw 1985). "Angle off" (AO) is shorthand for "angle off the tail" and refers to the angle between the shooter's LOS and an imaginary line extending from the tail of the target aircraft. AO is the reciprocal of TAA. (DON 1993a). Figure 16 from Shaw (1985) depicts some of these concepts.

The positional relationship between the shooter and the target aircraft is critical for both maneuvering and for weapons release considerations. The relative alignments of the shooter and target aircraft's flight paths will dictate how far ahead of the target aircraft the shooter must aim to hit the target at the end of the projectile TOF. The difference between the LOS from the shooter to the target, and the LOS of the weapon sight is the "lead angle",

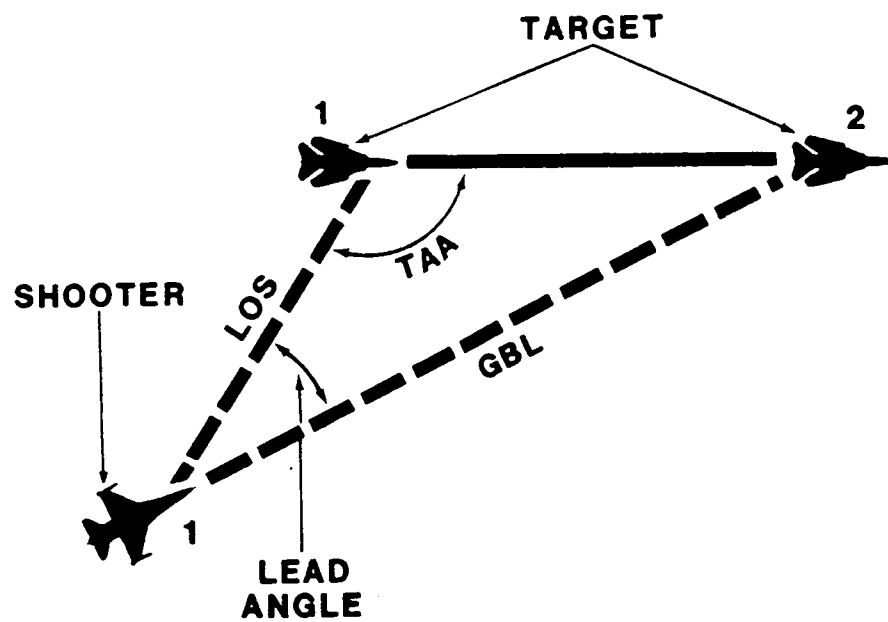


Figure 16: Gun Firing Geometry.

Source: Shaw (1985).

Notes:

- (a) Target position (1) is prior to firing. Target position (2) is after the time of flight of the projectile.
- (b) GBL = Gun Bore Line; LOS = Line-of-Sight; TAA = Target Aspect Angle

or simply “lead”. There is no lead required for targets with a TAA of 0° or 180° , and the greatest lead is required for a target with a TAA of 90° . Shots that require a lead angle are called “deflection” shots and as the TAA approaches 90° , “high-deflection” shots. (Shaw 1993).

There are two types of gunnery firing methods employed in air-to-air gunnery situations. The “tracking shot” takes place when the aiming reticle remains steady on the computed aim point, a “snapshot” or “raking guns” is when the reticle merely passes through the proper aim point without stopping. Tracking shots are characterized by near zero relative motion between the target and the aiming reticle. Raking guns are often used when the shooter is required to estimate the aim point without the assistance of a lead computing optical sight (LCOS). The pilot will estimate the appropriate aim point and the fly the reticle back and forth over this point while firing to compensate for any error in his mental computation.(Shaw 1985).

When attempting a tracking shot, the shooter must first maneuver his aircraft to decrease range-to-target to within the maximum effective range of the gun system, usually by moving inside the target aircraft’s turn radius. Then the shooter “pulls” an excess lead angle (positions his sighting system LOS ahead of the computed aim point), and as the target comes within range relaxes his aircraft angle of bank to reduce the shooter’s turn rate. This will allow the target to fly up to the appropriate lead angle, and to position the shooter’s aircraft in the optimum firing position. The optimum firing position for an aircraft with an LCOS is within 60° AO, and for an aircraft with a fixed sight within 30° AO. If the aircraft is equipped with an LCOS, the pilot may need to hold this orientation for the “settling time”

of the sight (while the fire control computer completes its computations) prior to opening fire. (Shaw 1985).

Snapshots afford less firing time on target, but may provide a less predictable flight path for the shooter and possibly greater survivability during the engagement. Snapshots are usually described in terms of the shooter's normal acceleration or "G". A low-G snapshot occurs when the shooter assumes a near wings level attitude with an excess lead angle and waits for the target to fly into the firing position. The shooter may apply raking fire over the estimated aim point after the target reaches the desired orientation. A high-G snapshot is very similar to a tracking shot, but the shooter will be unable to stop the relative motion of the target and the sighting reticle. The shooter will slow the relative motion as much as possible to allow the maximum amount of time with the target in the projectile stream. Snapshots normally occur somewhere between the low-G and high-G extremes. (Shaw 1985). Figure 17 from Shaw (1985) depicts a nominal effective gun envelope for tracking shots and snapshots.

The concepts of "lead pursuit" and "lag pursuit" are key to understanding maneuver during air-to-air combat. The underlying principle is that a pursuing aircraft can control his closure rate to the target by moving inside or outside of the target's turn radius. Assuming that the shooter and target are moving at nearly the same speed, when inside the target's turn the shooter will have less distance to cover and will close with the target. Similarly, outside the target's turn, the shooter will have farther to travel than the target and will open the distance to the target. A lead pursuit is good for obtaining an excess lead angle prior to attempting a tracking shot or high-G snapshot, and can allow a much slower aircraft to

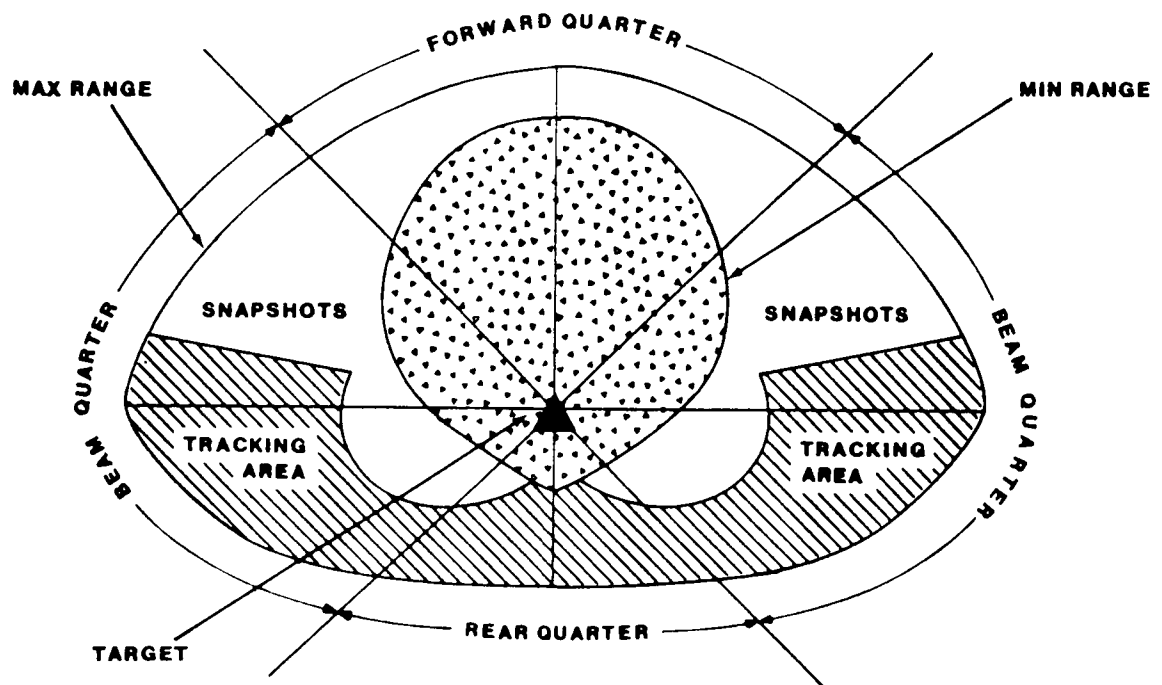


Figure 17: Effective Gun Firing Envelope.

Source: Shaw (1985).

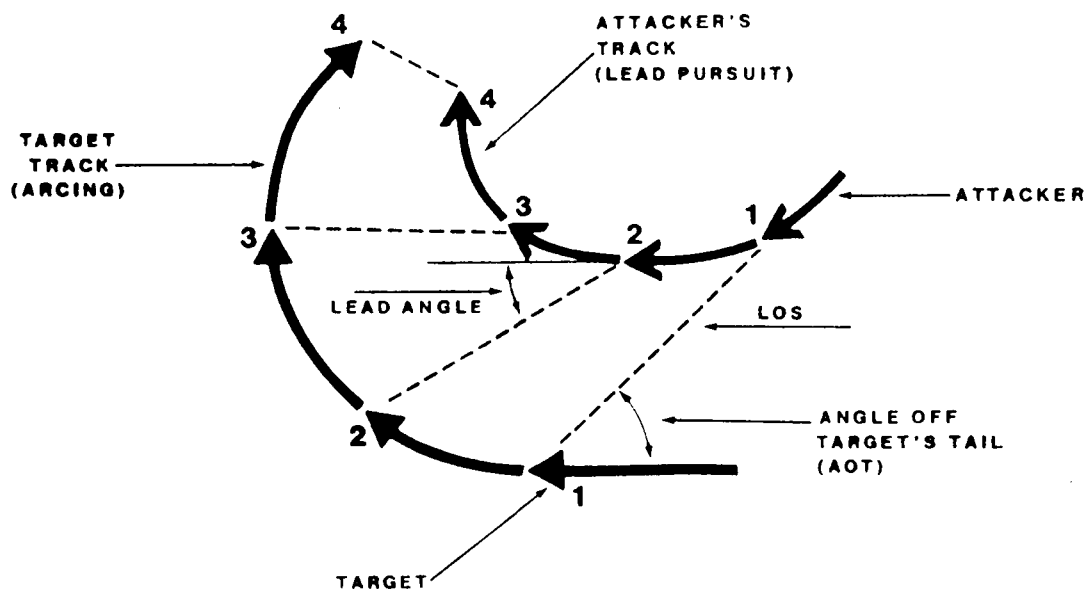
maintain a rear quarter position on a faster target. A lag pursuit is good for slowing excess closure rate if the shooter is in danger of overshooting the target, and can allow a much faster aircraft to maintain a rear quarter position on a slower target. (Shaw 1985). Figure 18 from Shaw (1985) depicts the lead and lag pursuit curves.

RANGE ESTIMATION

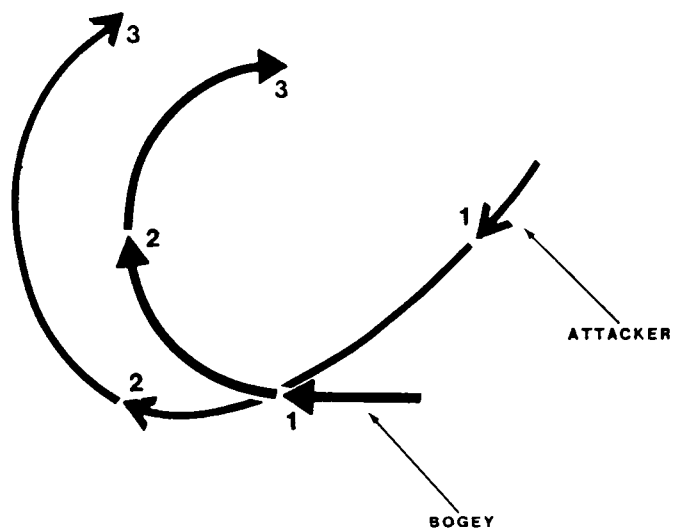
As previously noted, range to target is a key element required for accurate ballistics computations for air-to-air gunnery. Determining an accurate range to target without a range measuring device such as a laser rangefinder or an onboard radar requires that the pilot estimate the range. Range estimation for air-to-air combat is accomplished in three ways; estimation by eye, comparison to tracer burnout, and reference to sight mil values. (DON 1993a).

Range estimation made without any reference device is normally referred to as estimation by eye. Estimation by eye is notoriously inaccurate, and its accuracy is highly dependent on the target aspect and environment, meteorological conditions, and pilot training and visual acuity. Estimation by eye is the least preferred method of range estimation, but it has the advantage of speed over reference to sight mil values (see below), and the advantage of covertness over reference to tracer burnout. (DON 1993a).

Tracer projectiles carry a readily ignitable substance in their base which puts off a bright glow when the projectile is fired. The phrase "tracer burnout" refers to the point in the tracer projectile's trajectory where the tracer material is consumed and extinguishes. The



Lead Pursuit



Lag Pursuit

Figure 18: Lead and Lag Pursuit Curves.

Source: Shaw (1985).

amount of tracer material installed is selected to provide a specific time of burn which, when combined with the known speed of the projectile, provides a nominal distance to tracer burnout. For example, the M50 family of 20mm ammunition includes tracer rounds which have a nominal distance to burnout of 1200 to 1500 meters. Reference to tracer burnout involves firing tracer ammunition at the target, and then comparing the relative distance of the target to where the tracer material extinguishes. If tracer burnout occurs halfway to the target when using M50 tracer ammunition, then the pilot can estimate that the target is 2400 to 3000 meters away. (DON 1993a).

Using tracer burnout as a reference for range estimation has several drawbacks. Tracer ammunition is visible from quite a distance away and firing tracers invariably gives away the location of the firing platform. If the target aircraft was unaware of the attacking aircraft, firing tracer ammunition at him will quickly apprise him of the situation and will normally elicit aggressive defensive maneuvering. In fact, some commands have restricted the use of tracer ammunition for air-to-air firing for just this reason. From Shaw (1985).

“[The Commanding Officer] ordered the tracer ammo removed ... I’ll never forget the spectacular results we got. Our kill rate went up from 50 to 100 per cent.”

Colonel Charles W. King, USAF
5 Victories, WW-II

Another drawback to using tracer burnout for range estimation is that it still requires estimation by eye to judge the relative distance between tracer burnout and the target, with all of the inaccuracies inherent in that method. Finally, different families of tracer ammunition give off different colored signatures which can identify the firing aircraft to

enemy ground forces and enemy aircraft not involved in the fight (most Russian made tracer ammunition burns with a green color while most US made tracers burn red).

The final method for range estimation is the use of reference to sight mil values. As was previously noted, the term “mil” is pilot shorthand for milliradian ($1^\circ = 17.5$ mils). Reticle schemes for gun and rocket sights are developed so that the reticle structure provides a set of known mil values for reference in range estimation. These values are consistent if the pilot’s eye is positioned with proper relief to the sight, a point in space known as the design eye point (DEP). Figure 19 depicts the gun sight reticle from the AH-1W “SuperCobra” HUD.

The significance of using mil values for range estimation lies in the basic angular relationship of the milliradian to range. For any unit of range (foot, meter, kilometer, etc.), one mil of angular displacement is equal to 1/1000th of the range in lateral span. So, at 1000 feet, an object that occupies one mil of angular displacement represents a span that is 1 foot wide. This relationship has given rise to the following formulas which are used to compute target range. It is important to note that the units of measure for range and target width must be consistent.

$$\text{Target Width} = \frac{(\text{Range}) (\text{Mil Value})}{1000} \quad \text{Range} = \frac{(\text{Target Width}) (1000)}{\text{Mil Value}}$$

Armed with these formulas and a reference of possible target widths, the pilot can compute range to target with a reasonable degree of accuracy. Inaccuracies creep in when the width of the target is not known exactly (for example if the target does not provide a perfect

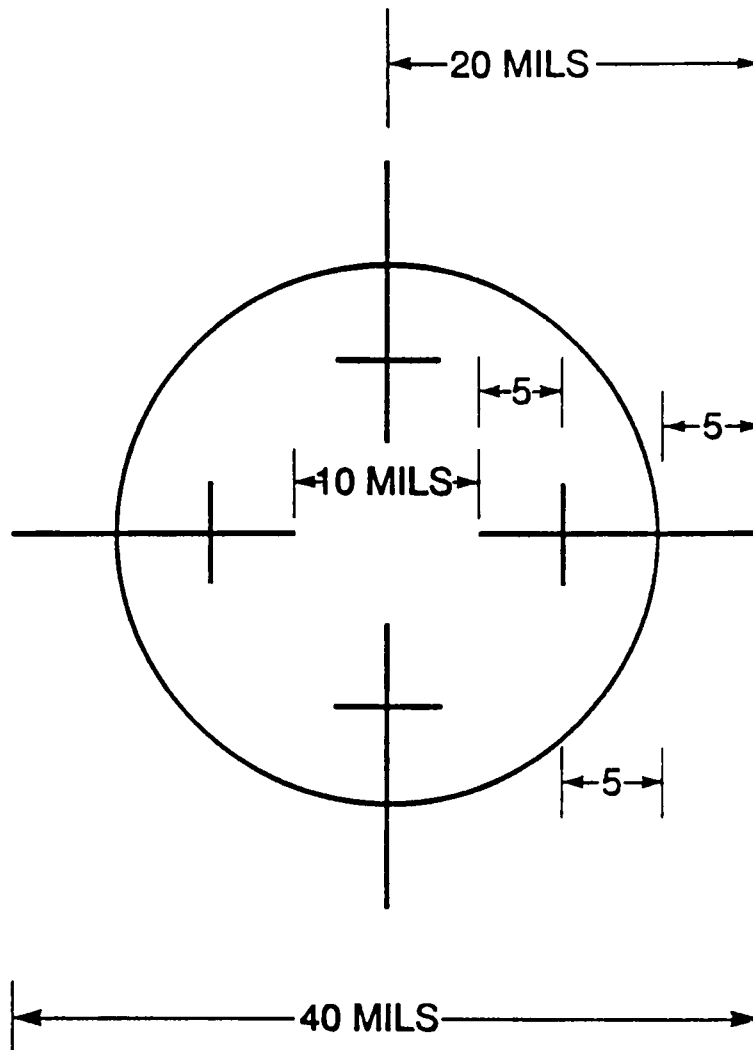


Figure 19: AH-1W Gun Reticle Diagram Showing Mil Sizing.

Source: DON (1993a).

90° aspect) or if the pilot shifts his eye position nearer or farther away from the sight, which skews the mil sizing relationship.

Of course, it is not possible for the average pilot to mentally perform these calculations during the execution of an air-to-air engagement. Therefore, the relationships must be computed prior to launching on a combat sortie, based on the known widths of target vehicles and aircraft the pilot can reasonably expect to meet. The AH-1 Tactical Manual includes several depictions for pilot study which illustrate this concept. Figure 20 presents one such depiction of a Russian MiG-29 at different ranges as viewed through the AH-1W rocket reticle.

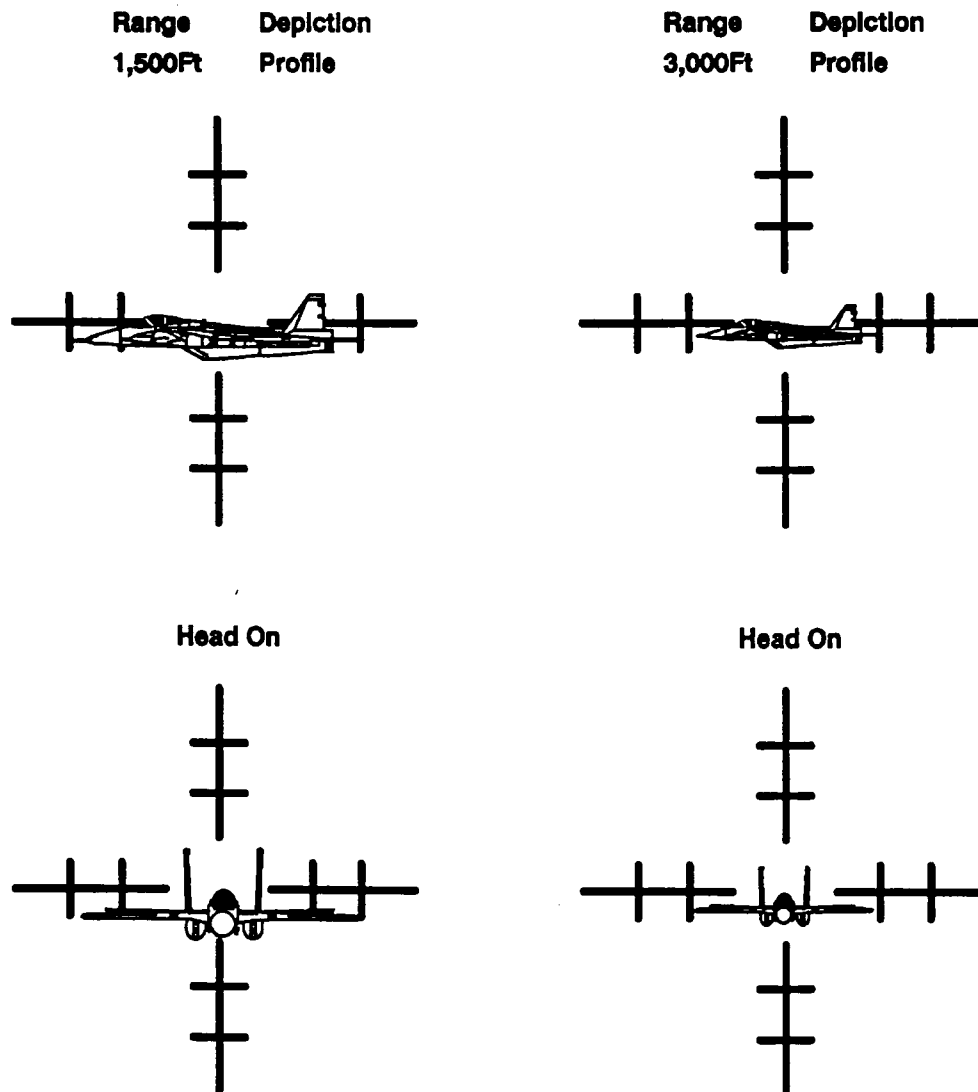


Figure 20: MiG-29 AH-1W HUD Rocket Reticle Depiction.

Source: DON (1993a).

APPENDIX C

COMPUTER HARDWARE

Computer

A Gateway 2000® P75 Intel Pentium® based desktop computer served as the software development and system operation central processing unit. A Gateway 2000® 15 inch monitor served as the display for the gun tracking accuracy program.

Controller

A Thrustmaster® Mark II FCS™ computer gaming joystick served as the controller for the gun tracking simulation.

COMPUTER SOFTWARE

Microsoft Visual Basic® 3.0

Microsoft Visual Basic® 3.0 was used to write both the gun tracking and ballistics simulations as detailed in the body of the report.

Crescent QuickPak Scientific®

A Visual Basic add-in module from Crescent QuickPak Scientific® was used to

compute probabilities and x values during the generation of normally distributed pseudo-random variables.

Microsoft Excel®

Microsoft Excel® was used to reduce and manipulate the data during the data analysis and model construction phases of the simulation effort. Of particular use was the curve-fitting modules and the statistical and numerical analysis tools included with Excel®

Jantel Scientific SigmaPlot® for Windows®

Jantel Scientific SigmaPlot® for Windows® was used to create the plots and diagrams presented in this thesis. Its excellent mathematical transform engine was also used during data analysis and manipulation.

APPENDIX D

GUN TRACKING SIMULATION PROGRAM CODE

The Visual Basic® program code for the gun tracking simulation is presented below. Program code in Visual Basic® is grouped by the Windows® form which contains the code. Each form which is displayed during the running of the program and each object on that form, such as a text box, check box, or menu bar item, contains the code required to execute its function. Additional code is contained in modules which hold function and subroutine definitions as well as variable and constant definitions. During program development, each group of code is stored as an individual file. Form files are saved with a “.FRM” appendix and modules with a “.BAS” appendix. When the program is completed, Visual Basic® constructs a single executable file using the saved form and module files.

Some lines of code are very long and wrap across more than one line in the listing. A line of code which wraps across more than one line in this listing is indicated by the symbol “➡” at the end of the line. This symbol has no meaning in Visual Basic®, and merely serves to indicate that the next line should be considered a continuation of the line in which the symbol appears. Comments are italicized. Comments reside in the lines of code to aid in writing and debugging the code, but are ignored by the program during run time.

For the purpose of preparing this appendix, the Visual Basic® project files were saved as text files and then formatted using a word processor. Visual Basic® does not support page formatting in its print function.

FRMABOUT.FRM - 1

Option Explicit

Sub cmdOkay_Click ()

 Unload frmAbout

End Sub

Sub Form_Load ()

positions screen in center of screen

 frmAbout.Top = (Screenheight% - 4000) / 2

 frmAbout.Left = (Screenwidth% - 7550) / 2

End Sub

FRMCALIB.FORM -1

Option Explicit

Sub cmdCancel_Click ()

Dim jcap As JOYCAPS

Dim jpos As JOYPOS

Dim i!, rc%

Unload frmCalibrate

Get position information

rc% = joyGetPos%(JOYSTICKID1, jpos)

Inform the user of any problem

If (rc% <> JOYERR_NOERROR) Then

MsgBox "Joystick Driver not installed or Joystick unplugged!", MB_OK ➡

+ MB_EXCLAMATION, "Joystick Error!"

frmConfigure.optJoystick.Value = False

frmConfigure.optJoystick.Visible = False

frmConfigure.optKeyboard.Value = True

Joystick\$ = "Off"

Keyboard\$ = "On"

End If

frmConfigure.Show Modal

If Joystick\$ = "On" Then

frmCalibrate.Show Modal

End If

frmStart.Show Modal

End Sub

Sub cmdOkay_Click ()

Unload frmCalibrate

End Sub

FRMCALIB.FRM - 2

Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)

 If KeyCode = KEY_RETURN Then Unload Me

End Sub

Sub Form_Load ()

positions screen in center of display

 frmCalibrate.Top = (Screenheight% - 4500) / 2

 frmCalibrate.Left = (Screenwidth% - 7500) / 2

End Sub

Sub tmrCalibrate_Timer ()

 Dim jcap As JOYCAPS

 Dim jpos As JOYPOS

 Dim i!, rc%

 If Joystick\$ = "On" Then

Get joystick position information

 rc% = joyGetPos%(JOYSTICKID1, jpos)

Get joystick position & save center positions

 lblInstructions.Caption = "Center Joystick and Click Trigger"

 If ((jpos.Buttons And JOY_BUTTON1) = JOY_BUTTON1) Then

 jX& = (CLng(jpos.X) And &HFFFF&)

 jY& = (CLng(jpos.Y) And &HFFFF&)

 joyCtrX& = jX&

 joyCtrY& = jY&

 lblCenter.Caption = "OKAY"

 lblCenter.BackColor = &HFF&

 cmdOkay.Enabled = True

 cmdCancel.Enabled = False

 End If

 End If

End Sub

FRMCONF2.FRM - 1

Option Explicit

Joystick variables

Dim jpos As JOYPOS

Sub cmdOkay_Click ()

 If optPractice1.Value = True Then

 LeadHelp\$ = "Show"

 Datawrite\$ = "No"

 End If

 If optPractice2.Value = True Then

 LeadHelp\$ = "Noshow"

 Datawrite\$ = "No"

 End If

 If optData.Value = True Then

 LeadHelp\$ = "Noshow"

 Datawrite\$ = "Yes"

 End If

 Unload frmConfigure2

End Sub

Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)

 If KeyCode = KEY_RETURN Then Unload frmConfigure2

End Sub

Sub Form_Load ()

 frmConfigure2.Top = (Screenheight% - 4500) / 2

 frmConfigure2.Left = (Screenwidth% - 7500) / 2

 If Keyboard\$ = "On" Then optData.Visible = False

End Sub

FRMCONFL.FRM - 1

Option Explicit

Joystick variables

Dim jpos As JOYPOS

Sub cmdOkay_Click ()

 If optJoystick.Value = True Then

 Joystick\$ = "On"

 Keyboard\$ = "Off"

 End If

 If optKeyboard.Value = True Then

 Joystick\$ = "Off"

 Keyboard\$ = "On"

 End If

 If optPractice1.Value = True Then

 LeadHelp\$ = "Show"

 Datawrite\$ = "No"

 End If

 If optPractice2.Value = True Then

 LeadHelp\$ = "Noshow"

 Datawrite\$ = "No"

 End If

 If optData.Value = True Then

 LeadHelp\$ = "Noshow"

 Datawrite\$ = "Yes"

 End If

 If optKeyboard.Value = True And optData.Value = True Then

 Datawrite\$ = "No"

 optData.Visible = False

 MsgBox "Cannot Record Data Utilizing KeyBoard Input. Data Recording ➡

Disabled.", MB_OK + MB_EXCLAMATION, "Input Error!"

 End If

 Unload frmConfigure

FRMCONFI.FRM - 2

End Sub

Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)

 If KeyCode = KEY_RETURN Then Unload Me

End Sub

Sub Form_Load ()

 frmConfigure.Top = (Screenheight% - 4500) / 2

 frmConfigure.Left = (Screenwidth% - 7500) / 2

End Sub

FRMCONT.FRM - 1

Option Explicit

Dim rc%

Joystick variables

Dim jpos As JOYPOS

Sub cmdContinue_Click ()

 Unload frmContinue
 frmStart.Show Modal

End Sub

Sub cmdContinue_KeyDown (KeyCode As Integer, Shift As Integer)

 If KeyCode = KEY_RETURN Then
 frmStart.Show Modal
 Unload frmContinue
 End If

End Sub

Sub Form_Load ()

 frmContinue.Top = (Screenheight% - 1005) / 4
 frmContinue.Left = (Screenwidth% - 1215) / 4

End Sub

FRMOPT.FRM - 1

Option Explicit

Joystick variables

Dim jpos As JOYPOS

Sub cmdCancel_Click ()

 Unload frmOptions
 frmStart.Show Modal

End Sub

Sub cmdOkay_Click ()

 If optFixed.Value = True Then Reticlecounter% = 1
 If optFixedLead.Value = True Then Reticlecounter% = 2
 If optCCIP.Value = True Then Reticlecounter% = 3
 If chkLeadHelp.Value = 1 Then LeadHelp\$ = "Show"
 If chkLeadHelp.Value = 0 Then LeadHelp\$ = "Noshow"

Reconfigure for practice runs

 Datawrite\$ = "No"

 Unload frmOptions
 frmStart.Show Modal

End Sub

Sub cmdOkay_KeyDown (KeyCode As Integer, Shift As Integer)

 If KeyCode = KEY_RETURN Then
 Datawrite\$ = "No"
 Unload frmOptions
 frmStart.Show Modal
 End If

End Sub

Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)

 If KeyCode = KEY_RETURN Then Unload frmOptions

FRMOPT.FRM - 2

End Sub

Sub Form_Load ()

centers form in display

frmOptions.Top = (Screenheight% - 4500) / 2

frmOptions.Left = (Screenwidth% - 7500) / 2

End Sub

FRMREGIS.FRM - 1

Option Explicit

Joystick variables

Dim jpos As JOYPOS

Sub cmdOkay_Click ()

 Lastname\$ = txtName.Text
 FlightHours\$ = txtFlightHours.Text
 Qual\$ = cboQual.Text
 Airtype\$ = cboAircraftType.Text
 Unload frmRegister

End Sub

Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)

 If KeyCode = KEY_RETURN Then Unload Me

End Sub

Sub Form_Load ()

 frmRegister.Top = (Screenheight% - 4500) / 2
 frmRegister.Left = (Screenwidth% - 7500) / 2
 cboQual.AddItem "Pilot"
 cboQual.AddItem "Non-Pilot"
 cboQual.ListIndex = 0
 cboAircraftType.AddItem "Fighter/Attack"
 cboAircraftType.AddItem "Transport/Utility/Recce"
 cboAircraftType.AddItem "Other"
 cboAircraftType.AddItem "None"
 cboAircraftType.ListIndex = 0

End Sub

FRMSIM.FRM - 1

Option Explicit

Dim rc% *General variable*
Dim temp! *General variable*
Dim temp1! "
Dim temp2! "

Joystick variables

Dim jpos As JOYPOS
Dim jcap As JOYCAPS

Sub Form_KeyDown (KeyCode As Integer, Shift As Integer)

 If Keyboard\$ = "On" Then

 Select Case (KeyCode)

 Case KEY_LEFT
 Roll! = Roll! + 1 / 1.5

 Case KEY_UP
 Pitch! = Pitch! - 5.333 / 3

 Case KEY_RIGHT
 Roll! = Roll! - 1 / 1.5

 Case KEY_DOWN
 Pitch! = Pitch! + 5.333 / 3

 End Select

 End If

End Sub

Sub Form_Load ()

 Today = Now
 frmWorkArea.Show

Get position information

 rc% = joyGetPos%(JOYSTICKID1, jpos)

Inform the user of any problem

```
If (rc% <> JOYERR_NOERROR) Then
    MsgBox "Joystick Driver not installed or Joystick unplugged!", MB_OK ➡
    + MB_EXCLAMATION, "Joystick Error!"
    frmConfigure.optJoystick.Value = False
    frmConfigure.optJoystick.Visible = False
    frmConfigure.optKeyboard.Value = True
    Joystick$ = "Off"
    Keyboard$ = "On"
```

End If

```
frmAbout.Show Modal
frmRegister.Show Modal
frmConfigure.Show Modal
```

```
If Joystick$ = "On" Then
    frmCalibrate.Show Modal
End If
```

Get the next available file number

```
FileNum = FreeFile
```

Create the file "lastname.CVS"

```
If Datawrite$ = "Yes" Then
    Open "D:\\" + Lastname$ + ".CSV" For Output As FileNum
    Write #FileNum, Lastname$, FlightHours$, Qual$, Airtype$, ➡
    Format(Today, "mm/dd/yy h:mm")
End If
```

```
frmStart.Show Modal
```

End Sub

Sub Form_Resize ()

```
Randomize
```

Calculate new form height & width & center

```
ScreenHeight% = frmWorkArea.Height
ScreenWidth% = frmWorkArea.Width
```

```
frmHeight! = ScaleHeight  
frmWidth! = ScaleWidth  
frmCenterX! = frmWidth! / 2  
frmCenterY! = frmHeight! / 2
```

Position Pitch, Roll and Score labels

```
lblPitch.Left = frmWidth! - 83  
lblPitch2.Left = frmWidth! - 83  
lblRoll.Left = frmCenterX! - (80 / 2)  
lblRoll2.Left = frmCenterX! - (80 / 2)  
lblScore.Left = frmCenterX! - (80 / 2)  
lblPitch.Top = frmCenterY! - 25 / 2  
lblPitch2.Top = frmCenterY! + 27 / 2  
lblRndctr.Top = frmCenterY! - 25 / 2  
lblSpeed.Top = frmCenterY! + 27 / 2  
lblScore.Top = frmHeight! - 25 - 8
```

Calculate roll circle unit size

```
Rollunit! = Sqr(((frmHeight!) ^ 2) + ((frmWidth!) ^ 2)) * 3
```

Initialize horizon line

```
CtrlX! = frmCenterX!  
CtrlY! = frmCenterY!  
LineEndX1! = CtrlX! - (Rollunit! / 2)  
LineEndX2! = CtrlX! + (Rollunit! / 2)  
LineEndY1! = frmCenterY!  
LineEndY2! = frmCenterY!
```

Determine spacing between endpoints of perspective lines

```
Linespacefar! = frmWidth! / 10  
Linespacenear! = (frmWidth! * 25) / 10
```

Initialize variables

```
Reticlecounter% = 1
```

End Sub

Sub Form_Unload (Cancel As Integer)

Close the open data file

```
Close FileNum
```

FRMSIM.FRM - 4

End Sub

Sub lblGo_Click ()

tmrRun.Enabled = True

End Sub

Sub lblPause_Click ()

tmrRun.Enabled = False

End Sub

Sub mnuAbout_Click ()

frmAbout.Show Modal

End Sub

Sub mnuExit_Click ()

Call EndProgram

End Sub

Sub tmrRun_Timer ()

Increment line counter

Linecounter! = Linecounter! + 1

Determine time since last timer click

Laptimer! = Timer - MasterClock!

MasterClock! = Timer

TimeCounter! = TimeCounter! + Laptimer!

If Laptimer! = 0 GoTo skipstep:

If TgtRange! = 0 GoTo skipstep:

Clear screen

frmWorkArea.Cls

If Joystick\$ = "On" Then

Get joystick position

rc% = joyGetPos%(JOYSTICKID1, jpos)
jX& = (CLng(jpos.X) And &HFFFF&)
jY& = (CLng(jpos.Y) And &HFFFF&)

Get the joystick capabilities

rc% = joyGetDevCaps%(JOYSTICKID1, jcap, JOYCAPSIZE)

Get maximum range into a long, for translations

joyMaxX& = (CLng(jcap.Xmax) And &HFFFF&)
joyMaxY& = (CLng(jcap.Ymax) And &HFFFF&)

Determine commanded rollrate based on joystick position

If Abs(joyCtrX& - jX&) >= 10 Then RollRate! = MAXROLLRATE! * ➡
(joyCtrX& - jX&) / joyCtrX&

Determine commanded pitchrate based on joystick position

If Abs(joyCtrY& - jY&) >= 10 Then PitchRate! = MAXPITCHRATE! * ➡
(joyCtrY& - jY&) / joyCtrY&

End If

Reduce roll rate over time to provide roll damping

If RollRate! > 0 Then
RollRate! = RollRate! - (Laptimer! * ROLLDAMP!)
If RollRate! < 0 Then RollRate! = 0
ElseIf RollRate! < 0 Then
RollRate! = RollRate! + (Laptimer! * ROLLDAMP!)
If RollRate! > 0 Then RollRate! = 0
Else
RollRate! = 0
End If

If using keyboard compute RollRate!

If Keyboard\$ = "On" Then
RollRate! = (Roll! - OldRoll!) / Laptimer
End If

Limit roll rate

If RollRate! >= MAXROLLRATE! Then RollRate! = MAXROLLRATE!
If RollRate! <= -MAXROLLRATE! Then RollRate! = -MAXROLLRATE!

Generate roll with rollrate

Roll! = Roll! + RollRate! * Laptimer!

Reduce roll over time to provide spiral stability

If Roll! > 0 Then
 Roll! = Roll! - (Laptimer! * ROLLSTAB!)
ElseIf Roll! < 0 Then
 Roll! = Roll! + (Laptimer! * ROLLSTAB!)
Else
 Roll! = 0
End If

Generate roll label

If Roll! >= ROLLLIMIT! Then
 lblRoll.ForeColor = &HC0&
 lblRoll.FontSize = 13.5
 Roll! = ROLLLIMIT!
 lblRoll.Caption = "LIMIT"
ElseIf Roll! <= -ROLLLIMIT! Then
 lblRoll.ForeColor = &HC0&
 lblRoll.FontSize = 13.5
 Roll! = -ROLLLIMIT!
 lblRoll.Caption = "LIMIT"
Else
 lblRoll.ForeColor = &H80000008
 lblRoll.FontSize = 13.5
 lblRoll.Caption = Int(-Roll!)
End If

Reduce pitch rate over time to provide pitch damping

If PitchRate! > 0 Then
 PitchRate! = PitchRate! - (Laptimer! * PITCHDAMP!)
 If PitchRate! < 0 Then PitchRate! = 0
ElseIf PitchRate! < 0 Then
 PitchRate! = PitchRate! + (Laptimer! * PITCHDAMP!)
 If PitchRate! > 0 Then PitchRate! = 0
Else

```
PitchRate! = 0
End If
```

If using keyboard compute PitchRate!

```
If Keyboard$ = "On" Then
    PitchRate! = (Pitch! - OldPitch!) / Laptimer
End If
```

Limit pitch rate

```
If PitchRate! >= MAXPITCHRATE! Then PitchRate! = MAXPITCHRATE!
If PitchRate! <= -MAXPITCHRATE! Then PitchRate! = -MAXPITCHRATE!
```

Generate pitch with pitchrate

```
Pitch! = Pitch! + PitchRate! * Laptimer!
```

Reduce pitch over time to provide speed stability

```
If Pitch! > 0 Then
    Pitch! = Pitch! - (Laptimer! * PITCHSTAB! * 5.333)
ElseIf Pitch! < 0 Then
    Pitch! = Pitch! + (Laptimer! * PITCHSTAB! * 5.333)
Else
    Pitch! = 0
End If
```

Generate pitch label

```
If Pitch! / 5.333 >= PITCHLIMIT! Then
    Pitch! = PITCHLIMIT! * 5.333
    lblPitch.ForeColor = &HC0&
    lblPitch.FontSize = 13.5
    lblPitch.Caption = "LIMIT"
ElseIf Pitch! / 5.333 <= -PITCHLIMIT! Then
    Pitch! = -PITCHLIMIT! * 5.333
    lblPitch.ForeColor = &HC0&
    lblPitch.FontSize = 13.5
    lblPitch.Caption = "LIMIT"
Else
    lblPitch.ForeColor = &H80000008
    lblPitch.FontSize = 13.5
    lblPitch.Caption = Int(-Pitch! / 5.333)
End If
```

Compute sin & cos values of roll to reduce trig function calls

```
SinRoll! = Sin(Roll! * DEG_TO_RAD!)
CosRoll! = Cos(Roll! * DEG_TO_RAD!)
```

Adjust speed based on pitch and roll angle

```
Speed! = Speed! - (Laptimer! * -(Pitch!) / 5.3333) - (Laptimer! * (Abs(Roll! / 8)))
If Speed! >= 150 Then Speed! = 150
If Speed! <= 90 Then Speed! = 90
```

Compute turn rate as a factor of roll angle and speed

```
TurnRate! = (Sgn(Roll!)) * (((-30 / 8100) * ((Abs(Roll!) - 90) ^ 2) + 30) * 60 / ➡
Speed!) / 7
```

Compute TurnCtl! flag (-5 to 5) to adjust perspective line movement

```
TurnCtl! = TurnCtl! - TurnRate! * Laptimer!
If TurnCtl! > 20 Then TurnCtl! = TurnCtl! - 20
If TurnCtl! < -20 Then TurnCtl! = TurnCtl! + 20
If TurnCtl! > 15 Then TurnCtl! = TurnCtl! - 15
If TurnCtl! < -15 Then TurnCtl! = TurnCtl! + 15
If TurnCtl! > 10 Then TurnCtl! = TurnCtl! - 10
If TurnCtl! < -10 Then TurnCtl! = TurnCtl! + 10
If TurnCtl! > 5 Then TurnCtl! = TurnCtl! - 5
If TurnCtl! < -5 Then TurnCtl! = TurnCtl! + 5
```

Determine distance from old center line to old center target

```
Tgtunit! = Sqr((TgtCenterX! - CtrlineX!) ^ 2 + (TgtCenterY! - CtrlineY!) ^ 2)
```

Determine TgtAngle! (polar angle of target from center of horizon line)

```
If TgtCenterX! > CtrlineX! Then
    If TgtCenterY! > CtrlineY! Then
        TgtAngle! = Abs(Atn((TgtCenterY! - CtrlineY!) / (TgtCenterX! - ➡
CtrlineX!))) * RAD_TO_DEG!) - OldRoll!
    Else
        TgtAngle! = -Abs(Atn((TgtCenterY! - CtrlineY!) / (TgtCenterX! - ➡
CtrlineX!))) * RAD_TO_DEG!) - OldRoll!
    End If
ElseIf TgtCenterX! = CtrlineX! Then
    If TgtCenterY! < CtrlineY! Then
        TgtAngle! = -90 - OldRoll!
    ElseIf TgtCenterY! = CtrlineY! Then
        Tgtunit! = 0
```

```

Else
    TgtAngle! = 90 - OldRoll!
End If
Else
    If TgtCenterY! > CtrlineY! Then
        TgtAngle! = (180) - Abs(Atn((TgtCenterY! - CtrlineY!) / ➡
            (TgtCenterX! - CtrlineX!)) * RAD_TO_DEG!) - OldRoll!
    Else
        TgtAngle! = (180) + Abs(Atn((TgtCenterY! - CtrlineY!) / ➡
            (TgtCenterX! - CtrlineX!)) * RAD_TO_DEG!) - OldRoll!
    End If
End If

```

Draw horizon line to background

Determine the location of the center line

CtrlineX! = Pitch! * SinRoll! + frmCenterX!

CtrlineY! = (-1) * Pitch! * CosRoll! + frmCenterY!

Determine the X & Y components of horizon line relative to roll angle

LineValX! = Rollunit! * CosRoll!

LineValY! = Rollunit! * SinRoll!

LineEndX1! = CtrlineX! - (LineValX! / 2)

LineEndX2! = CtrlineX! + (LineValX! / 2)

LineEndY1! = CtrlineY! - (LineValY! / 2)

LineEndY2! = CtrlineY! + (LineValY! / 2)

Set draw width

frmWorkArea.DrawWidth = 2

Draw the horizon line

frmWorkArea.Line (LineEndX1!, LineEndY1!)-(LineEndX2!, LineEndY2!)

Draw perspective lines

Determine center of a line parallel to the horizon line, one formwidth below it

Ctr2lineX! = -SinRoll! * Rollunit! + CtrlineX!

Ctr2lineY! = CosRoll! * Rollunit! + CtrlineY!

Determine line ends

Per1endX1! = CtrlineX! - (5 * Linespacefar! * CosRoll!) - TurnCtl! * ➡
 Linespacefar! * CosRoll! / 5

Per1endX2! = Ctr2lineX! - (5 * Linespacenear! * CosRoll!) - TurnCtl! * ➡
 Linespacenear! * CosRoll! / 5

Per1endY1! = CtrlineY! - (5 * Linespacefar! * SinRoll!) - TurnCtl! * ➡
 Linespacefar! * SinRoll! / 5

```

Per1endY2! = Ctr2lineY! - (5 * Linespacenear! * SinRoll!) - TurnCtl! * ➡
Linespacenear! * SinRoll! / 5
Per2endX1! = CtrlineX! - (4 * Linespacefar! * CosRoll!) - TurnCtl! * ➡
Linespacefar! * CosRoll! / 5
Per2endX2! = Ctr2lineX! - (4 * Linespacenear! * CosRoll!) - TurnCtl! * ➡
Linespacenear! * CosRoll! / 5
Per2endY1! = CtrlineY! - (4 * Linespacefar! * SinRoll!) - TurnCtl! * ➡
Linespacefar! * SinRoll! / 5
Per2endY2! = Ctr2lineY! - (4 * Linespacenear! * SinRoll!) - TurnCtl! * ➡
Linespacenear! * SinRoll! / 5
Per3endX1! = CtrlineX! - (3 * Linespacefar! * CosRoll!) - TurnCtl! * ➡
Linespacefar! * CosRoll! / 5
Per3endX2! = Ctr2lineX! - (3 * Linespacenear! * CosRoll!) - TurnCtl! * ➡
Linespacenear! * CosRoll! / 5
Per3endY1! = CtrlineY! - (3 * Linespacefar! * SinRoll!) - TurnCtl! * ➡
Linespacefar! * SinRoll! / 5
Per3endY2! = Ctr2lineY! - (3 * Linespacenear! * SinRoll!) - TurnCtl! * ➡
Linespacenear! * SinRoll! / 5
Per4endX1! = CtrlineX! - (2 * Linespacefar! * CosRoll!) - TurnCtl! * ➡
Linespacefar! * CosRoll! / 5
Per4endX2! = Ctr2lineX! - (2 * Linespacenear! * CosRoll!) - TurnCtl! * ➡
Linespacenear! * CosRoll! / 5
Per4endY1! = CtrlineY! - (2 * Linespacefar! * SinRoll!) - TurnCtl! * ➡
Linespacefar! * SinRoll! / 5
Per4endY2! = Ctr2lineY! - (2 * Linespacenear! * SinRoll!) - TurnCtl! * ➡
Linespacenear! * SinRoll! / 5
Per5endX1! = CtrlineX! - (1 * Linespacefar! * CosRoll!) - TurnCtl! * ➡
Linespacefar! * CosRoll! / 5
Per5endX2! = Ctr2lineX! - (1 * Linespacenear! * CosRoll!) - TurnCtl! * ➡
Linespacenear! * CosRoll! / 5
Per5endY1! = CtrlineY! - (1 * Linespacefar! * SinRoll!) - TurnCtl! * ➡
Linespacefar! * SinRoll! / 5
Per5endY2! = Ctr2lineY! - (1 * Linespacenear! * SinRoll!) - TurnCtl! * ➡
Linespacenear! * SinRoll! / 5
Per6endX1! = CtrlineX! - TurnCtl! * Linespacefar! * CosRoll! / 5
Per6endX2! = Ctr2lineX! - TurnCtl! * Linespacenear! * CosRoll! / 5
Per6endY1! = CtrlineY! - TurnCtl! * Linespacefar! * SinRoll! / 5
Per6endY2! = Ctr2lineY! - TurnCtl! * Linespacenear! * SinRoll! / 5
Per7endX1! = CtrlineX! + (1 * Linespacefar! * CosRoll!) - TurnCtl! * ➡
Linespacefar! * CosRoll! / 5

```

```

Per7endX2! = Ctr2lineX! + (1 * Linespacenear! * CosRoll!) - TurnCtl! * ➤
Linespacenear! * CosRoll! / 5
Per7endY1! = CtrlineY! + (1 * Linespacefar! * SinRoll!) - TurnCtl! * ➤
Linespacefar! * SinRoll! / 5
Per7endY2! = Ctr2lineY! + (1 * Linespacenear! * SinRoll!) - TurnCtl! * ➤
Linespacenear! * SinRoll! / 5
Per8endX1! = CtrlineX! + (2 * Linespacefar! * CosRoll!) - TurnCtl! * ➤
Linespacefar! * CosRoll! / 5
Per8endX2! = Ctr2lineX! + (2 * Linespacenear! * CosRoll!) - TurnCtl! * ➤
Linespacenear! * CosRoll! / 5
Per8endY1! = CtrlineY! + (2 * Linespacefar! * SinRoll!) - TurnCtl! * ➤
Linespacefar! * SinRoll! / 5
Per8endY2! = Ctr2lineY! + (2 * Linespacenear! * SinRoll!) - TurnCtl! * ➤
Linespacenear! * SinRoll! / 5
Per9endX1! = CtrlineX! + (3 * Linespacefar! * CosRoll!) - TurnCtl! * ➤
Linespacefar! * CosRoll! / 5
Per9endX2! = Ctr2lineX! + (3 * Linespacenear! * CosRoll!) - TurnCtl! * ➤
Linespacenear! * CosRoll! / 5
Per9endY1! = CtrlineY! + (3 * Linespacefar! * SinRoll!) - TurnCtl! * ➤
Linespacefar! * SinRoll! / 5
Per9endY2! = Ctr2lineY! + (3 * Linespacenear! * SinRoll!) - TurnCtl! * ➤
Linespacenear! * SinRoll! / 5
Per10endX1! = CtrlineX! + (4 * Linespacefar! * CosRoll!) - TurnCtl! * ➤
Linespacefar! * CosRoll! / 5
Per10endX2! = Ctr2lineX! + (4 * Linespacenear! * CosRoll!) - TurnCtl! * ➤
Linespacenear! * CosRoll! / 5
Per10endY1! = CtrlineY! + (4 * Linespacefar! * SinRoll!) - TurnCtl! * ➤
Linespacefar! * SinRoll! / 5
Per10endY2! = Ctr2lineY! + (4 * Linespacenear! * SinRoll!) - TurnCtl! * ➤
Linespacenear! * SinRoll! / 5
Per11endX1! = CtrlineX! + (5 * Linespacefar! * CosRoll!) - TurnCtl! * ➤
Linespacefar! * CosRoll! / 5
Per11endX2! = Ctr2lineX! + (5 * Linespacenear! * CosRoll!) - TurnCtl! * ➤
Linespacenear! * CosRoll! / 5
Per11endY1! = CtrlineY! + (5 * Linespacefar! * SinRoll!) - TurnCtl! * ➤
Linespacefar! * SinRoll! / 5
Per11endY2! = Ctr2lineY! + (5 * Linespacenear! * SinRoll!) - TurnCtl! * ➤
Linespacenear! * SinRoll! / 5

```

Draw perspective lines

frmWorkArea.Line (Per1endX1!, Per1endY1!)-(Per1endX2!, Per1endY2!)

```

frmWorkArea.Line (Per2endX1!, Per2endY1!)-(Per2endX2!, Per2endY2!)
frmWorkArea.Line (Per3endX1!, Per3endY1!)-(Per3endX2!, Per3endY2!)
frmWorkArea.Line (Per4endX1!, Per4endY1!)-(Per4endX2!, Per4endY2!)
frmWorkArea.Line (Per5endX1!, Per5endY1!)-(Per5endX2!, Per5endY2!)
frmWorkArea.Line (Per6endX1!, Per6endY1!)-(Per6endX2!, Per6endY2!)
frmWorkArea.Line (Per7endX1!, Per7endY1!)-(Per7endX2!, Per7endY2!)
frmWorkArea.Line (Per8endX1!, Per8endY1!)-(Per8endX2!, Per8endY2!)
frmWorkArea.Line (Per9endX1!, Per9endY1!)-(Per9endX2!, Per9endY2!)
frmWorkArea.Line (Per10endX1!, Per10endY1!)-(Per10endX2!, Per10endY2!)
frmWorkArea.Line (Per11endX1!, Per11endY1!)-(Per11endX2!, Per11endY2!)

```

Adjust range to target based on time, target closing

NOTE: 2 pixels = 1 Mil, 1 Mil = 0.05625 degrees

If TimeCounter! <= 2 Then

TgtRange! = MAXRANGE! *initial target range*

Else

Own craft chasing on target @ 120 kts (nominal) [1 kt = 1.6878 ft/sec]

Target running at 60 kts or 101.3 ft/sec

Own craft closing at 60 kts or 101.3 ft/sec if Speed! = 120 kts

TgtRange! = (TgtRange! - (((Speed! * Cos(DEG_TO_RAD * Abs(Pitch!)) ➡
/ 5.33) * 1.6878) - 101.3) * Laptimer!))

End If

Determine target size based upon slant range to target

TgtRadius! = 2 * (TGTSIZE! * RangeScale(TgtRange!))

Adjust position of target for change in roll

TgtCenterX! = CtrlineX! + (Cos((TgtAngle! + Roll!) * DEG_TO_RAD!)) * Tgtunit!

TgtCenterY! = CtrlineY! + (Sin((TgtAngle! + Roll!) * DEG_TO_RAD!)) * Tgtunit!

Scale target movement as a factor of range

MoveScale! = RangeScale(TgtRange!)

If no target evasive movement is desired set Movestop\$ to "Yes" [used for debugging only]

Dim Movestop\$

Movestop\$ = "No"

If Movestop\$ <> "Yes" Then

Move target in an evasive manner

Movetimer! = Movetimer! + 2

If Movetimer! > 360 Then Movetimer! = 0

If TimeCounter! > 2 Then

```

TgtRateX! = MOVEXCONST! * (Cos(Movetimer! * ➤
DEG_TO_RAD!)) * Laptimer! * MoveScale! * Direction1%
TgtRateY! = MOVEYCONST! * (Sin(Movetimer! * ➤
DEG_TO_RAD!)) * Laptimer! * MoveScale! * Direction2%
TgtCenterX! = TgtCenterX! + TgtRateX! * CosRoll! + TgtRateY! ➤
* SinRoll!
TgtCenterY! = TgtCenterY! + TgtRateX! * SinRoll! + TgtRateY! ➤
* CosRoll!

```

End If

End If

If no target relative movement is desired set Movestop2\$ to "Yes" [used for debugging only]

```

Dim Movestop2$
Movestop2$ = "No"
If Movestop2$ <> "Yes" Then

```

If were in a turn, determine relative motion of target

```

TgtCenterX! = TgtCenterX! + (TURNCONST! * TurnRate! * Laptimer! ➤
* CosRoll!)
TgtCenterY! = TgtCenterY! + (TURNCONST! * TurnRate! * Laptimer! ➤
* SinRoll!)

```

If we have pitch, determine relative motion of target

```

TgtCenterX! = TgtCenterX! + (PITCHCONST! * Laptimer! * Pitch! * ➤
SinRoll!)
TgtCenterY! = TgtCenterY! - (PITCHCONST! * Laptimer! * Pitch! * ➤
CosRoll!)

```

End If

Show roll rate

```

frmWorkArea.DrawWidth = 3
frmWorkArea.Line (frmCenterX! - 155, 12)-(frmCenterX! - 155, 29), RGB ➤
(255, 255, 0)
frmWorkArea.Line (frmCenterX! + 155, 12)-(frmCenterX! + 155, 29), RGB ➤
(255, 255, 0)
frmWorkArea.DrawWidth = 12
If Sgn(RollRate!) > 0 Then
    temp! = (105) * Abs(RollRate!) / MAXROLLRATE!

```

FRMSIM.FRM - 14

```
    frmWorkArea.Line (frmCenterX! - 90 / 2, 20.5)-(frmCenterX! - 90 / 2 - ➡  
    temp!, 20.5), RGB(255, 255, 0)  
ElseIf Sgn(RollRate!) < 0 Then  
    temp! = (105) * Abs(RollRate!) / MAXROLLRATE!  
    frmWorkArea.Line (frmCenterX! + 90 / 2, 20.5)-(frmCenterX! + 90 / 2 + ➡  
    temp!, 20.5), RGB(255, 255, 0)  
Else  
End If  
frmWorkArea.DrawWidth = 3
```

Show pitch rate

```
    frmWorkArea.DrawWidth = 3  
    frmWorkArea.Line (frmWidth! - 83 / 2 + 5, frmCenterY! - 120)-(frmWidth! - 83 ➡  
    / 2 - 5, frmCenterY! - 120), RGB(255, 255, 0)  
    frmWorkArea.Line (frmWidth! - 83 / 2 + 5, frmCenterY! + 125)-(frmWidth! - 83 ➡  
    / 2 - 5, frmCenterY! + 125), RGB(255, 255, 0)  
    frmWorkArea.DrawWidth = 12  
    If Sgn(PitchRate!) < 0 Then  
        temp! = (100) * Abs(PitchRate!) / MAXPITCHRATE!  
        frmWorkArea.Line (frmWidth! - 83 / 2, frmCenterY! - 30)-(frmWidth! - ➡  
        83 / 2, frmCenterY! - 30 - temp!), RGB(255, 255, 0)  
    ElseIf Sgn(PitchRate!) > 0 Then  
        temp! = (100) * Abs(PitchRate!) / MAXPITCHRATE!  
        frmWorkArea.Line (frmWidth! - 83 / 2, frmCenterY! + 20)-(frmWidth! - ➡  
        83 / 2, frmCenterY! + 20 + temp!), RGB(255, 255, 0)  
    Else  
    End If  
    frmWorkArea.DrawWidth = 3
```

*Compute CCIP reticle position, CCIPX! & CCIPY! adjusted for various flight conditions
CCIP values are computed for a standard 1 TOF for a PGU-28 = 2725 ft & a 2 mil gun
elevation*

1 mil = 2 pixels

CCIP starts at the center aiming position

CCIPX! = AimCenterX!

CCIPY! = AimCenterY!

CCIP adjusts for firing at an acute angle of pitch

Acute! = BulletDrop(TimeOfFlight(2725 / AdjustRangeForElevation(Pitch! / ➡
5.333), Speed!)) * RangeScale(2725)

Acute! = Acute! - BulletDrop(TimeOfFlight(2725, Speed!)) * RangeScale(2725)

CCIPY! = CCIPY! - 2 * Acute!

CCIP adjusts for roll

CCIPX! = CCIPX! + (BulletDrop(1) * RangeScale(2725)) * 2 * SinRoll!
 CCIPY! = CCIPY! + Abs((BulletDrop(1) * RangeScale(2725)) * 2 * SinRoll!)

CCIP adjusts for speed

CCIPX! = CCIPX! + 2 * (BulletDrop(TimeOfFlight(2725, Speed!)) -
 BulletDrop(TimeOfFlight(2725, 120))) * RangeScale(2725) * SinRoll!
 CCIPY! = CCIPY! + 2 * (BulletDrop(TimeOfFlight(2725, Speed!)) -
 BulletDrop(TimeOfFlight(2725, 120))) * RangeScale(2725) * CosRoll!

CCIP adjusts for turnrate

CCIPY! = CCIPY! + ((TurnRateModel(-Roll!, Speed!) * TimeOfFlight(2725, Speed!)) * 1000 * 2) / 2725

CCIP adjusts for pitchrate

rc% = 16.7 * ((PitchRate! / 5.333) / .05625) / 1000
 CCIPX! = CCIPX! + (((rc% * TimeOfFlight(2725, Speed!)) * 1000 * 2) / 2725) * SinRoll!
 CCIPY! = CCIPY! + (((rc% * TimeOfFlight(2725, Speed!)) * 1000 * 2) / 2725) * CosRoll!

Compute lead point

TOF! = TimeOfFlight(TgtRange!, Speed!)
 Drop! = 2 * BulletDrop(TOF!) * MoveScale!

If Reticletype\$ = "Fixedlead" Then

Lead for target rate

LeadX! = TgtCenterX! + ((TgtRateX! * CosRoll! + TgtRateY! * SinRoll!) * TOF! / Laptimer!) - Drop! * SinRoll!
 LeadY! = TgtCenterY! + ((TgtRateX! * SinRoll! + TgtRateY! * CosRoll!) * TOF! / Laptimer!) + Drop! * CosRoll!

Lead adjusts for firing at an acute angle of pitch

Acute! = BulletDrop(TimeOfFlight(TgtRange! / AdjustRangeForElevation(Pitch! / 5.333), Speed!)) * RangeScale(TgtRange!)
 Acute! = Acute! - BulletDrop(TOF!) * RangeScale(TgtRange!)
 LeadY! = LeadY! + 2 * Acute!

Lead adjusts for roll

LeadX! = LeadX! - (BulletDrop(TOF!) * RangeScale(TgtRange!)) * 2 * SinRoll!
 LeadY! = LeadY! - Abs((-BulletDrop(TOF!) * RangeScale(TgtRange!)) * 2 * SinRoll!)

Lead adjusts for speed

LeadX! = LeadX! - 2 * (BulletDrop(TOF!) - BulletDrop(TimeOfFlight(➤
TgtRange!, 120))) * RangeScale(TgtRange!) * SinRoll!

LeadY! = LeadY! - 2 * (BulletDrop(TOF!) - BulletDrop(TimeOfFlight(➤
TgtRange!, 120))) * RangeScale(TgtRange!) * CosRoll!

Lead adjusts for turnrate

LeadY! = LeadY! - ((TurnRateModel(-Roll!, Speed!) * TOF!) * 1000 * 2) ➤
/ TgtRange!

Lead adjusts for pitchrate

rc% = 16.7 * ((PitchRate! / 5.333) / .05625) / 1000

LeadX! = LeadX! - (((rc% * TOF!) * 1000 * 2) / TgtRange!) * SinRoll!

LeadY! = LeadY! - (((rc% * TOF!) * 1000 * 2) / TgtRange!) * CosRoll!

End If

If Reticletype\$ = "CCIP" Then

Lead for target rate

LeadX! = TgtCenterX! + ((TgtRateX! * CosRoll! + TgtRateY! * ➤
SinRoll!) * TOF! / Laptimer!) - Drop! * SinRoll!

LeadY! = TgtCenterY! + ((TgtRateX! * SinRoll! + TgtRateY! * ➤
CosRoll!) * TOF! / Laptimer!) + Drop! * CosRoll!

End If

Draw target circle to background

Set to filled circle

frmWorkArea.FillStyle = 0

Draw the target circle

Change color of target to blue if on target and in practice mode

If Datawrite\$ = "No" Then

frmWorkArea.FillColor = &HC000000

If Reticletype\$ = "Fixed" Then fixed reticle

If Sqr((AimCenterX! - TgtCenterX!) ^ 2 + (AimCenterY! - ➤
TgtCenterY!) ^ 2) <= TgtRadius! Then frmWorkArea.Circle ➤

(TgtCenterX!, TgtCenterY!), TgtRadius!, RGB(0, 0, 255)

If Sqr((AimCenterX! - TgtCenterX!) ^ 2 + (AimCenterY! - ➤
TgtCenterY!) ^ 2) > TgtRadius! Then frmWorkArea.FillColor = ➤

&HFF&

```

    If Sqr((AimCenterX! - TgtCenterX!) ^ 2 + (AimCenterY! - TgtCenterY!) ^ 2) > TgtRadius! Then frmWorkArea.Circle (TgtCenterX!, TgtCenterY!), TgtRadius!, RGB(255, 0, 0)
ElseIf Reticletype$ = "Fixedlead" Then fixed reticle, pull lead
    If Sqr((AimCenterX! - LeadX!) ^ 2 + (AimCenterY! - LeadY!) ^ 2) <= TgtRadius! Then frmWorkArea.Circle (TgtCenterX!, TgtCenterY!), TgtRadius!, RGB(0, 0, 255)
    If Sqr((AimCenterX! - LeadX!) ^ 2 + (AimCenterY! - LeadY!) ^ 2) > TgtRadius! Then frmWorkArea.FillColor = &HFF&
    If Sqr((AimCenterX! - LeadX!) ^ 2 + (AimCenterY! - LeadY!) ^ 2) > TgtRadius! Then frmWorkArea.Circle (TgtCenterX!, TgtCenterY!), TgtRadius!, RGB(255, 0, 0)
Else CCIP reticle
    If Sqr((CCIPX! - LeadX!) ^ 2 + (CCIPY! - LeadY!) ^ 2) <= TgtRadius! Then frmWorkArea.Circle (TgtCenterX!, TgtCenterY!), TgtRadius!, RGB(0, 0, 255)
    If Sqr((CCIPX! - LeadX!) ^ 2 + (CCIPY! - LeadY!) ^ 2) > TgtRadius! Then frmWorkArea.FillColor = &HFF&
    If Sqr((CCIPX! - LeadX!) ^ 2 + (CCIPY! - LeadY!) ^ 2) > TgtRadius! Then frmWorkArea.Circle (TgtCenterX!, TgtCenterY!), TgtRadius!, RGB(255, 0, 0)
End If
Else

    frmWorkArea.FillColor = &HFF&
    frmWorkArea.Circle (TgtCenterX!, TgtCenterY!), TgtRadius!, RGB (255, 0, 0)
End If

Draw reticle to background
    Set draw width
    frmWorkArea.DrawWidth = 3
    Set to no fill
    frmWorkArea.FillStyle = 1
    Determine reticle type
    If Reticletype$ = "Fixed" Then fixed reticle

        frmWorkArea.Circle (AimCenterX!, AimCenterY!), 40, RGB(255, 255, 0)
        frmWorkArea.Circle (AimCenterX!, AimCenterY!), 1, RGB(255, 255, 0)
        frmWorkArea.Line (AimCenterX! - 50, AimCenterY!)-(AimCenterX! - 30, AimCenterY!), RGB(255, 255, 0)

```

```

frmWorkArea.Line (AimCenterX! + 30, AimCenterY!)-(AimCenterX! + ➤
50, AimCenterY!), RGB(255, 255, 0)
frmWorkArea.Line (AimCenterX!, AimCenterY! - 50)-(AimCenterX!, ➤
AimCenterY! - 30), RGB(255, 255, 0)
frmWorkArea.Line (AimCenterX!, AimCenterY! + 30)-(AimCenterX!, ➤
AimCenterY! + 50), RGB(255, 255, 0)
ElseIf Reticletype$ = "Fixedlead" Then fixed reticle, pull lead
If LeadHelp$ = "Show" Then frmWorkArea.Circle (LeadX!, LeadY!), 2, ➤
RGB(255, 255, 255)
frmWorkArea.Circle (AimCenterX!, AimCenterY!), 40, RGB(255, 255, 0)
frmWorkArea.Circle (AimCenterX!, AimCenterY!), 1, RGB(255, 255, 0)
frmWorkArea.Line (AimCenterX! - 50, AimCenterY!)-(AimCenterX! - ➤
30, AimCenterY!), RGB(255, 255, 0)
frmWorkArea.Line (AimCenterX! + 30, AimCenterY!)-(AimCenterX! + ➤
50, AimCenterY!), RGB(255, 255, 0)
frmWorkArea.Line (AimCenterX!, AimCenterY! - 50)-(AimCenterX!, ➤
AimCenterY! - 30), RGB(255, 255, 0)
frmWorkArea.Line (AimCenterX!, AimCenterY! + 30)-(AimCenterX!, ➤
AimCenterY! + 50), RGB(255, 255, 0)
Else CCIP reticle
frmWorkArea.Circle (AimCenterX!, AimCenterY!), 1, RGB(0, 255, 0)
frmWorkArea.Line (AimCenterX! - 50, AimCenterY!)-(AimCenterX! - ➤
30, AimCenterY!), RGB(0, 255, 0)
frmWorkArea.Line (AimCenterX! + 30, AimCenterY!)-(AimCenterX! + ➤
50, AimCenterY!), RGB(0, 255, 0)
If LeadHelp$ = "Show" Then frmWorkArea.Circle (LeadX!, LeadY!), 2, ➤
RGB(255, 255, 255)
frmWorkArea.Circle (CCIPX!, CCIPY!), 40, RGB(255, 255, 0)
frmWorkArea.Circle (CCIPX!, CCIPY!), 2, RGB(255, 255, 0)
frmWorkArea.Line (CCIPX! - 50, CCIPY!)-(CCIPX! - 30, CCIPY!), ➤
RGB(255, 255, 0)
frmWorkArea.Line (CCIPX! + 30, CCIPY!)-(CCIPX! + 50, CCIPY!), ➤
RGB(255, 255, 0)
frmWorkArea.Line (CCIPX!, CCIPY! - 50)-(CCIPX!, CCIPY! - 30), ➤
RGB(255, 255, 0)
frmWorkArea.Line (CCIPX!, CCIPY! + 30)-(CCIPX!, CCIPY! + 50), ➤
RGB(255, 255, 0)
End If

```

Display speed

```
lblRndctr.ForeColor = &H80000008
lblRndctr.Caption = Int(Speed!)
```

Compute Run Score

```
If Reticletype$ = "Fixed" Then fixed reticle
    If Sqr((AimCenterX! - TgtCenterX!) ^ 2 + (AimCenterY! - TgtCenterY!) ^ 2) <= TgtRadius! Then Score! = Score! + Laptimer!
ElseIf Reticletype$ = "Fixedlead" Then fixed reticle, pull lead
    If Sqr((AimCenterX! - LeadX!) ^ 2 + (AimCenterY! - LeadY!) ^ 2) <= TgtRadius! Then Score! = Score! + Laptimer!
Else CCIP reticle
    If Sqr((CCIPX! - LeadX!) ^ 2 + (CCIPY! - LeadY!) ^ 2) <= TgtRadius! Then Score! = Score! + Laptimer!
End If
```

```
lblScore.Caption = "Score= " & Int(Score! * 100 / TimeCounter!) & "%"
```

Write data to file

Determine x and y coordinate aiming error

```
If Reticletype$ = "Fixed" Then
    MissX! = AimCenterX! - TgtCenterX!
    MissY! = AimCenterY! - TgtCenterY!
ElseIf Reticletype$ = "Fixedlead" Then
    MissX! = AimCenterX! - LeadX!
    MissY! = AimCenterY! - LeadY!
Else CCIP
    MissX! = CCIPX! - LeadX!
    MissY! = CCIPY! - LeadY!
End If
```

```
If Joystick$ = "On" Then
    Calculate percent of joystick travel
    JoyTravelX! = jX& / joyMaxX&
    JoyTravelY! = jY& / joyMaxY&
End If
```

Break output into 100 foot bins

```
DistanceBin! = Int(TgtRange! / 100)
DistanceBin! = DistanceBin! * 100
```

```
If Datawrite$ = "Yes" And TimeCounter! > 2 Then
    Write #FileNum, TimeCounter!, Laptimer!, DistanceBin!, TgtRange!, ➡
    TgtRadius!, MissX!, MissY!, Pitch! / 5.333, PitchRate! / 5.333, Roll!, ➡
    RollRate!, Speed!, JoyTravelX!, JoyTravelY!
End If
```

Store values

```
OldPitch! = Pitch!
OldRoll! = Roll!
```

Stop run if target reaches min range

```
If TgtRange! <= MINRANGE! Then
    Reticlecounter% = Reticlecounter% + 1
    tmrRun.Enabled = False
    lblRndctr.Caption = "STOP"
    frmWorkArea.lblScore.Visible = True
    If Datawrite$ = "Yes" Then Write #FileNum,
    If Datawrite$ = "Yes" Then Write #FileNum, "Score= ", Int(Score! * 100 ➡
    / TimeCounter!)
    If Datawrite$ = "Yes" Then Write #FileNum,
    If Reticlecounter% > 3 Then GoTo reconfigure:
    frmContinue.Show Modal
End If
```

GoTo skipstep:

reconfigure:

```
If Datawrite$ = "Yes" Then Call EndProgram
Reticlecounter% = 1
frmContinue.Show Modal
frmConfigure2.Show Modal
frmStart.Show Modal
```

skipstep:

End Sub

FRMSTART.FRM - 1

Option Explicit

Dim rc%

Joystick variables

Dim jpos As JOYPOS

Sub cmdExit_Click ()

 Call EndProgram

End Sub

Sub cmdOptions_Click ()

 Unload frmStart
 frmOptions.Show Modal

End Sub

Sub cmdOptions_KeyDown (KeyCode As Integer, Shift As Integer)

 If KeyCode = KEY_RETURN Then
 Unload frmStart
 frmOptions.Show Modal
 End If

End Sub

Sub cmdStart_click ()

 Call Initialize
 frmStart.Hide
 frmWorkArea.Caption = Reticletype\$

 If Datawrite\$ = "Yes" Then
 Write #Filenum,
 Write #Filenum, Reticletype\$
 Write #Filenum,
 End If

FRMSTART.FRM - 2

```
frmWorkArea.tmrRun.Enabled = True
```

```
End Sub
```

```
Sub cmdStart_KeyDown (KeyCode As Integer, Shift As Integer)
```

```
    If KeyCode = KEY_RETURN Then  
        Call Initialize  
        frmStart.Hide  
        frmWorkArea.tmrRun.Enabled = True  
    End If
```

```
End Sub
```

```
Sub cmdStartOver_Click ()
```

```
    Unload frmStart
```

```
    Get position information
```

```
    rc% = joyGetPos%(JOYSTICKID1, jpos)
```

```
    Inform the user of any problem
```

```
    If (rc% <> JOYERR_NOERROR) Then  
        MsgBox "Joystick Driver not installed or Joystick unplugged!", MB_OK ➡  
        + MB_EXCLAMATION, "Joystick Error!"  
        frmConfigure.optJoystick.Value = False  
        frmConfigure.optJoystick.Visible = False  
        frmConfigure.optKeyboard.Value = True  
        Joystick$ = "Off"  
        Keyboard$ = "On"  
    End If
```

```
    frmRegister.Show Modal  
    frmConfigure.Show Modal
```

```
    If Joystick$ = "On" Then  
        frmCalibrate.Show Modal  
    End If
```

```
    frmStart.Show Modal
```

FRMSTART.FRM - 3

End Sub

Sub cmdStartOver_KeyDown (KeyCode As Integer, Shift As Integer)

 If KeyCode = KEY_RETURN Then
 Unload frmStart

Get position information

 rc% = joyGetPos%(JOYSTICKID1, jpos)

Inform the user of any problem

 If (rc% <> JOYERR_NOERROR) Then
 MsgBox "Joystick Driver not installed or Joystick unplugged!", 
 MB_OK + MB_EXCLAMATION, "Joystick Error!"
 frmConfigure.optJoystick.Value = False
 frmConfigure.optJoystick.Visible = False
 frmConfigure.optKeyboard.Value = True
 Joystick\$ = "Off"
 Keyboard\$ = "On"
 End If

 frmRegister.Show Modal
 frmConfigure.Show Modal

 If Joystick\$ = "On" Then
 frmCalibrate.Show Modal
 End If

 frmStart.Show Modal
 End If

End Sub

Sub Form_Load ()

 frmStart.Top = (Screenheight% - 3780) / 2
 frmStart.Left = (Screenwidth% - 2175) / 2

End Sub

FRMSTART.FRM - 4

Sub tmrStart_Timer ()

 If ((jpos.Buttons And JOY_BUTTON2) = JOY_BUTTON2) Then cmdStart_click

End Sub

SIM.BAS - 1

Option Explicit

Function AdjustRangeForElevation (X!)

X! input = elevation angle from aircraft to target in degrees (up or down)

*Y! output = ratio of slant range to horizontal range from gun to target in feet
(around 2 at 60 degrees up or down and 1 at 0 degrees)*

Dim Y!

$Y! = (1 / \cos(X! * \text{DEG_TO_RAD!})) * (1 - (\tan(2 / \text{DEG_TO_MIL!}) * \tan(X! * \text{DEG_TO_RAD!})))$

AdjustRangeForElevation = Y!

End Function

Function BulletDrop (X!)

X! input = time of flight (TOF) in seconds

Y! output = bullet drop in feet

Dim Y!

$Y! = (-.2542 * (X! ^ 4)) + (2.4298 * (X! ^ 3)) - (17.14 * (X! ^ 2)) + (6.6498 * X! - .0064$

BulletDrop = Y!

End Function

Sub EndProgram ()

End

End Sub

Sub Initialize ()

Dim rc%

frmworkArea.lblScore.Visible = False

SIM.BAS - 2

Write data header to file

```
If Datawrite$ = "Yes" Then
    Print #Filenum, "TimeCounter!, Laptimer!, DistanceBin!, TgtRange!, ➡
    TgtRadius!,MissX!, MissY!, Pitch!, PitchRate!, Roll!, RollRate!, Speed!, ➡
    JoyTravelX!,JoyTravelY!"
End If
```

Reinitialize horizon line

```
CtrlX! = frmCenterX!
CtrlY! = frmCenterY!
LineEndX1! = CtrlX! - (Rollunit! / 2)
LineEndX2! = CtrlX! + (Rollunit! / 2)
LineEndY1! = frmCenterY!
LineEndY2! = frmCenterY!
```

Initialize variables

```
BeginTime! = Timer
MasterClock! = Timer
TimeCounter! = 0
Laptimer! = 0
Linecounter! = 0
Movetimer! = 0
Roll! = 0
RollRate! = 0
Pitch! = 0
PitchRate! = 0
TurnRate! = 0
Altitudedelta! = 0
Speed! = 120
OldRoll! = Roll!
OldPitch! = Pitch!
TgtRateX! = 0
TgtRateY! = 0
LeadX! = 0
LeadY! = 0
Score! = 0
```

Compute initial values

```
TgtRange! = MAXRANGE!
TOF! = TimeOfFlight(TgtRange!, Speed!)
Drop! = BulletDrop(TOF!) * RangeScale(TgtRange!) * 2
```

Choose Reticle Type & Initial Position

Select Case (Reticlecounter%)

Case 1

Reticletype\$ = "Fixed"
 AimCenterX! = frmCenterX!
 AimCenterY! = frmCenterY! + 25
 TgtCenterX! = AimCenterX!
 TgtCenterY! = AimCenterY!

Case 2

Reticletype\$ = "Fixedlead"
 AimCenterX! = frmCenterX!
 AimCenterY! = frmCenterY! + 25
 TgtCenterX! = AimCenterX!
 TgtCenterY! = AimCenterY!
 LeadY! = Drop!
 TgtCenterY! = TgtCenterY! - LeadY!

Case 3

Reticletype\$ = "CCIP"
 AimCenterX! = frmCenterX!
 AimCenterY! = frmCenterY! + 25
 TgtCenterX! = AimCenterX!
 TgtCenterY! = AimCenterY!
 LeadY! = Drop!
 TgtCenterY! = TgtCenterY! - LeadY!

End Select

Initialize label captions

frmworkArea.lblPitch.Caption = Pitch! / 5.333
 frmworkArea.lblRoll.Caption = Roll!

Set initial target direction

Randomize
 rc% = Int(2 * Rnd - 1)
 If Sgn(rc%) >= 0 Then
 Direction1% = 1
 Else
 Direction1% = -1
 End If
 Randomize
 rc% = Int(2 * Rnd - 1)
 If Sgn(rc%) >= 0 Then

SIM.BAS - 4

```
                Direction2% = 1
Else
                Direction2% = -1
End If
```

End Sub

Function RangeScale (X!)

X! input = slant range to target in feet

Y! output = ratio of movement as a factor of range in thousands of feet

```
Dim Y!
```

```
Y! = 1000 / X!
RangeScale = Y!
```

End Function

Function TimeOfFlight (X!, Y!)

X! input = range to target in feet

Y! input = airspeed in knots

Z! output = time of flight (TOF) in seconds

```
Dim Z!, Z1!, Z2!, Z3!, Z4!
```

TOF @ 0 KIAS

```
Z1! = (4E-16 * (X! ^ 4)) + (.0000000000001 * (X! ^ 3)) + (.00000003 * (X! ^ 2)) ➡
+ (.0003 * X!) - .00007
```

TOF @ 50 KIAS

```
Z2! = (4E-16 * (X! ^ 4)) + (.0000000000001 * (X! ^ 3)) + (.00000003 * (X! ^ 2)) ➡
+ (.0003 * X!) - .00009
```

TOF @ 100 KIAS

```
Z3! = (4E-16 * (X! ^ 4)) + (.00000000000007 * (X! ^ 3)) + (.00000003 * (X! ^ 2)) ➡
+ (.0003 * X!) - .0001
```

SIM.BAS - 5

TOF @ 150 KIAS

$Z4! = (3E-16 * (X! ^ 4)) + (.000000000002 * (X! ^ 3)) + (.000000002 * (X! ^ 2)) \rightarrow$
 $+ (.0003 * X!) - .0209$

If Y! > 0 And Y! <= 50 Then

$Z! = ((\text{Abs}(Z1! - Z2!)) * ((50 - Y!) / 50)) + Z2!$

ElseIf Y! > 50 And Y! <= 100 Then

$Z! = ((\text{Abs}(Z2! - Z3!)) * ((100 - Y!) / 50)) + Z3!$

ElseIf Y! > 100 And Y! <= 150 Then

$Z! = ((\text{Abs}(Z3! - Z4!)) * ((150 - Y!) / 50)) + Z4!$

End If

TimeOfFlight = Z!

End Function

Function TurnRateModel (X!, Y!)

X! input = roll angle in degrees

Y! input = airspeed in knots

Z! ouput = lateral velocity of round in fps due to turn rate

Muzzle to point of rotation distance is 16.7 feet

1 mil = .05625 deg

Dim Z!

$Z! = -\text{Sgn}(X!) * (((.0006 + .000015 * (Y! - 90))) * X! ^ 2) + (.2057 - (.001815 * \rightarrow$
 $(Y! - 90))) * X!$

$Z! = (16.7) * (Z! / .05625) / 1000$

TurnRateModel = Z!

End Function

SMGLOBAL.BAS - 1

Option Explicit

VB Stuff

Global Const Modal = 1

Registration variables

Global Lastname\$	<i>Tracker's last name</i>
Global FlightHours\$	<i>Tracker's flight hours</i>
Global Qual\$	<i>Tracker's qualification</i>
Global Airtype\$	<i>Tracker's community</i>
Global Today	<i>Extracts today's date and time</i>
Global Filenum	<i>Used to establish data file</i>

Size and position variables

Global ScreenHeight%	<i>As it says</i>
Global ScreenWidth%	<i>As it says</i>
Global frmHeight!	<i>Viewable form height (scaleheight)</i>
Global frmWidth!	<i>Viewable form width (scalewidth)</i>
Global frmCenterX!	<i>Lateral center of view window</i>
Global frmCenterY!	<i>Vertical center of view window</i>
Global Rollunit!	<i>Establishes line length for horizon</i>
Global AimCenterX!	<i>Reticle Center</i>
Global AimCenterY!	<i>Reticle Center</i>
Global CCIPX!	<i>Used to compute CCIP reticle position</i>
Global CCIPY!	<i>Used to compute CCIP reticle position</i>
Global Acute!	<i>Used to adjust CCIP for firing at an acute angle of pitch</i>
Global TgtCenterX!	<i>Target center</i>
Global TgtCenterY!	<i>Target center</i>
Global Movescale!	<i>Used to compute scaled target move distance</i>
Global TgtRadius!	<i>Target size used to determine range</i>
Global Tgtunit!	<i>Establishes distance of target from center of form</i>
Global TgtAngle!	<i>Angle from horizon line to target center</i>
Global TgtRateX!	<i>Target movement rate in X & Y</i>
Global TgtRateY!	
Global LeadX!	<i>Aiming lead in X & Y coordinates</i>
Global LeadY!	

SMGLOBAL.BAS - 2

Global TOF!	<i>Bullet time of flight</i>
Global Drop!	<i>Bullet drop as a factor of TOF!</i>
Global MissX!	<i>Data variable</i>
Global MissY!	<i>Data variable</i>
Global JoyTravelX!	<i>Data variable</i>
Global DistanceBin!	<i>Data variable</i>
Global JoyTravelY!	<i>Data variable</i>

Line variables

Global CtrlineX!	<i>Center point of horizon line</i>
Global CtrlineY!	<i>Center point of horizon line</i>
Global LineValX!	<i>Horizon line length corrected for roll</i>
Global LineValY!	<i>Horizon line length corrected for roll</i>
Global LineEndX1!	<i>Horizon line end</i>
Global LineEndY1!	<i>Horizon line end</i>
Global LineEndX2!	<i>Horizon line end</i>
Global LineEndY2!	<i>Horizon line end</i>
Global Ctr2lineX!	<i>Center of bottom end line for perspective lines</i>
Global Ctr2lineY!	
Global Per1endX1!	<i>Perspective line #1</i>
Global Per1endY1!	
Global Per1endX2!	
Global Per1endY2!	
Global Per2endX1!	<i>Perspective line #2</i>
Global Per2endY1!	
Global Per2endX2!	
Global Per2endY2!	
Global Per3endX1!	<i>Perspective line #3</i>
Global Per3endY1!	
Global Per3endX2!	
Global Per3endY2!	
Global Per4endX1!	<i>Perspective line #4</i>
Global Per4endY1!	
Global Per4endX2!	
Global Per4endY2!	
Global Per5endX1!	<i>Perspective line #5</i>
Global Per5endY1!	
Global Per5endX2!	
Global Per5endY2!	

Global Per6endX1!	<i>Perspective line #6</i>
Global Per6endY1!	
Global Per6endX2!	
Global Per6endY2!	
Global Per7endX1!	<i>Perspective line #7</i>
Global Per7endY1!	
Global Per7endX2!	
Global Per7endY2!	
Global Per8endX1!	<i>Perspective line #8</i>
Global Per8endY1!	
Global Per8endX2!	
Global Per8endY2!	
Global Per9endX1!	<i>Perspective line #9</i>
Global Per9endY1!	
Global Per9endX2!	
Global Per9endY2!	
Global Per10endX1!	<i>Perspective line #10</i>
Global Per10endY1!	
Global Per10endX2!	
Global Per10endY2!	
Global Per11endX1!	<i>Perspective line #11</i>
Global Per11endY1!	
Global Per11endX2!	
Global Per11endY2!	
Global Turnctr!	<i>Used to generate visual turn rate</i>
Global TurnCtl!	<i>(-5 to 5) keeps track of variation in perspective line position due to turn rate</i>
Global Linespacefar!	<i>Perspective line spacing at the horizon</i>
Global Linespacenear!	<i>Perspective line spacing at the bottom</i>
<i>Flight variables</i>	
Global Pitch!	<i>-Pitch / 5.333 = Pitch in degrees (due to program logic and display scaling)</i>
Global PitchRate!	<i>-Pitch rate / 5.333 = Pitch rate in degrees per second (due to program logic and display scaling)</i>
Global Roll!	<i>Roll in degrees</i>
Global RollRate!	<i>Roll rate in degrees per second</i>
Global SinRoll!	<i>Computed sin of roll angle</i>
Global CosRoll!	<i>Computed cos of roll angle</i>
Global TurnRate!	<i>Turn rate in mils (pixels) per second for display motion</i>

SMGLOBAL.BAS - 4

Global OldPitch!	<i>Stores previous iteration</i>
Global OldRoll!	<i>Stores previous iteration</i>
Global TgtRange!	<i>Slant range from gun to target in feet</i>
Global Altitudedelta!	<i>Vertical difference between gun and target in feet</i>
Global Speed!	<i>Airspeed in knots</i>
Global Direction1%	<i>Random integers (1 or -1) generated to establish target</i>
Global Direction2%	<i>evasive motion</i>

Counter variables

Global BeginTime!	<i>Used to compute program time</i>
Global Linecounter!	<i># of program iterations</i>
Global TimeCounter!	<i>Total run time</i>
Global MasterClock!	<i>Used to compute Laptimer!</i>
Global Laptimer!	<i>Provides time in seconds of each runtime iteration</i>
Global Movetimer!	<i>Provides counter for target movement</i>
Global Counter!	<i>General use</i>
Global Score!	<i>Tracking Score</i>

Flags

Global Reticletype\$	<i>Fixed, Fixedlead, CCIP</i>
Global Reticlecounter%	<i>Steps thru reticle types</i>
Global LeadHelp\$	<i>Show or Noshow</i>
Global Joystick\$	<i>On or Off</i>
Global Keyboard\$	<i>On or Off</i>
Global Recorddata\$	<i>On or Off</i>
Global Practice\$	<i>Yes or No</i>
Global Datawrite\$	<i>Yes or No</i>

Set constants

Global Const DEG_TO_RAD! = (3.141593 / 180)	<i>Degree to radian conversion</i>
Global Const RAD_TO_DEG! = (180 / 3.141593)	<i>Radian to degree conversion</i>
Global Const DEG_TO_MIL! = (6400 / 360)	<i>Degree to milliradian conversion</i>
Global Const MIL_TO_DEG! = (360 / 6400)	<i>Milliradian to degree conversion</i>
Global Const KNOTS_TO_FPS! = 1.68781	<i>Knots to Feet/Second conversion</i>
Global Const ROLLIMIT! = 60	<i>Max Roll in degrees</i>
Global Const PITCHLIMIT! = 45	<i>Max Pitch in degrees</i>
Global Const PITCHCONST! = .5	<i>Factor to move target relative to own aircraft</i>
Global Const TURNCONST! = 40	

SMGLOBAL.BAS - 5

Global Const PITCHCMD! = 40	<i>Amount of change to Pitch Rate for Pitch Command</i>
Global Const ROLLCMD! = 20	<i>Amount of change to Roll Rate for Roll Command</i>
Global Const MOVEXCONST! = 25	<i>Target movement factor in X & Y coordinates</i>
Global Const MOVEYCONST! = 25	
Global Const PITCHSTAB! = .2	<i>Pitch stability factor</i>
Global Const PITCHDAMP! = 5	<i>Pitch rate damping factor</i>
Global Const ROLLSTAB! = .2	<i>Roll stability factor</i>
Global Const ROLLDAMP! = 5	<i>Roll rate damping factor</i>
Global Const MAXPITCHRATE! = 60	<i>Max pitch rate</i>
Global Const MAXROLLRATE! = 25	<i>Max roll rate</i>
Global Const TGTSIZE! = 10	<i>Target radius</i>
Global Const MAXRANGE! = 6500	<i>Max range to target</i>
Global Const MINRANGE! = 500	<i>Min range to target</i>

KeyCode (KeyDown, KeyUp)

Global Const KEY_LBUTTON = &H1
Global Const KEY_RBUTTON = &H2
Global Const KEY_RETURN = &HD
Global Const KEY_SPACE = &H20
Global Const KEY_END = &H23
Global Const KEY_HOME = &H24
Global Const KEY_LEFT = &H25
Global Const KEY_UP = &H26
Global Const KEY_RIGHT = &H27
Global Const KEY_DOWN = &H28
Global Const KEY_PGUP = &H63
Global Const KEY_PGDN = &H69

Joystick error return values

Global Const JOYERR_NOERROR = 0	<i>No error</i>
Global Const JOYERR_PARMS = 165	<i>Bad parameters</i>
Global Const JOYERR_NOCANDO = 166	<i>Request not completed</i>
Global Const JOYERR_UNPLUGGED = 167	<i>Joystick is unplugged</i>

Constants used with JOYPOS

Global Const JOY_BUTTON1 = &H1
Global Const JOY_BUTTON2 = &H2
Global Const JOY_BUTTON3 = &H4

SMGLOBAL.BAS - 6

Global Const JOY_BUTTON4 = &H8

Joystick ID constants

Global Const JOYSTICKID1 = 0

Global Const JOYSTICKID2 = 1

Joystick Capabilities

Type JOYCAPS

Mid As Integer	<i>Manufacturer ID</i>
Pid As Integer	<i>Product ID</i>
Pname As String * 32	<i>Product name</i>
Xmin As Integer	<i>Min and max position values</i>
Xmax As Integer	
Ymin As Integer	
Ymax As Integer	
Zmin As Integer	
Zmax As Integer	
NumButtons As Integer	<i>Number of buttons</i>
PeriodMin As Integer	<i>Minimum message period when captured</i>
PeriodMax As Integer	<i>Maximum message period when captured</i>

End Type

Size of JOYCAP structure (needed for function call)

Global Const JOYCAPSIZE = 54

Joystick Position Information

Type JOYPOS

X As Integer	<i>X, Y, and Z position</i>
Y As Integer	
Z As Integer	
Buttons As Integer	<i>Button states</i>

End Type

Message Box constants

Global Const MB_OK = 0

Global Const MB_EXCLAMATION = 48

SMGLOBAL.BAS - 7

Windows API Joystick functions

```
Declare Function joyGetDevCaps% Lib "MMSystem" (ByVal JoyId%, JCaps As JOYCAPS, ByVal CapSize%)
```

```
Declare Function joyGetPos% Lib "MMSystem" (ByVal JoyId%, JPos As JOYPOS)
```

Global joystick variables

```
Global joyX&, joyY&
```

```
Global joyMaxX&, joyMaxY&
```

```
Global jX&, jY&, mx&, my&
```

```
Global joyCtrX&, joyCtrY&
```

APPENDIX E

BALLISTICS SIMULATION PROGRAM CODE

The Visual Basic® program code for the ballistics simulation is presented below. Program code in Visual Basic® is grouped by the Windows® form which contains the code. Each form which is displayed during the running of the program and each object on that form, such as a text box, check box, or menu bar item, contains the code required to execute its function. Additional code is contained in modules which hold function and subroutine definitions as well as variable and constant definitions. During program development, each group of code is stored as an individual file. Form files are saved with a “.FRM” appendix and modules with a “.BAS” appendix. When the program is completed, Visual Basic® constructs a single executable file using the saved form and module files.

Some lines of code are very long and wrap across more than one line in the listing. A line of code which wraps across more than one line in this listing is indicated by the symbol “➡” at the end of the line. This symbol has no meaning in Visual Basic®, and merely serves to indicate that the next line should be considered a continuation of the line in which the symbol appears. Comments are italicized. Comments reside in the lines of code to aid in writing and debugging the code, but are ignored by the program during run time.

For the purpose of preparing this appendix, the Visual Basic® project files were saved as text files and then formatted using a word processor. Visual Basic® does not support page formatting in its print function.

FRMBALLI.FRM - 1

Option Explicit

Dim Model, Rangebin

Dim RangeText As String

Sub cmdCompVary_Click ()

Dim Counter As Integer, Hit As Integer, Linecount As Integer, rc As Double

Dim XMiss As Double, YMiss As Double, XMiss1 As Double, YMiss1 As Double

Dim XMiss2 As Double, YMiss2 As Double, Rangebin As Integer, Model As Integer

Dim Roll As Double, Pitch As Double, PitchRate As Double

Dim Speed As Double, Turnrate As Double, TOF As Double

Dim Ph1 As Double, Ph2 As Double, Ph3 As Double

Dim Ph4 As Double, Ph5 As Double, Ph6 As Double

Dim Totalhit1 As Integer, Totalhit2 As Integer, Totalhit3 As Integer

Dim Totalhit4 As Integer, Totalhit5 As Integer, Totalhit6 As Integer

Get the next available file number

Filenum = FreeFile

Open the file "COMP.CVS"

Open "D:\COMP.CSV" For Append As Filenum

For Rangebin = 6500 To 500 Step -500

Linecount = 0

Totalhit1 = 0

Totalhit2 = 0

Totalhit3 = 0

Totalhit4 = 0

Totalhit5 = 0

Totalhit6 = 0

Model = 1

Hit = 2

For Counter = 1 To 5000

Linecount = Linecount + 2

Call MissGen(Model, Rangebin, XMiss, YMiss)

Xmiss1 = XMiss

Ymiss1 = YMiss

Call MissGen(Model, Rangebin, XMiss, Ymiss)

```

Xmiss2 = XMiss
Ymiss2 = YMiss
Call Dispersion(XMiss1, XMiss2, YMiss1, YMiss2)
Call ParametersGen(Roll, Pitch, PitchRate, Speed, Turnrate)
TOF = TimeOfFlight(CDbl(Rangebin), Speed)
Roll = Roll * 3.14159265359 / 180

```

Compute hit for all comp

```

Call HitCount(XMiss1, YMiss1, 10, Rangebin, Hit)
If Hit = 1 Then Totalhit1 = Totalhit1 + 1
Call HitCount(XMiss2, YMiss2, 10, Rangebin, Hit)
If Hit = 1 Then Totalhit1 = Totalhit1 + 1
Hit = 2

```

Compute hit for no speed comp

```

rc = (BulletDrop(TOF) - BulletDrop(TimeOfFlight(CDbl( ➡
Rangebin), 120))) * (1000 / Rangebin)
Call HitCount(XMiss1 - rc * Sin(Roll), YMiss1 - rc * Cos(Roll), ➡
10, Rangebin, Hit)
If Hit = 1 Then Totalhit2 = Totalhit2 + 1
Call HitCount(XMiss2 - rc * Sin(Roll), YMiss2 - rc * Cos(Roll), ➡
10, Rangebin, Hit)
If Hit = 1 Then Totalhit2 = Totalhit2 + 1
Hit = 2

```

Compute hit for no pitch comp

```

rc = BulletDrop(TimeOfFlight(Rangebin / ➡
AdjustRangeForElevation(Pitch), Speed)) - BulletDrop(TOF)
rc = rc * 1000 / Rangebin
Call HitCount(XMiss1, YMiss1 + rc, 10, Rangebin, Hit)
If Hit = 1 Then Totalhit3 = Totalhit3 + 1
Call HitCount(XMiss2, YMiss2 + rc, 10, Rangebin, Hit)
If Hit = 1 Then Totalhit3 = Totalhit3 + 1
Hit = 2

```

Compute hit for no roll comp

```

rc = BulletDrop(TOF) * 1000 / Rangebin * Sin(Roll)

```

```

Call HitCount(XMiss1 - rc, YMiss1 - Abs(rc), 10, Rangebin, Hit)
If Hit = 1 Then Totalhit4 = Totalhit4 + 1
Call HitCount(XMiss2 - rc, YMiss2 - Abs(rc), 10, Rangebin, Hit)
If Hit = 1 Then Totalhit4 = Totalhit4 + 1
Hit = 2

```

Compute hit for no pitchrate comp

```

rc = 16.7 * (PitchRate / .05625) / 1000
rc = rc * TOF * 1000 / Rangebin
Call HitCount(XMiss1 - rc * Sin(Roll), YMiss1 - rc * Cos(Roll), ➡
10, Rangebin, Hit)
If Hit = 1 Then Totalhit5 = Totalhit5 + 1
Call HitCount(XMiss2 - rc * Sin(Roll), YMiss2 - rc * Cos(Roll), ➡
10, Rangebin, Hit)
If Hit = 1 Then Totalhit5 = Totalhit5 + 1
Hit = 2

```

Compute hit for no turnrate comp

```

rc = TurnRateModel(-Roll * 180 / 3.14159265359, Speed) * TOF ➡
* 1000 / Rangebin
Call HitCount(XMiss1 - rc * Sin(Roll), YMiss1 - rc * Cos(Roll), ➡
10, Rangebin, Hit)
If Hit = 1 Then Totalhit6 = Totalhit6 + 1
Call HitCount(XMiss2 - rc * Sin(Roll), YMiss2 - rc * Cos(Roll), ➡
10, Rangebin, Hit)
If Hit = 1 Then Totalhit6 = Totalhit6 + 1
Hit = 2

```

Next

```

Ph1 = Totalhit1 / Linecount
Ph2 = Totalhit2 / Linecount
Ph3 = Totalhit3 / Linecount
Ph4 = Totalhit4 / Linecount
Ph5 = Totalhit5 / Linecount
Ph6 = Totalhit6 / Linecount

```

Beep

```

Write #Filenum, "All comp", Rangebin, Int(Ph1 * 100) / 100, 1
Write #Filenum, "No speed", Rangebin, Int(Ph2 * 100) / 100, Int(Ph2 / ➡
Ph1 * 100) / 100
Write #Filenum, "No pitch", Rangebin, Int(Ph3 * 100) / 100, Int(Ph3 / ➡
Ph1 * 100) / 100
Write #Filenum, "No roll", Rangebin, Int(Ph4 * 100) / 100, Int(Ph4 / Ph1 ➡
* 100) / 100
Write #Filenum, "No pitchrate", Rangebin, Int(Ph5 * 100) / 100, Int(Ph5 ➡
/ Ph1 * 100) / 100
Write #Filenum, "No turnrate", Rangebin, Int(Ph6 * 100) / 100, Int(Ph6 / ➡
Ph1 * 100) / 100

```

Next

Close the file

Close Filenum

lblCompVary.Caption = "Done"

End Sub

Sub cmdDraw_Click ()

```

Dim Counter As Integer, XMiss As Double, YMiss As Double, XMiss1 As ➡
Double, YMiss1 As Double
Dim XMiss2 As Double, YMiss2 As Double, size As Integer, Model As Integer, ➡
Range As Integer

```

Get the next available file number

Filenum = FreeFile

Open the file "PICTURE.CVS"

Open "D:\PICTURE.CSV" For Append As Filenum

Range = Val(txtRange.Text)

```

Write #Filenum, "Target Size", "10", "Range", Range
Write #Filenum,

```

For Model = 1 To 3

```

        For Counter = 1 To 1000

            Call MissGen(Model, Range, XMiss, YMiss)
            Xmiss1 = XMiss
            Ymiss1 = YMiss
            Call MissGen(Model, Range, XMiss, YMiss)
            Xmiss2 = XMiss
            Ymiss2 = YMiss
            Call Dispersion(XMiss1, XMiss2, YMiss1, YMiss2)
            Write #Filenum, Model, XMiss1 * Range / 1000, YMiss1 * Range ➡
            / 1000
            Write #Filenum, Model, XMiss2 * Range / 1000, YMiss2 * Range ➡
            / 1000

        Next

        Write #Filenum,

        Beep

    Next

    Close the file
    Close Filenum

    lblDraw.Caption = "Done"

End Sub

Sub cmdModelVary_Click ()

    Dim Counter As Integer, Hit As Integer, Totalhit As Integer, Ph As Double
    Dim XMiss As Double, YMiss As Double, XMiss1 As Double, YMiss1 As Double
    Dim XMiss2 As Double, YMiss2 As Double, Rangebin As Integer, Model As Integer
    Dim Linecount As Integer, AvgProb As Double, Avgcount As Double

    Get the next available file number
    Filenum = FreeFile

    Open the file "MODEL.CVS"
    Open "D:\MODEL.CSV" For Append As Filenum

```

FRMBALLI.FRM - 6

```
AvgProb = 0
Avgcount = 0
Linecount = 0
Totalhit = 0
```

```
For Model = 1 To 3
```

```
    For Rangebin = 6500 To 500 Step -500
```

```
        Avgcount = Avgcount + 1
```

```
        For Counter = 1 To 5000
```

```
            Linecount = Linecount + 2
            Call MissGen(Model, Rangebin, XMiss, YMiss)
            Xmiss1 = XMiss
            Ymiss1 = YMiss
            Call MissGen(Model, Rangebin, XMiss, YMiss)
            Xmiss2 = XMiss
            Ymiss2 = YMiss
            Call Dispersion(XMiss1, XMiss2, YMiss1, YMiss2)
            Call HitCount(XMiss1, YMiss1, 10, Rangebin, Hit)
            If Hit = 1 Then Totalhit = Totalhit + 1
            Call HitCount(XMiss2, YMiss2, 10, Rangebin, Hit)
            If Hit = 1 Then Totalhit = Totalhit + 1
```

```
        Next
```

```
        Ph = Totalhit / Linecount
```

```
        Beep
```

```
        Write #Filenum, Model, Rangebin, Int(Ph * 100) / 100
        AvgProb = AvgProb + Ph
        Totalhit = 0
        Linecount = 0
```

```
    Next
```

```
    AvgProb = AvgProb / Avgcount
    Write #Filenum, Model, "Avg", Int(AvgProb * 100) / 100
```

FRMBALLI.FRM - 7

```
Write #Filenum,  
AvgProb = 0  
Avgcount = 0  
Beep
```

```
Next
```

Close the file

```
Close Filenum
```

```
lblModelVary.Caption = "Done"
```

```
End Sub
```

```
Sub cmdTgtSize_Click ()
```

```
Dim Counter As Integer, Hit As Integer, Totalhit As Integer, Ph As Double  
Dim XMiss As Double, YMiss As Double, XMiss1 As Double, YMiss1 As Double  
Dim XMiss2 As Double, YMiss2 As Double, Rangebin As Integer, Sizechange ➡  
As Integer  
Dim Linecount As Integer, Model As Integer, size As Double
```

Get the next available file number

```
Filenum = FreeFile
```

Open the file "TARGET.CVS"

```
Open "D:\TARGET.CSV" For Append As Filenum
```

```
Linecount = 0  
Totalhit = 0  
size = 1.25  
Model = 1
```

```
For Sizechange = 1 To 5  
size = size * 2
```

```
For Rangebin = 6500 To 500 Step -500
```

```
For Counter = 1 To 5000
```

```
Linecount = Linecount + 2
```

```

        Call MissGen(Model, Rangebin, XMiss, YMiss)
        Xmiss1 = XMiss
        Ymiss1 = YMiss
        Call MissGen(Model, Rangebin, XMiss, YMiss)
        Xmiss2 = XMiss
        Ymiss2 = YMiss
        Call Dispersion(XMiss1, XMiss2, YMiss1, YMiss2)
        Call HitCount(XMiss1, YMiss1, size, Rangebin, Hit)
        If Hit = 1 Then Totalhit = Totalhit + 1
        Call HitCount(XMiss2, YMiss2, size, Rangebin, Hit)
        If Hit = 1 Then Totalhit = Totalhit + 1

    Next

    Ph = Totalhit / Linecount

    Beep

    Write #Filenum, size, Rangebin, Int(Ph * 100) / 100
    Totalhit = 0
    Linecount = 0

Next

    Write #Filenum,

    Beep

Next

    lblTargetSize.Caption = "Done"

Close the file
    Close Filenum

End Sub

Sub Form_Load ()

Get the next available file number
    Filenum = FreeFile

```

FRMBALLI.FRM - 9

Create the file "MODEL.CVS"

Open "D:\MODEL.CSV" For Output As Filenum

Close the file

Close Filenum

Get the next available file number

Filenum = FreeFile

Create the file "TARGET.CVS"

Open "D:\TARGET.CSV" For Output As Filenum

Close the file

Close Filenum

Get the next available file number

Filenum = FreeFile

Create the file "PICTURE.CVS"

Open "D:\PICTURE.CSV" For Output As Filenum

Close the file

Close Filenum

Get the next available file number

Filenum = FreeFile

Create the file "COMP.CVS"

Open "D:\COMP.CSV" For Output As Filenum

Close the file

Close Filenum

End Sub

Sub tmrOutput_Timer ()

lblModelVary.Caption = CStr(Model) + " " + CStr(Rangebin)

End Sub

BALLISTI.BAS - 1

Option Explicit

Global Filenum

Sub Dispersion (XMiss1#, XMiss2#, YMiss1#, YMiss2#)

This subroutine calls the GaussRand () subroutine and creates a pseudo-random projectile distribution around the XMiss and YMiss point.

Input:

Xmiss and Ymiss values from MissGen subroutine

Output: Modified XMiss and YMiss values

Dim Z As Double, T As Double, Y1 As Double, Y2 As Double

Z = .8

Call GaussRand(3.75, Z, Y1, Y2)

Xmiss1 = XMiss1 + Y1

Xmiss2 = XMiss2 + Y2

Call GaussRand(3.75, Z, Y1, Y2)

Ymiss1 = YMiss1 + Y1

Ymiss2 = YMiss2 + Y2

End Sub

Sub GaussRand (Range#, Prob#, Y1#, Y2#)

This subroutine creates two uniformly distributed pseudo-random variables which are modified using the Box-Muller transformation to normally distributed (gaussian) psuedo-random variables with a mean of 0 and a standard deviation of 1. The normal range of the gaussian variables is -3.999 to 3.999 (.0001 to .9999 probabilities). Since Visual Basic can produce a uniform random variable as small as 1.45E-45, (which would yield a gaussian variable value in excess of 100), all output variables in excess of 4 (ABS(Y) > 4) are discarded, and the subroutine

BALLISTI.BAS - 2

repeats itself.

The subroutine also calls the NORMAL subroutine to modify the output variables so that a range and a probability limit around the mean can be set. For example, in the dispersion subroutine, variables are established which simulate 80% of the gun rounds impacting within plus/minus 3.75 mils of the guns nominal trajectory line. range = 3.75 : prob = .8

Input:

range = variable for the range of values.

*prob = variable for the probability of distribution
to be found in the specified range, input as $0 < prob < 1$.*

Output:

Y1 & Y2 = the normally distributed variables outputted by the subroutine.

Dim Pi As Double, X1 As Double, X2 As Double, X As Double

Pi = 3.14159265359

redo:

Randomize

X1 = Rnd

X2 = Rnd

Y1 = Sqr(-2 * Log(X1)) * Cos(2 * Pi * X2) *Box-Muller Transformation*

Y2 = Sqr(-2 * Log(X1)) * Sin(2 * Pi * X2)

If Abs(Y1) Or Abs(Y2) > 4 Then GoTo redo:

Prob = (1 - (1 - Prob) / 2)

Call NORMAL(2, X, Prob)

Range = Range / X

BALLISTI.BAS - 3

Y1 = Y1 * Range

Y2 = Y2 * Range

End Sub

Sub HitCount (XMiss#, YMiss#, Size#, Rangebin%, Hit%)

This subroutine determines if the simulated projectile hit the target based on modified X and Y miss angles, target size, and target range.

Input

XMiss = miss distance in the x axis in mils

YMiss = miss distance in the y axis in mils

Size = target radius in feet

Rangebin = range to target in 100s of feet

Output

Hit: 1 = Yes

2 = No

Dim S As Double, T As Double

S = Sqr(XMiss ^ 2 + YMiss ^ 2)

T = Size * 1000 / Rangebin

If S <= T Then

Hit = 1

Else

Hit = 2

End If

End Sub

BALLISTI.BAS - 4

Sub MissGen (IFLAG%, Rangebin%, XMiss#, YMiss#)

This subroutine generates miss distances based on rangebin and gun system model type.

Input

IFLAG = model type

1 = fixed reticle/gun compensation

2 = fixed reticle/fixed gun

3 = CCIP reticle/fixed gun

Rangebin = range in 500 foot increments (6500 - 500)

Output

XMiss = miss distance in the x axis in mils

YMiss = miss distance in the y axis in mils

Dim rc As Double, Y1 As Double, Y2 As Double

Randomize

Select Case IFLAG

Case 1

Xmiss = (-2.995 * Log(Rangebin) + 27.68) * Int((3 * Rnd) - 1)

Xmiss = XMiss + ((-2.0271 * Log(Rangebin) + 18.709) * Int((3 * Rnd) - 1))

Call GaussRand(4.15182, .8413, Y1, Y2)

Xmiss = XMiss + Y1

Xmiss = XMiss / 2

Ymiss = (-1.4482 * Log(Rangebin) + 13.956) * Int((3 * Rnd) - 1)

Ymiss = YMiss + ((-1.0316 * Log(Rangebin) + 9.8672) * Int((3 * Rnd) - 1))

Call GaussRand(2.488926, .8413, Y1, Y2)

Ymiss = YMiss + Y1

Ymiss = YMiss / 2

Case 2

BALLISTI.BAS - 5

```
Xmiss = (-1.1809 * Log(Rangebin) + 20.462) * Int((3 * Rnd) - 1)
Xmiss = XMiss + ((-1.1415 * Log(Rangebin) + 15.812) * Int((3 * Rnd) - 1))
Call GaussRand(7.625385, .8413, Y1, Y2)
Xmiss = XMiss + Y1
Xmiss = XMiss / 2
```

```
Ymiss = (1.758 * Log(Rangebin) - 2.5947) * Int((3 * Rnd) - 1)
Ymiss = YMiss + ((.9415 * Log(Rangebin) - .4354) * Int((3 * Rnd) - 1))
Call GaussRand(9.141017, .8413, Y1, Y2)
Ymiss = YMiss + Y1
Ymiss = YMiss / 2
```

Case Else

```
Xmiss = (-1.3332 * Log(Rangebin) + 20.991) * Int((3 * Rnd) - 1)
Xmiss = XMiss + ((-1.1056 * Log(Rangebin) + 15.438) * Int((3 * Rnd) - 1))
Call GaussRand(7.226173, .8413, Y1, Y2)
Xmiss = XMiss + Y1
Xmiss = XMiss / 2
```

```
Ymiss = (1.4151 * Log(Rangebin) + .395) * Int((3 * Rnd) - 1)
Ymiss = YMiss + ((.1397 * Log(Rangebin) + 5.798) * Int((3 * Rnd) - 1))
Call GaussRand(7.865352, .8413, Y1, Y2)
Ymiss = YMiss + Y1
Ymiss = YMiss / 2
```

End Select

End Sub

Sub ParametersGen (Roll#, Pitch#, Pitchrate#, Speed#, Turnrate#)

This subroutine generates pseudo-random flight parameters. For the first gun system model in order to examine the relative importance of the various feedback loops to the gun.

The distribution is gaussian or uniform as noted. Range values are derived values from data analysis at one stad deviation for normally distibuted variables.

Output

BALLISTI.BAS - 6

Roll = roll angle in degrees

Pitch = pitch in degrees

Pitchrate = pitch rate in degrees / second

Speed = speed in knots

Turnrate = turnrate in degrees / second

Dim Range As Double, Prob As Double, Y1 As Double, Y2 As Double

Randomize

Generate psuedo-random roll angle, gaussian distribution, max angle 60, mean 0

Redoroll:

Roll = Int((5.110186 * 2 + 1) * Rnd - 5.110186)

Range = 7.541496

Prob = .8413

Call GaussRand(Range, Prob, Y1, Y2)

Roll = Roll + Y1

If Abs(Roll) > 60 Then GoTo Redoroll:

Generate psuedo-random pitch angle, gaussian distribution, max angle 45, mean 0

Redopitch:

Pitch = Int((2.024575 * 2 + 1) * Rnd - 2.024575)

Range = 1.747698

Prob = .8413

Call GaussRand(Range, Prob, Y1, Y2)

Pitch = Pitch + Y1

If Abs(Pitch) > 45 Then GoTo Redopitch:

Generate psuedo-random pitch rate, gaussian distribution, max rate = 60, mean 0

Redopitchrate:

Pitchrate = Int((.741048 * 2 + 1) * Rnd - .741048)

Range = .893324

Prob = .8413

BALLISTI.BAS - 7

Call GaussRand(Range, Prob, Y1, Y2)
Pitchrate = Pitchrate + Y1

If Abs(Pitchrate) > 60 Then GoTo Redopitchrate:

Generate psuedo-random speed, uniform distribution, 90 to 150 knots

Randomize

Speed = Int(150 - 90 + 1) * Rnd + 90

Compute turn rate based on Roll and Speed, max rate 6.84.

Turnrate = -Sgn(Roll) * (((.0006 + .000015 * (Speed - 90))) * Roll ^ 2) + (.2507 ➡
- (.001815 * (Speed - 90))) * Roll)

End Sub

MODSIM.BAS - 1

Option Explicit

Function AdjustRangeForElevation (Pitch#)

Pitch input = elevation angle from aircraft to target in degrees (up or down)

*Y output = ratio of slant range to horizontal range from gun to target in feet
(around 2 at 60 degrees up or down and 1 at 0 degrees)*

Dim Y As Double

$$Y = (1 / \cos(\text{Pitch} * 3.14159265359 / 180)) * (1 - (\tan(2 / 6400 / 360) * \tan(\text{Pitch} * 3.14159265359 / 180)))$$

AdjustRangeForElevation = Y

End Function

Function BulletDrop (TOF#)

TOF input = time of flight (TOF) in seconds

Drop output = bullet drop in feet

Dim Drop As Double

$$\text{Drop} = (-.2542 * (\text{TOF} ^ 4)) + (2.4298 * (\text{TOF} ^ 3)) - (17.14 * (\text{TOF} ^ 2)) + (6.6498 * \text{TOF}) - .0064$$

BulletDrop = Drop

End Function

Sub EndProgram ()

End

End Sub

Function TimeOfFlight (Range#, Speed#)

Input

MODSIM.BAS - 2

Range = range to target in feet
Speed = airspeed in knots

Output

TOF = time of flight (TOF) in seconds

Dim Z1 As Double, Z2 As Double, Z3 As Double, Z4 As Double, TOF As Double

TOF @ 0 KIAS

Z1 = (4E-16 * (Range ^ 4)) + (.0000000000001 * (Range ^ 3)) + (.000000003 * (Range ^ 2)) + (.0003 * Range) - .00007

TOF @ 50 KIAS

Z2 = (4E-16 * (Range ^ 4)) + (.0000000000001 * (Range ^ 3)) + (.000000003 * (Range ^ 2)) + (.0003 * Range) - .00009

TOF @ 100 KIAS

Z3 = (4E-16 * (Range ^ 4)) + (.00000000000007 * (Range ^ 3)) + (.000000003 * (Range ^ 2)) + (.0003 * Range) - .0001

TOF @ 150 KIAS

Z4 = (3E-16 * (Range ^ 4)) + (.00000000000002 * (Range ^ 3)) + (.000000002 * (Range ^ 2)) + (.0003 * Range) - .0209

If Speed > 0 And Speed <= 50 Then

TOF = ((Abs(Z1 - Z2)) * ((50 - Speed) / 50)) + Z2

ElseIf Speed > 50 And Speed <= 100 Then

TOF = ((Abs(Z2 - Z3)) * ((100 - Speed) / 50)) + Z3

ElseIf Speed > 100 And Speed <= 150 Then

TOF = ((Abs(Z3 - Z4)) * ((150 - Speed) / 50)) + Z4

End If

TimeOfFlight = TOF

End Function

Function TurnRateModel (X#, Y#)

Input :

MODSIM.BAS - 3

X! = roll angle in degrees

Y! = airspeed in knots

Output:

Z! ouput = lateral velocity of round in fps due to turn rate

Muzzle to point of rotation distance is 16.7 feet

1 mil = .05625 deg

Dim Z As Double

$Z = -\text{Sgn}(X) * (((.0006 + .000015 * (Y - 90))) * X^2) + (.2057 - (.001815 * (Y - 90))) * X$

$Z = (16.7) * (Z / .05625) / 1000$

TurnRateModel = Z

End Function

NORMAL.BAS - 1: Crescent (1990)

DefInt I-N
DefDbl A-H, O-Z

Static Sub NORMAL (IFLAG, X, PROB)

Normal distribution subroutine

Input

IFLAG = type of computation

1 = given x, compute probability

2 = given probability, compute x

Output (or input)

X = x value

PROB = probability; 0 < PROB < 1

Select Case IFLAG

Case 1

N(x)

X1 = Abs(X)

If (X1 > 7#) Then

PROB = 0#

Else

PROB = X1 * (.0000380036 + X1 * (.0000488906 + X1 * (

.000005383))

PROB = 1# + X1 * (.049867347 + X1 * (.0211410061 + X1 * (

.0032776263 + PROB)))

PROB = .5 * PROB ^ (-16#)

End If

If (X < 0#) Then PROB = 1# - PROB

Case 2

N(p)

If (PROB = .5) Then

X = 0#

Exit Sub

End If

NORMAL.BAS - 2: Crescent (1990)

```
If (PROB > .5) Then
    X1 = 1# - PROB
```

```
Else
    X1 = PROB
```

```
End If
```

```
If (X1 = .05) Then
    X = 1.64485
```

```
ElseIf (X1 = .025) Then
    X = 1.95996
```

```
ElseIf (X1 = .01) Then
    X = 2.32635
```

```
ElseIf (X1 = .005) Then
    X = 2.57583
```

```
Else
    X1 = -Log(4# * X1 * (1# - X1))
    X = (-3.231081277E-09 * X1 + 8.360937017E-08) * X1 - ➡
    .00000104527497
    X = (X * X1 + .000005824238515) * X1 + .000006841218299
    X = ((X * X1 - .0002250947176) * X1 - .000836435359) * X1 + ➡
    .03706987906
    X = X * X1 + 1.570796288
    X = Sqr(X1 * X)
```

```
End If
```

```
If (PROB > .5) Then X = -X
```

```
End Select
```

```
End Sub
```

APPENDIX F

INSTRUCTIONS FOR RUNNING THE GUN TRACKING SIMULATION PROGRAM

The enclosed floppy disk drive contains the executable program "track.exe". It was designed to run on a 75mhz Pentium based computer using a joystick operating with a Microsoft Windows® driver. The development operating system was Windows for Workgroups® 3.11, but it should run under Windows® 3.1 and Windows 95® as well.

The joystick is not necessary to run the program, as all form selections are made with a mouse or by using the computer "tab" and "enter" keys. Control of the flight model is possible using the keyboard cursor keys, however control is changed from a rate command controller to a position command controller, and flight control is notchy. One technique which seems to work well is to make the cursor key inputs a series of taps on the appropriate key rather than just holding the cursor key down.

The data run mode should not be selected unless a drive with recordable media installed is designated as the "D:" drive. If there is no "D:" drive or the "D:" drive is not equipped with recordable media, the program will crash in the data run mode. The training modes do not require a recordable "D:" drive and the program will run without problem in those modes without a drive "D:". The program was designed to operate on a computer which was configured with a RAM drive as the "D:" drive. If a slower recordable drive is installed as the "D:" drive, the program will freeze during data writes.

If the data run mode is utilized, a name must be typed in the name block on the registration window. This name is used as the data file name, and without one, the program will default to the training modes.

There is one documented bug in the program. If the options button is chosen on the “start” form, and a specific flight model is selected, the program will go back to the “start” form as designed, but then it will sometimes go to a second “configure” window. After this window is cleared and the program again displays the “start” form, the program will run as designed.

To run the program either

- Select FILE on the pull down menu bar of the Windows® program manager or file manager. Select RUN from the pull down menu. Type in “A:\track” in the run entry line and hit return. If the floppy drive is not the “A:\” drive then substitute the appropriate letter for the floppy drive.

or

- Open the Windows® file manager, display the contents of the floppy drive that the enclosed disk has been loaded into and double-click on the file “track.exe”.

VITA

Travis Milam Allen was born in Kountze, Texas on August 7th, 1958. The son of a career Marine Corps pilot, he attended primary and secondary schools at Marine Corps bases across the United States. He received a Presidential appointment to the United States Naval Academy at Annapolis, Maryland and graduated with a Bachelor of Science degree in Mathematics and a commission as a Second Lieutenant in the US Marine Corps in May, 1980. He received primary and advanced flight training at NAS Whiting Field in Pensacola, Florida and was designated a Naval Aviator in January, 1982. He served tours in several Marine attack helicopter squadrons and participated in aerial combat operations in Saudi Arabia and Kuwait during the Gulf War. His combat decorations include the Distinguished Flying Cross, Single Mission Air Medal, Strike/Flight Air Medals, and the Navy/Marine Corps Commendation Medal. He was posted to the US Naval Test Pilot school at NAWC-AD Patuxent River, Maryland in 1992 and designated as an Engineering Test Pilot in December, 1993.

He is currently serving as the Senior AH-1W "SuperCobra" Engineering Test Pilot at the Naval Rotary Wing Aircraft Test Squadron at Patuxent River.