

FOR REFERENCE
NOT TO BE TAKEN FROM THIS ROOM

ADAPTIVE GUIDANCE OF SHORT-RANGE
TACTICAL MISSILES

by

ÖZCAN AĞABAY

B.S. in M.E. and Physics, Boğaziçi University, 1983

Bogazici University Library



14

39001100313405

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science
in
Mechanical Engineering

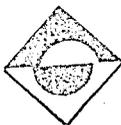
Boğaziçi University

1986

ACKNOWLEDGEMENTS

I would like to thank Doç.Dr. Ahmet Kuzucu for his useful suggestions which were of great help throughout this work.

Also I am grateful to Mr. Alper Oysal for the painstaking typing of the manuscript.



ABSTRACT

Missile control methods are briefly outlined, the notation and conventions used in guided missile literature are introduced. A realistic simulation model is formed considering the differential equations governing motion of the missile in plane, the measurement and control system and the intercept geometry. Computer simulations are repeated for various attack geometries and target escape scenarios using both proportional navigation guidance and suboptimal adaptive control. Good missile performance in terms of miss distance is obtained for proportional navigation guidance. Suboptimal adaptive control can be more useful if other performance criteria are important, such as impact angle at target.

ÖZET

Füze kontrol yöntemleri kısaca açıklandı, güdümlü füze literatüründe kullanılan yazılım ve kurallar tanımlandı. Füzenin düzlemsel hareketinin, ölçüm ve kontrol sisteminin ve buluşma geometrisinin diferansiyel denklemleri yardımıyla gerçekçi bir benzeşim modeli kuruldu. Oransal yönlendirme güdümü ve altoptimal uyurlayıcı kontrol kullanılarak değişik saldırı geometrileri ve hedef kaçış senaryoları için bilgisayar benzeşimleri yinelendi. Oransal yönlendirme güdümü ile hedefi yakalama açısından iyi sonuçlar elde edildi. Altoptimal uyarlayıcı kontrolün diğer verim ölçütlerinin, hedefe çarpma açısı gibi, önemli olduğu durumlarda daha kullanışlı olabileceği sonucuna varıldı.

TABLE OF CONTENTS

	<u>PAGE</u>
ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
ÖZET	v
LIST OF FIGURES	viii
LIST OF TABLES	x
LIST OF SYMBOLS	xi
I. INTRODUCTION	1
II. MISSILE CONTROL METHODS	4
2.1 Introduction to Missile Control Methods	4
2.2 Aerodynamic Lateral Control	6
2.3 Thrust Vector Control	11
2.4 Notation and Conventions	13
2.5 Euler's Equations of Motion for a Rigid Body	15
III. IN PLANE MISSILE FLIGHT DYNAMICS AND ATTACK GEOMETRY	18
3.1 3 DOF Differential Equations of Motion	18
3.2 Measurement and Control System	24
3.3 Launch Conditions	26
3.4 State Variables for the Intercept Problem	27
IV. APPLIED GUIDANCE LAWS	29
4.1 Proportional Navigation Guidance	29
4.2 The General State Regulator Problem	33
4.3 Apparent Linearization	38
4.4 Suboptimal Adaptive Control	41

	<u>PAGE</u>
V. COMPUTER SIMULATION	46
5.1 Description of Computer Programs Used	46
5.2 Scaling	50
5.3 Scenarios	53
5.4 Time-to-go Estimation	54
VI. SIMULATION RESULTS AND DISCUSSION	57
6.1 General Results	57
6.2 Simulation with PNG Law	63
6.3 Suboptimal Guidance Results	70
VII. CONCLUSION	76
BIBLIOGRAPHY	78
APPENDIX - SIMULATION PROGRAM LISTING	80

LIST OF FIGURES

	<u>PAGE</u>	
Fig. 2.1.1	Missile position and error signals	5
Fig. 2.1.2	Control surfaces looking from rear of missile	5
Fig. 2.2.1	Unstable (missile)	7
Fig. 2.2.2	Neutrally stable (missile)	7
Fig. 2.2.3	Stable (missile)	7
Fig. 2.2.4	Rear control surfaces supersonic	9
Fig. 2.2.5	Rear control surfaces subsonic	9
Fig. 2.2.6	Canard controls	9
Fig. 2.2.7	Moving wings	9
Fig. 2.4.1	Force, moment, etc. conventions	10
Fig. 3.1.1	Intercept geometry	18
Fig. 3.2.1	Measurement and control system	25
Fig. 3.3.1	Launch conditions	26
Fig. 4.1.1	Proportional navigation trajectories	30
Fig. 5.3.1	Launch scenarios	53
Fig. 5.4.1	Homing time estimation	55
Fig. 6.1.1	Missile speed vs. time	57
Fig. 6.1.2	Trajectories for $\sigma_0 = 0^\circ$	59
Fig. 6.1.3	Trajectories for $\sigma_0 = 45^\circ$	60
Fig. 6.1.4	Trajectories for $\sigma_0 = 90^\circ$	61
Fig. 6.1.5	Trajectories for $\sigma_0 = 135^\circ$	62
Fig. 6.1.6	Trajectories for $\sigma_0 = 180^\circ$	63
Fig. 6.2.1	Missile and target positions at $t=0$, $t=2.2$ secs	65
Fig. 6.2.2	Miss distance vs. launch angle	69

		<u>PAGE</u>
Fig. 6.3.1	Control variation with time	71
Fig. 6.3.2	Lateral velocity variation with time	71
Fig. 6.3.3	Control variation with time	73
Fig. 6.3.4	Lateral velocity variation with time	73

LIST OF TABLES

		<u>PAGE</u>
Table 2.4.1	Notation	10
Table 6.2.1	Lateral velocity variation with time	65
Table 6.2.2	Best launch angles for different scenarios	67

LIST OF SYMBOLS

A, B, C	Moment of inertia about each axis
\underline{A}	State matrix
a_b	Base area
a_e	Nozzle exit area (rocket motor)
a_T	Target lateral acceleration
\underline{B}	Control matrix
C_m	Coefficient in Eq.(3.1.3)
C_x	Coefficient in Eq.(3.1.2)
C_α	Coefficient in Eq.(3.1.1)
D, E, F	Products of inertia
\underline{D}_k	Matrix in Eq.(4.4.12)
e_x, e_y	Intercept errors
\underline{G}	Gain Matrix
H	Hamiltonian
h	Altitude in km
I_y	Moment of inertia of the missile
J	Cost functional
\underline{K}	Riccati matrix
K_q	Rate gyro coefficient
kk	Doubling number
L, M, N	Moments acting on missile
L_R	Reference length
M	Mach number
m	Mass of missile
P_N	Navigation constant

P_{α}	Angle of attack stabilization coefficient
p, q, r	Angular rates
\underline{Q}	State penalization matrix
\hat{q}	Coefficient in Eq.(3.1.1)
R	Range
\underline{R}	Control penalization matrix
\underline{S}	Final state penalization matrix
S_R	Reference area
s	Step size used in evaluating the matrix exponential
T	Thrust
T_f	Time-to-go
T_p	Time-to-pass
u, v, w	Components of missile velocity
\underline{u}	Control vector
v_m	Missile speed
v_T	Target speed
X, Y, Z	Forces acting on missile
\underline{X}	Scaled state vector
X_{CGB}	Center of gravity location at burnout
x, y, z	Roll, pitch and yaw axis respectively
\underline{x}	State vector
x_m, y_m	Missile trajectory components
α	Incidence angle
β	Angle between LOS and missile at launch
θ	Flight path angle
δ_z	Flipper deflection
δ_{zd}	Rate gyro deflection
δ_{zel}	Flipper actuation servo input

Ω	Missile heading angle
Ω_T	Target heading angle
$\underline{\lambda}$	Adjoint vector
$\underline{\phi}$	Fundamental matrix
$\underline{\phi}_{ij}$	Submatrices of the fundamental matrix
σ	Line-of-sight angle
τ_g	Rate gyro time constant
τ_s	Flipper actuation servo time constant

I. INTRODUCTION

In Garnell and East (1) the definition of a guided missile is given as: "A guided missile is one which is usually fired in a direction approximately towards the target and subsequently receives steering commands from the guidance system to improve its accuracy." By accuracy we mean the missile should be oriented towards the target in such a way that at some final time the position of the missile is close to that of the target; i.e., the miss distance is small.

Guidance laws for short range tactical missiles have become a well researched topic over the past 40 years with publication of analytical treatment and implementation of missile guidance going back to 1940's. Thus, much of the guidance development available in the literature predates that which is known as modern control theory. These early concepts, now commonly referred to as classical guidance, have been used from that time to the present to command missiles during their homing phases of flight to target impact. The performance of classical guidance techniques were satisfactory against the targets they were designed for. However, the performance of missiles may be seriously degraded in engagement against targets with predicted characteristics of the 1990's and beyond. The guidance laws currently in wide use may not be adequate in defeating such treats. Thus, it is predicted that fundamental advances in the application of control systems theory is required to enhance the guidance effectiveness of future missile systems.

It is observed that guidance laws typically fit within one of the following categories: line-of-sight (LOS), pursuit and proportional navigation guidance (PNG), these three being classical schemes; optimal guidance and other guidance laws dominated by differential game methods. A short description of each of the basic guidance laws follows.

The LOS guidance scheme is one in which the missile is guided on an LOS course in an attempt to remain on the line joining the target and the point of control.

One of the most straightforward means to assure impact is to keep the missile, which must have velocity superiority, pointed at the target. This is the principle of pursuit guidance, which has two basic variations: attitude pursuit, in which the missile's longitudinal axis is directed at the target; and velocity pursuit, in which the missile's velocity vector is kept pointed at the target.

Proportional navigation guidance (PNG) probably had its origins among the ancient seafarers who realised that a collision was ensured if two constant velocity vessels maintained constant relative bearing while closing in range. Thus, unlike pursuit schemes, which seek to null the LOS, PNG seeks to null the LOS rate, while closing on the target.

Since the mid-1960's the missile guidance literature has been permeated by techniques based upon optimal control. Most formulations consider terminal miss distance and running control effort in the cost functional. A running cost on the state is not appropriate in this framework, since the purpose is to minimize at final time and not continuously during flight. In an optimal control law the problem is to find such a control that the miss distance is small while also the control is kept as small as possible. Certainly some weighing is necessary to adjust the relative importance of terminal miss distance and running control effort.

The performance of any realistic optimal control law in a missile application is dependent on the estimation of final time, or in other words time-to-go. Typically, an estimate of the range between target and missile and the rate of change of this range are obtained from radar or other ranging devices; the time-to-go estimate is then calculated. This process works well as long as the range and range-rate information are accurate.

A significant portion of the literature on guidance laws does not readily fall within the coverage of the four previous schemes. Some of this work concentrates on specialized applications of control theory, particularly differential games, while others represent very simple straightforward implementation of ad-hoc controllers.

An extensive survey of both classical and optimal guidance covering work done in this field till 1979 can be found in a paper by Pastrick et.al. (2), upon which this introduction is based to a great extent.

II. MISSILE CONTROL METHODS

2.1. Introduction to Missile Control Methods

Before going into mathematical detail concerning the motion of a missile in space as a result of guidance commands, some definitions and discussion are desirable.

It is convenient to start with a definition of the task of a missile control system. It is one of the tasks of the guidance system to detect whether the missile is flying too low or too high, or too much to the left or to the right. It measures these deviations and sends signals to the control system to reduce these errors to zero. The task of the control system is therefore to manoeuvre the missile quickly and efficiently as a result of these signals. Suppose the guidance "sees" the missile at m relative to its boresight and that we interpret this to mean that the missile is too far to the right and too low. In a Cartesian coordinate system the guidance angular error detector produces two signals, a left-right signal and an up-down signal which are transmitted to two separate servos, say rudder servos and elevator servos. Fig. 2.1.1 shows that this same information can be expressed in polar coordinates; i.e., R and ϕ . If the same information is expressed in another way then the control system must be mechanised differently. The usual method is to regard the ϕ signal as a command to roll through an angle of ϕ measured from the vertical and then to manoeuvre outwards by means of the missile's elevators. The method of control compatible with polar coordinates is called twist and steer.

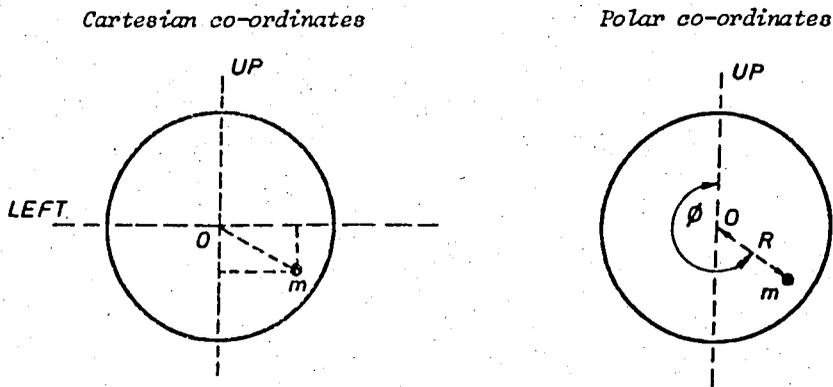


Figure 2.1-1 Missile position and error signals

Guided missiles usually have one or two axes of symmetry. If a missile has four control surfaces as shown in Fig. 2.1.2 one regards surfaces 1 and 3 as elevators and 2 and 4 as rudders even if the missile should roll subsequently. If 1 and 3 are mechanically linked together such that a servo must impart the same rotation to both of

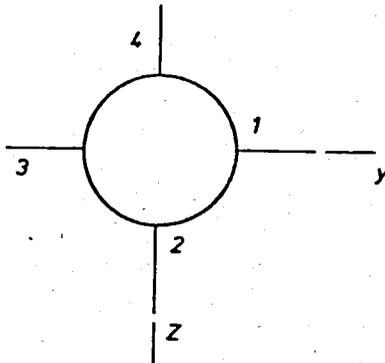


Figure 2.1-2 Control surfaces looking from rear of missiles

them these surfaces are elevators pure and simple. The same argument applies to the rudders. Suppose now surfaces 1 and 3 each have their own servo, then it is possible for them to act as ailerons. If looking in the direction y one surface is rotated ξ^0 clockwise and the other surface ξ^0 anti-clockwise then a pure couple is imparted to the missile

about the fore and aft axis and this will tend to make the missile roll. Such control surfaces are called ailerons. We can double the power of the ailerons by doing the same thing to control surfaces 2 and 4. If now the aerodynamics are linear, i.e., the normal forces are proportional to incidence, then the principle of superposition applies. Commands for elevator, rudder and ailerons movements can be added electrically resulting in unequal movements to opposite control surfaces. In this way we have the means to control roll motion as well as the up-down (i.e., pitch) motion and left-right (i.e., yaw) motion.

2.2 Aerodynamic Lateral Control

With the Cartesian control system the pitch control system is made identical to yaw control system so we need to discuss one channel only; in this respect the nomenclature differs from that used in aircraft. With missiles lateral movement usually means up-down or left-right. With polar control one rolls and elevates. The following remarks apply to the elevation channel in twist and steer missiles also.

The majority of tactical missiles have fixed main lifting surfaces (often called wings) with their center of pressure somewhere near the missile center of gravity and rear control surfaces. With subsonic missiles it may be more efficient to use the controls as flaps immediately behind the wings as the flap controls the circulation over the whole surface. With supersonic flow the control surface cannot affect the flow ahead of itself and therefore is placed as far to the rear as possible in order to exert the maximum moment on the missile. Rear control surfaces often make a convenient arrangement of components. Usually it is desirable to have the propulsion system placed centrally in the missile so that the center of gravity shifts due to propellant

usage are minimized. It is convenient and sometimes essential to have the warhead and fuze at the front together with any associated electronics including the guidance receiver. This leaves the control system to occupy the rear and with the propulsion blast pipe passing through its center. If there are four servos it is not difficult to design a neat servo package round this pipe.

When considering lateral forces and moments on missiles it is convenient first of all to consider the combined normal forces due to incidence on the body, wings and control surfaces as acting through a point of the body called the center of pressure (c.p.) and to regard the control surfaces as permanently locked in the central position. If the c.p. is ahead of the center of gravity (c.g.) then the missile is said to be statically unstable. If it coincides with the c.g. then it is said to be neutrally stable and if it is behind the c.g. it is said to be statically stable. This of course is the reason why feathers are placed at the rear end of an arrow to move the c.p. aft. These three possible conditions are shown in Fig. 2.2.1 to 2.2.3. The missiles are shown with a small incidence, i.e., the body is not pointing in the same direction as the velocity vector U_m . In the unstable condition any perturbation of the body away from the direction of the velocity vector

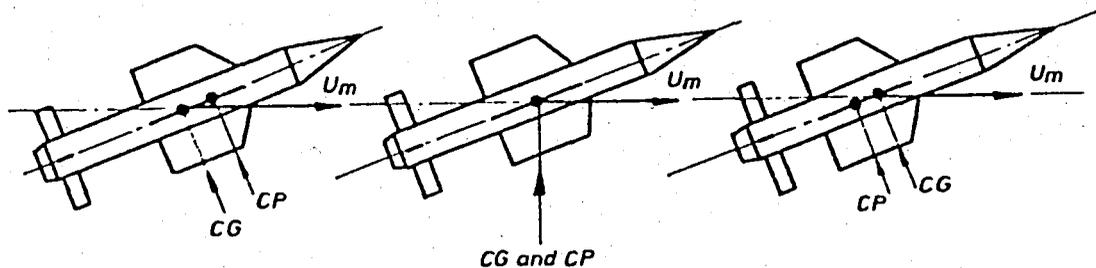


Fig.2.2.1 Unstable

Fig.2.2.2 Neutrally stable

Fig.2.2.3 Stable

results in a moment about the c.g. which tends to increase this perturbation. Conversely in the stable case any perturbation of the body direction results in a moment which tends to oppose or decrease this perturbation. The distance of the c.p. to the c.g. is called the static margin. Since the lateral force and hence the lateral manoeuvre by aerodynamic means is obtained by exerting a moment on the body such that some incidence occurs it follows that if the static margin is excessive, the missile is unnecessarily stable and control moment will be relatively ineffective in producing a sizeable manoeuvre. There has to be a compromise between stability and manoeuvrability. Now consider a missile whose forward speed is constant, with a steady body and wing incidence of β and a control surface movement from the central position of ζ . Only motion in the horizontal plane is considered and the missile is assumed to be not rolling; the effects of gravity are zero in this plane. Fig. 2.2.4 shows the normal force N due to the body, wings and rear control surfaces assumed to be in the central position; this N force acts through the c.p. But there will be an additional force N_c due to the control surfaces being deflected by an amount ζ . Neglecting the small damping moment due to the fact that the missile is executing a steady turn, this picture can represent dynamic equilibrium if the rudder movement $N_c \ell_c$ is numerically equal to Nx^* where x^* is the static margin. If $\ell_c/x^* = 10$ say then $N = 10N_c$, and the total lateral force = $9N_c$. This force is in the opposite sense to N_c . Since x^* is typically 5 per cent or less of the body length it is easily seen that a small absolute change in the static margin can affect the manoeuvrability of the missile. Thus the standard method of obtaining a large lateral force on a missile is to have a large moment arm by placing the control surfaces as far from the c.g. as possible. If the c.p. of the body and the wings alone is at the c.g. then ξ^0 of control surface movement will produce the same amount of body incidence. If the c.p. as just defined

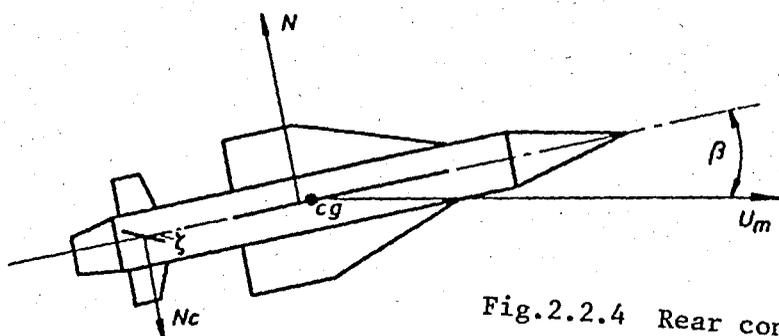


Fig.2.2.4 Rear control surfaces supersonic

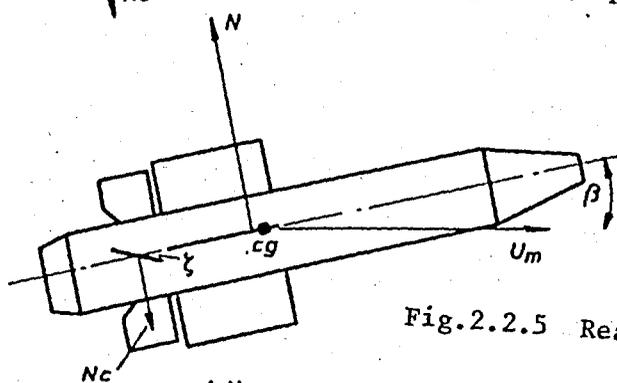


Fig.2.2.5 Rear control surfaces subsonic

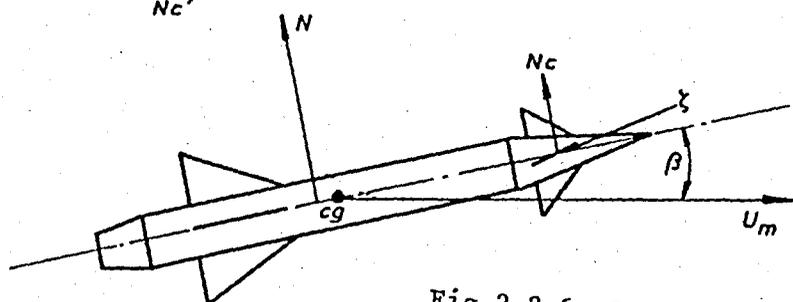


Fig.2.2.6 Canard controls

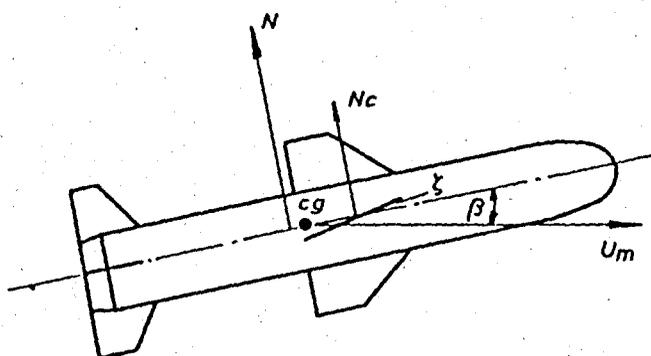


Fig.2.2.7 Moving wings

is in front of the c.g. then ζ^0 of rudder movement will produce more than ζ^0 of body incidence. If this c.p. is behind the c.g. then less than ζ^0 of body incidence will result. If a missile has no autopilot (i.e., no instrument feedback) a considerable static margin has to be allowed to ensure stability in flight, say 5 per cent or more of the overall length. With instrument feedback zero or even negative static margins can be used, thus assisting manoeuvrability. It should be noted that the overall c.p. can never be regarded as in a fixed position. The c.p. of the body in particular will vary with incidence and Mach number.

Since the main object of siting a control surface is to place it as far from the c.g. as possible, a position as far forward as is practicable appears a logical choice. Forward control surfaces are often called "canards" named after ducks who apparently steer themselves by moving their heads. Fig. 2.2.6 shows another possible case of dynamic equilibrium. In this case it is seen that the lateral force due to the missile as a whole now adds to the force due to the deflection of the control surface and therefore if $l_c/x^* = 10$ as before, then the total normal force is $11N_c$ compared with $9N_c$ with rear controls. Also, the final sense of the total normal force is in the same sense as the control force. Canards therefore are slightly more effective in the use of lateral control forces. Canards will not render the missile unstable since as can be noted from Fig 2.2.6 the main lifting surfaces are rather further aft to make the overall c.p. aft of the c.g., which is the stability criterion

To use servos to move the main lifting surfaces and employ small fixed rear stabilizing surfaces is unusual. There could be the rare occasion when the servos are more conveniently placed near the center of the missile. However, the main reason for adopting this configuration would be, for a given lateral acceleration, to minimize the body

incidence. For instance if the propulsion system is a ram jet the air intake is likely to choke if the body incidence is large, say 15° or more. However, there are some distinct penalties involved in the use of moving wings. Clearly the servos will be appreciably larger to cope with the increased inertia of the load and the larger aerodynamic hinge moments. Also, moving wings are an inefficient way of producing a large normal force due to the small moment arm available. Owing to the fact that the whole bending moment at the wing root has to be taken by the shaft, the wing will have to be designed much thicker around the mid chord. This not only increases the structure weight but at supersonic speeds it will increase the drag; the pressure drag varying with the thickness-to-chord ratio squared. It is desirable to make the center section of the missile square in cross section to eliminate a large wing-body gap when the wing is deflected; such a gap considerably reduces the generated normal force. And finally since the moment arm is small, the position of the c.g. is critical as a small shift will make an appreciable change in the control moment arm. Nevertheless, if the maximum g requirements are low and the speed is subsonic, such as for an anti-ship missile, the overall weight penalty may not be excessive if small moving wings are used.

2.3 Thrust Vector Control

A completely different method of steering a missile is to alter the direction of the efflux from the propulsion motor and such a method is known as thrust vector control (TVC). This method of control is clearly not primarily dependent on the dynamic pressure of the atmosphere, but on the other hand it is inoperative after motor burn-out. In many situations there are advantages in having a boost-coast velocity profile. TVC is therefore likely to have a limited application. The following situations make TVC essential or desirable.

(a) It is essential to use TVC in the vertical launch phase of all intercontinental ballistic missiles as these missiles, whose total weight is well over 90 per cent fuel, have to be launched extremely gradually to avoid dynamic loading. Aerodynamic controls would be completely ineffective for some time and the missile would topple over due to a small inevitable thrust misalignment unless an attitude sensor and TVC were used.

(b) If a missile is separated some distance from its controller such as in the anti-tank system Swingfire and rapid gathering is required to achieve a short minimum range then it must be possible to manoeuvre the missile almost immediately after launch.

(c) In a short range air-to-air missile, one may be trying to hit a fast crossing target with no aim-off and with a flight time of a few seconds. The exceptional manoeuvrability one can obtain with TVC would give the system a better coverage.

(d) It can be argued that some systems would be cheaper and simpler if one launched vertically and then turned over rapidly, thus eliminating an expensive and heavy launcher.

(e) Vertical launch followed by a rapid turnover is an attractive concept for missiles carried and launched from a vehicle; 360° arc of fire is obtainable, and storage and reloading is almost certainly facilitated.

(f) Submarine launched missiles surfacing in different sea conditions may well need very early course correction.

2.4 Notation and Conventions

The reference axis system standardized in the guided missile literature is centered on the c.g. and fixed in the body as follows:

x axis, called the roll axis forward along the axis of symmetry if one exists, but in any case in the plane of symmetry.

y axis, called the pitch axis, outwards and to the right if viewing the missile from behind.

z axis, called the yaw axis, downwards in the plane of symmetry to form a right handed orthogonal system with the other two.

Table 2.4.1 defines the forces and moments acting on the missile, the linear and angular velocities, and the moments of inertia ; these quantities are shown in Fig. 2.4.1. The moments of inertia about 0 are defined as:

$$A = \Sigma \delta m (y^2 + z^2) \quad (2.4.1)$$

$$B = \Sigma \delta m (z^2 + x^2) \quad (2.4.2)$$

$$C = \Sigma \delta m (x^2 + y^2) \quad (2.4.3)$$

The products of inertia are defined as,

$$D = \Sigma \delta m y z \quad (2.4.4)$$

$$E = \Sigma \delta m x z \quad (2.4.5)$$

$$F = \Sigma \delta m x y \quad (2.4.6)$$

	Roll axis x	Pitch axis y	Yaw axis z
Angular rates	p	q	r
Component of missile velocity along each axis	u	v	w
Component of force acting on missile along each axis	X	Y	Z
Moments acting on missile about each axis	L	M	N
Moments of inertia about each axis	A	B	C
Products of inertia	D	E	F

Table 2.4-1 Notation

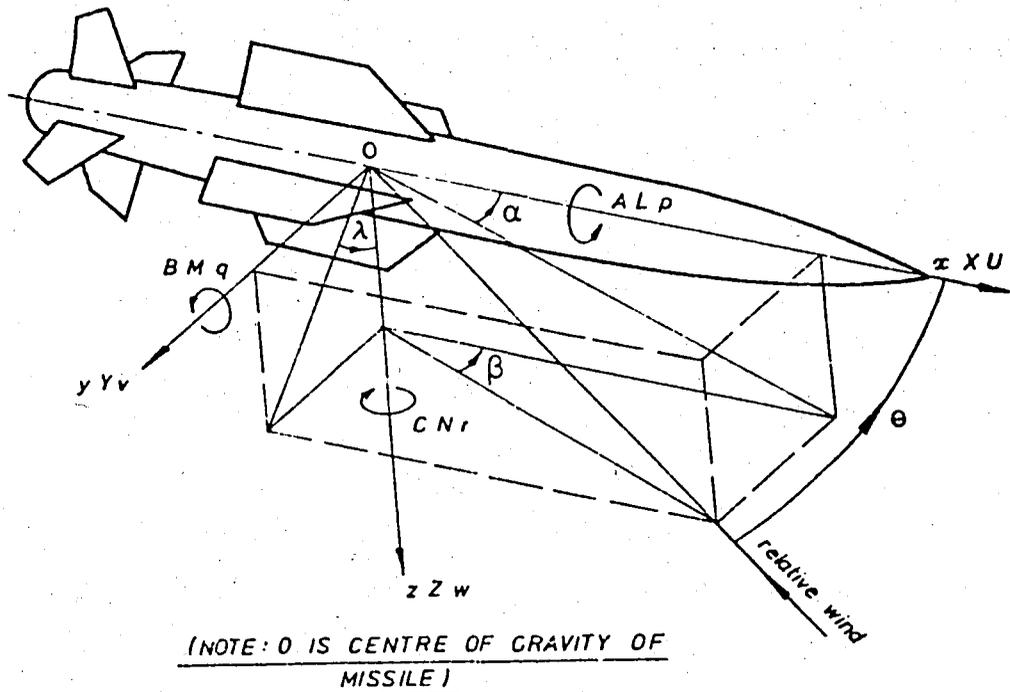


Figure 2.4-1 Force, moment etc. conventions

The yaw plane is the Oy plane and the pitch plane is the Oz plane.

The following angles are defined,

- β : incidence in the yaw plane
 α : incidence in the pitch plane
 λ : incidence plane angle
 θ : total incidence such that

$$\tan\alpha = \tan\theta\cos\lambda$$

$$\text{and } \tan\beta = \tan\theta\sin\lambda$$

The reason why U , the missile velocity along the x axis is denoted by a capital letter is to emphasize that it is a large positive quantity changing at most a few per cent per second. The angular rates and components of velocity along the pitch and yaw planes however, tend to be much smaller quantities which can be positive or negative and can have much larger rates of change.

2.5 Euler's' Equations of Motion for a Rigid Body

There are six equations of motion for a body with six degrees of freedom, three force equations and three moment equations. If the missile mass is m they are:

$$m(\dot{u} + qw - rv) = x \quad (2.5.1)$$

$$m(\dot{v} + rU - pw) = y \quad (2.5.2)$$

$$m(\dot{w} - qU + pv) = z \quad (2.5.3)$$

$$A\dot{p} - (B - C)qr + D(r^2 - q^2) - E(pq + \dot{r}) + F(rq - \dot{q}) = L \quad (2.5.4)$$

$$B\dot{q} - (C - A)rp + E(p^2 - r^2) - F(qr + \dot{p}) + D(pq - \dot{r}) = M \quad (2.5.5)$$

$$C\dot{r} - (A - B)pq + F(q^2 - p^2) - D(rp + \dot{q}) + E(qr - \dot{p}) = N \quad (2.5.6)$$

The first equation does not really concern us; we are interested in the acceleration perpendicular to the velocity vector as this will result in a change in the velocity direction. In any case in order to determine the change in the forward speed we must know the magnitude of the propulsive and drag forces. Now consider Eq.(2.5.2), the term $-mpw$ is saying that there is a force in the y direction due to incidence in pitch ($\alpha = w/U$) and roll motion. In other words the pitching motion of the missile is coupled to the yawing motion on account of roll rates. The term mpv in Eq.(2.5.3) is also saying that yawing motion induces forces in the pitch plane if rolling motion is present. This is most undesirable since we require these two "channels" to be completely uncoupled. Ideally rudder movements should produce forces and moments in the yaw plane and result in yawing motion only; elevators should result in a manoeuvre in the pitch plane. Cross-coupling between the planes must contribute to system inaccuracy. To reduce these undesirable effects the designer tries to keep roll rates as small as possible, and in a simplified analysis one usually neglects the term pw and pv if roll rates are expected to be small and incidence (v and w are proportional to incidence) is not large.

Now consider the moment equations. Ideally these should read

$$A\dot{p} = L \quad ; \quad B\dot{q} = M \quad ; \quad C\dot{r} = N$$

i.e., moments about a given axis produce angular accelerations about that axis. All other terms in these equations are cross-coupling terms and are undesirable from the point of view of system accuracy. We note that three out of four of the cross coupling terms in each equation disappear if there are two axes of symmetry, and two will be zero and one will be small if there is one axis of symmetry and the missile is reasonable symmetric about another axis. With two planes of symmetry

and a small roll rate these equations reduce to

$$m(\dot{U} + qw - rv) = X \quad (2.5.7)$$

$$m(\dot{v} + rU) = Y \quad (2.5.8)$$

$$m(\dot{w} - qU) = Z \quad (2.5.9)$$

$$A\dot{p} - (B - C)qr = L \quad (2.5.10)$$

$$B\dot{q} = M \quad (2.5.11)$$

$$C\dot{r} = N \quad (2.5.12)$$

The justification for neglecting the terms pq , pr , pv , pw is that the terms q , r , v , and w are not large and if p is small then their products can be neglected. Eq.(2.5.10) shows that there is zero coupling between the pitch and roll and yaw and roll motions if there are two axes of symmetry ($B = C$) and unless the missile is very unsymmetrical the cross-coupling should be weak.

The intercept geometry is shown in Fig. 3.1.1 where the various angles, velocities etc. used in the equations of motion are shown. One point needs to be noted concerning Fig. 3.3.1. The axes labeled as x' , y' are only for reference and the plane they form is in reality the xz (pitch) plane as defined by the conventional reference system mentioned in Sec. 2.4. The axis labeled as x is the roll axis of the missile. Some of the angles shown in Fig. 3.1.1 are named as:

α : angle of attack (incidence)

Ω : missile heading angle

Ω_T : target heading angle

σ : LOS angle

Three degree of freedom (3 DOF) differential equations of motion in the vertical plane in the body coordinate system are below.

$$\dot{w} = \frac{1}{m} \hat{q} C_z(\alpha, \delta_z; M) + qw \quad (3.1.1)$$

$$\dot{u} = \frac{1}{m} [T + \hat{q} C_x(\alpha; M, h)] - qw \quad (3.1.2)$$

$$\dot{q} = \hat{q} \frac{L_R}{I_y} [C_m(\alpha; \delta_z; M) + \frac{L_R}{v_m} C_{mq}(h) q] \quad (3.1.3)$$

$$\dot{\theta} = q \quad (3.1.4)$$

where

$$\alpha = \text{atan}(w/u) \quad (3.1.5)$$

$$\theta = \Omega + \alpha \quad (3.1.6)$$

$$v_m = \sqrt{u^2 + w^2} \quad (3.1.7)$$

$$\hat{q} = \frac{\rho}{2} V_m^2 S_R \quad (3.1.8)$$

$$M = V_m/c \quad (3.1.9)$$

$$C_x = -C_c \quad (3.1.10)$$

$$C_z = -C_N \quad (3.1.11)$$

$$C_c = C_{C_0}(\alpha; M) + C_f(M)(h - 6.096)/3.048 + C_B(M)(1 - a_e/a_b) \quad (3.1.12)$$

$$C_N = C_{N_\alpha}(\alpha; M)\alpha + C_{N\delta_z}(M)\delta_z \quad (3.1.13)$$

$$C_m = \left[C_{N\alpha}(\alpha; M)(X_{CG}(t) - X_{CP}(\alpha, M)/L_R) \right] \alpha + \left[C_{m\delta_z}(M) + C_{N\delta_z}(M)(X_{CG}(t) - X_{CGB}/L_R) \right] \delta_z \quad (3.1.14)$$

The constants and variables (some of which are tabulated and some numerically evaluated) used in the above equations are briefly explained below. First the constants:

- S_R : reference area
- L_R : reference length
- X_{CGB} : center of gravity location at burnout
- a_e : nozzle exit area (rocket-motor)
- a_b : base area

Tabulated variables:

- $\rho(h)$: air density ($\text{kp/m}^2\text{s}^2$) vs. altitude (km)
- $p(h)$: static pressure (kp/m^2) vs. altitude (km)
- $c(h)$: speed of sound (m/s) vs. altitude (km)
- $C_B(M)$: base drag coefficient vs. Mach number

- $C_{mq}(M)$: pitch damping moment coefficient (per rad) vs. Mach number
 $C_{N\delta_z}(M)$: trim normal force effectiveness (per deg) vs. Mach number
 $C_f(M)$: friction drag coefficient vs. Mach number
 $C_{Co}(\alpha;M)$: axial force coefficient vs. Mach number and incidence (deg)
 $C_{N\alpha}(\alpha;M)$: normal force coefficient vs. Mach number and incidence (deg)
 $C_{m\delta_z}(M)$: trim pitching moment effectiveness (per deg) vs. Mach number
 $X_{Cp}(\alpha;M)$: center of pressure location (m), from nose vs. Mach number and incidence
 $T(t)$: rocket motor thrust (kP) vs. time (s)

Numerically evaluated variables:

m : mass of missile

I_y : moment of inertia of missile

X_G : center of gravity of missile

Since during boost mass is continuously ejected the mass, moment of inertia, and center of gravity of the missile will change in this period. Thrust duration is assumed to be 2.2 seconds and above mentioned variables are assumed to vary linearly with time from their initial values to their values at burnout.

The force and moment terms in the equations of motion are related to missile aerodynamics. First we consider the normal force term in Eq.(3.1.1)

$$N = \hat{q} C_N(\alpha; \delta_z, M) \quad (3.1.15)$$

with

$$C_N = C_{N\alpha}(\alpha; M)\alpha + C_{N\delta_z}(M)\delta_z \quad (3.1.16)$$

where $C_{N\alpha}$ and $C_{N\delta_z}$ are tabulated coefficients. Next we consider the axial force term in Eq.(3.1.2) T is the rocket thrust and

$$C = \hat{q} C_c \quad (3.1.17)$$

with

$$C_c = C_{Co}(\alpha;M) + \Delta C_F(M,h) + C_B(1 - \eta) \quad (3.1.18)$$

where

C_{Co} : axial force coefficient

ΔC_F : friction drag coefficient, function of altitude $h(\text{km})$, due to air viscosity change with altitude

$$\Delta C_F = C_f(h - 6.096)/3.048 \quad (3.1.19)$$

C_{BASE} : base drag coefficient for flight during motor burning, the base drag is reduced by the ratio of the nozzle exit area divided by the base area

$$C_{BASE} = C_B(1 - a_e/a_b) \quad (3.1.20)$$

Here also C_{Co} , C_f and C_B are tabulated coefficients. Lastly we consider the pitching moment term in Eq.(3.1.3).

$$C_m = C_{N\alpha}(\alpha;M)\alpha [X_{CG}(t) - X_{Cp}(\alpha;M)] / L_R + \quad (3.1.21)$$

$$[C_{m\delta_z}(M) + C_{N\delta_z}(M)(X_{CG}(t) - X_{CGB}) / L_R] \delta_z$$

The first line represents the static pitching moment which is obtained by multiplying the normal force coefficient with the static margin; i.e., by the dimensionless distance separating center of pressure and center of gravity, both locations being measured from the nose, positive backwards. The second term is the flipper deflection moment, which is corrected by the change of center of gravity location during boost. While the expression of the last line accounts for the natural damping moment

in pitch (q :rad's). All pitching moment coefficients are given for zero bank angle. Here, as well, X_{Cp} , $C_{N\delta_z}$, $C_{m\delta_z}$ and C_{mq} are tabulated coefficients.

Inspecting Fig. 3.1.1 we see that the target dynamics can be represented by

$$\dot{\Omega}_T = a_T/v_T \quad (3.1.22)$$

$$\dot{x}_T = v_T \cos\Omega_T \quad (3.1.23)$$

$$\dot{y}_T = v_T \sin\Omega_T \quad (3.1.24)$$

where a_T : lateral acceleration of the target.

Similarly the intercept error can be approximated by

$$e_x = x_T - x_m \quad (3.1.25)$$

$$e_y = y_T - y_m \quad (3.1.26)$$

From Eqs. (3.1.25) and (3.1.26) we obtain

$$\dot{e}_x = v_T \cos\Omega_T - v_m \cos(\theta - \alpha) \quad (3.1.27)$$

$$\dot{e}_y = v_T \sin\Omega_T - v_m \sin(\theta - \alpha) \quad (3.1.28)$$

The trajectory equations for the missile may be helpful.

They are

$$\dot{x}_m = v_m \cos\Omega \quad (3.1.29)$$

$$\dot{y}_m = v_m \sin\Omega \quad (3.1.30)$$

We define the relative range R as

$$R = \sqrt{e_x^2 + e_y^2} \quad (3.1.31)$$

3.2 Measurement and Control System

Let us consider any aspect of the motion of a missile through space. Forces and moments will produce accelerations and hence velocities and displacements with respect to the earth; or, as is often stated with respect to inertial space. If we wish to make a closed loop system of the means of controlling the motion of a missile then we must use instruments to measure accelerations, velocities and displacements in space. Accelerometers, rate gyros and position gyros are used for this purpose. It is usual to call a system comprising missile fin or thrust vector servos, an airframe, instruments and any electronics and networks necessary to close the loop an autopilot; but this nomenclature is not universal.

In the system under consideration a flipper actuation servo and a rate gyro, to measure the angular rate about the pitch axis, are used. The flipper actuation servo is a first order element with Laplace transform

$$\delta_z(s) = \frac{1}{\tau_s + 1} \delta_{z_s} \quad (3.2.1)$$

where

$$\delta_{z_s} = \delta_{z_{el}} - \delta_{z_d} \quad (3.2.2)$$

The rate gyro has a Laplace transform

$$\delta_{z_d} = \frac{K_g \tau_g s}{1 + \tau_g s} q(s) \quad (3.2.3)$$

From Eqs. (3.2.1) and (3.2.3) we can write

$$\dot{\delta}_z = -\frac{1}{\tau_s} \delta_z + \frac{1}{\tau_s} \delta_{z_s} \quad (3.2.4)$$

$$\dot{\delta}_{z_d} = \frac{1}{\tau_g} \delta_{z_d} + K_q \dot{q} \quad (3.2.5)$$

where

- τ_s : servo time constant
- K_q : rate gyro coefficient
- τ_g : rate gyro time constant

Combining Eqs.(3.2.2) and (3.2.4) we obtain

$$\dot{\delta}_z = -\frac{1}{\tau_s} \delta_z - \frac{1}{\tau_s} \delta_{z_d} + \frac{1}{\tau_s} \delta_{z_{el}} \quad (3.2.6)$$

Fig. 3.2.1 shows a block diagram of the measurement and control system.

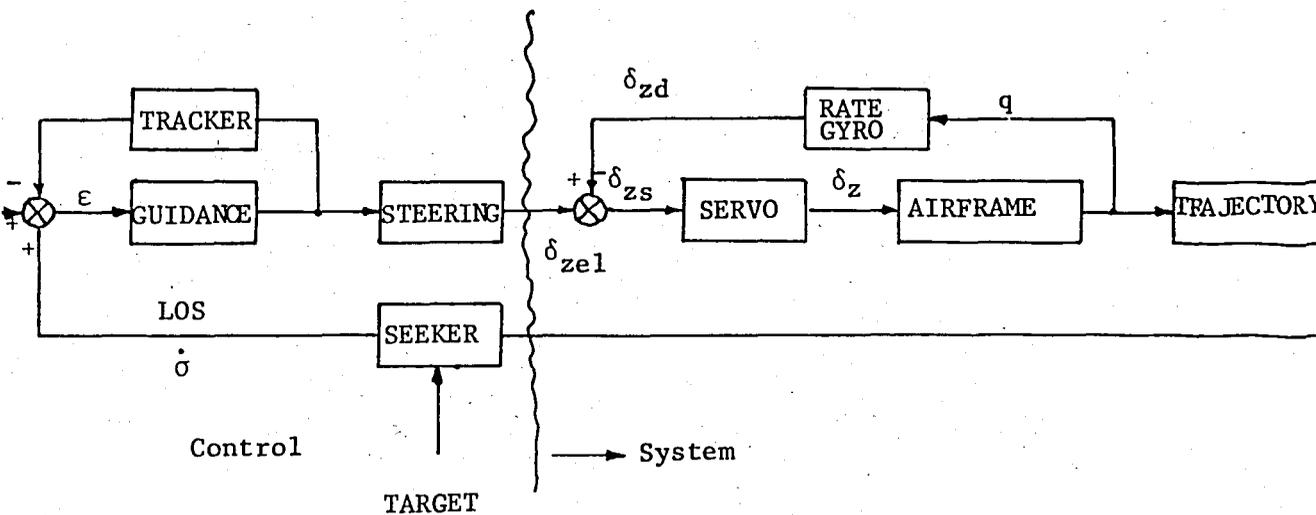


Figure 3.2-1 Measurement and control system

3.3 Launch Conditions

Since the manoeuvrability of any missile is limited it is important that the missile is launched so that it is approximately directed towards the target. Bearing this fact in mind two launch conditions seem plausible. One is the lead pursuit approach, the other one lead collision approach. These two approaches are shown schematically in Fig. 3.3.1.

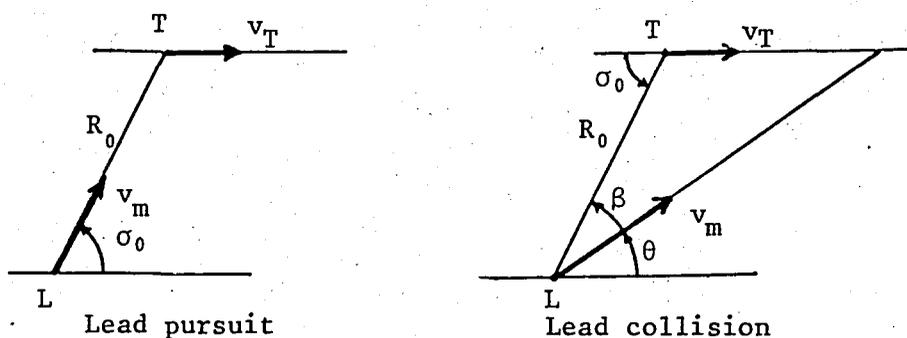


Figure 3.1-1 Launch conditions

In essence lead pursuit is nothing but simply firing the missile towards the point the target is at launch time. From Fig. 3.3.1 we see that $\theta_0 = \sigma_0$ for lead pursuit. This approach has the serious drawback that the velocity of the target is not taken into consideration. The target will be moving away from the point the missile is directed towards and this may endanger the success of the missile; especially if the control is not activated immediately after launch as is the case for the system under consideration. When the control system is activated the target might be quite away from the point it was initially and this may give rise to large control signals. Excessive control may cause overshoots and erode the stability margins of the control system.

Lead collision approach overcomes most of drawbacks caused by lead pursuit. In this approach the missile is fired towards the point we hope the missile and target will collide. This hope would have been fulfilled if both missile and target velocities had remained constant

and neither the missile nor the target manoeuvred. In Fig. 3.3.1 β is the angle the missile makes with the LOS. Using sine law we can write

$$\frac{v_m \Delta t}{\sin(\pi - \sigma_0)} = \frac{v_T \Delta t}{\sin \beta} \quad (3.3.1)$$

and rearranging we obtain

$$\sin \beta = \frac{v_T}{v_m} \sin \sigma_0 \quad (3.3.2)$$

Since the missile should have velocity superiority over the target we see that the missile is aimed towards a point ahead of the target at launch. Note that if the target is stationary lead collision is equivalent to lead pursuit.

3.4 State Variables for the Intercept Problem

In this section we collect the differential equations for the intercept problem together and rewrite them in the form of state equations below.

$$\dot{x}_1 = \frac{1}{m} \hat{q}(M) C_z(M, x_6) + x_2 x_3 \quad (3.4.1)$$

$$\dot{x}_2 = \frac{1}{m} [T + \hat{q}(M) C_x(M, h)] - x_1 x_3 \quad (3.4.2)$$

$$\dot{x}_3 = \frac{L_R}{I_y} \hat{q}(M) \left[C_m(M, x_6) + \frac{L_R}{v_m} C_{mq}(M) x_3 \right] \quad (3.4.3)$$

$$\dot{x}_4 = x_3 \quad (3.4.4)$$

$$\dot{x}_5 = -\frac{1}{\tau_g} x_5 + K_q \dot{x}_3 \quad (3.4.5)$$

$$\dot{x}_6 = -\frac{1}{\tau_s} (x_5 + x_6) + \frac{1}{\tau_s} u' \quad (3.4.6)$$

$$\dot{x}_7 = a_T/v_T \quad (3.4.7)$$

$$\dot{x}_8 = v_T \cos x_7 - \frac{v}{m} \cos(x_4 - \alpha(x_1, x_2)) \quad (3.4.8)$$

$$\dot{x}_9 = v_T \sin x_7 - \frac{v}{m} \sin(x_4 - \alpha(x_1, x_2)) \quad (3.4.9)$$

where $x_1 = w$, $x_2 = u$, $x_3 = q$, $x_4 = \theta$, $x_5 = \delta_{z_d}$, $x_6 = \delta_z$,

$$x_7 = \Omega_T , x_8 = e_x , x_9 = e_y$$

and

$$u' = \delta_{z_{e1}}$$

The variables and constants appearing in the above equations are defined earlier.

Inspecting Eqs. (3.4.1) through (3.4.9) we note that the only control variable is u' (i.e., $\delta_{z_{e1}}$) which constitutes the input to the flipper actuation servo. The control is applied to vary the deflection of the flipper which in turn affects the other state variables through the nonlinear state equations above.

IV. APPLIED GUIDANCE LAWS

4.1 Proportional Navigation Guidance

PNG as mentioned in the introduction seeks to null the line-of-sight (LOS) rate, while closing on the target. PNG is a guidance law in which, ideally, the angular rate of the missile flight path is directly proportional to the angular LOS rate of change, i.e.

$$\dot{\theta} = P_N \dot{\sigma} \quad (4.1.1)$$

where $\dot{\theta}$ represents the flight path angular rate relative to a fixed reference, $\dot{\sigma}$ is the LOS rate relative to a fixed reference, P_N is the so-called navigation constant.

In order to obtain some feel for the problem it is worth considering a special case of an interception when missile and target speeds are constant and $v_m/v_T = 2$ say. The target is assumed to fly straight and we aim directly at the target. The guidance system is engineered such that a rate of change of trajectory ($\dot{\theta}$) which is P_N times the rate of change of line-of-sight ($\dot{\sigma}$) is produced. Two cases are considered (a) $P_N = 1$ and (b) $P_N = 4$ and lags on the system are neglected. In Fig. 4.1.1 M_0, M_1, M_n and T_0, T_1, T_n are positions of the missile and the target at launch and at successive intervals of time after launch. Dotted lines represent the LOS. If the navigation constant is unity then it is not difficult to see that as the trajectory changes at the same rate as the line-of-sight and one aims at the target in the first place then the line drawn tangential to the missile flight path

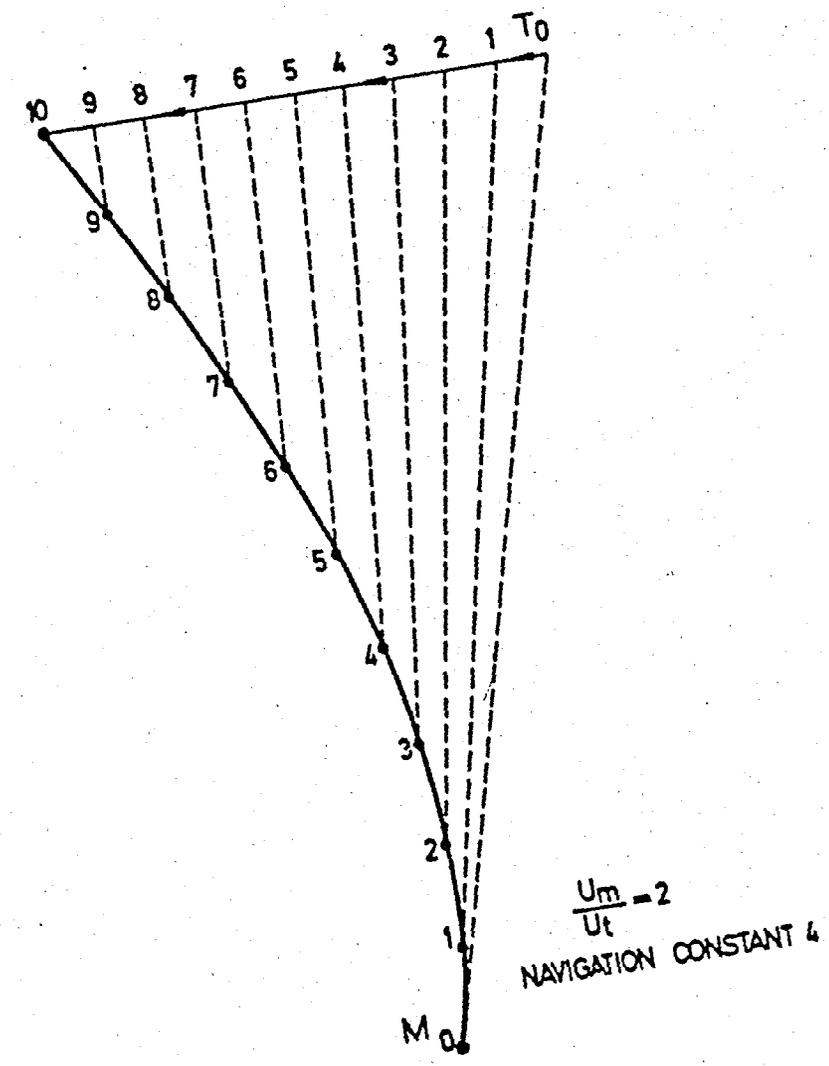
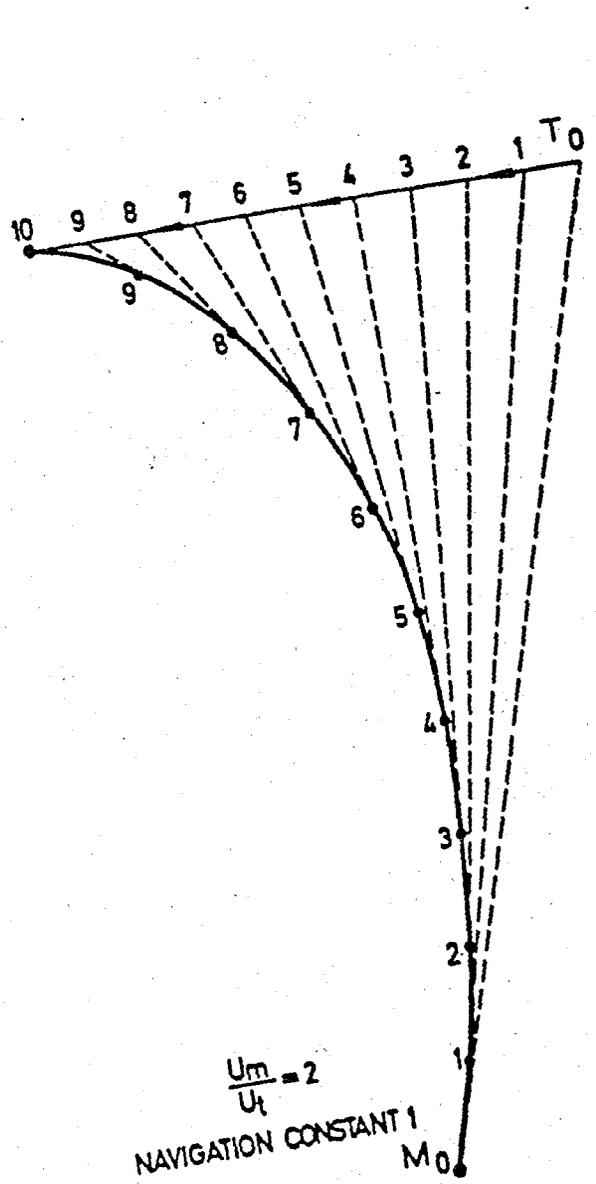


Figure 4.1.1 Proportional navigation trajectories

must start and remain coincident with the line-of-sight. Since a tangent to the flight path indicates the instantaneous direction of the velocity such a trajectory is called the "pursuit course" as it is the sort of trajectory a dog might conceivably follow in chasing a rabbit. He always heads for the target and never attempts to aim ahead. If a navigation constant of say four is used, initially the line-of-sight rate must be the same as in the first case, but the missile steering commands are four times as great; as a result the missile veers off much more to the left. Examination of the diagram shows that the line-of-sight rate reduces as the engagement proceeds. It is important to realise that such a guidance law automatically establishes a lead angle.

If the line-of-sight does not rotate in space (i.e., $\dot{\sigma} = 0$) then no steering commands are necessary as one is on a collision course. If the line-of-sight does rotate (i.e., $\dot{\sigma}$ exists) then a change of trajectory direction is required and it must be in such a sense as to reduce $\dot{\sigma}$. Clearly the plane of the manoeuvre must be in the plane of $\dot{\sigma}$. Imagine a roll stabilized missile. The homing head measures the vertical and horizontal component of line-of-sight rate and passes these signals suitably scaled to the elevator and rudder servos respectively.

It should be noted that Eq.(4.1.1) shows the ideal case, which in general is not attainable. Thus, we cannot control our system so that Eq.(4.1.1) is exactly satisfied. The reason being that we cannot change θ as we wish. The only control is applied to the flipper actuation servo which in turn affects θ and other state variables through the coupled state equations. Also, the time constants for the flipper actuation servo and the rate gyro inserts time lags into the control system. We use Eq.(4.1.1) in evaluating the control $\delta_{z_{el}}$ such that

$$\delta_{z_{el}} = P_N \dot{\sigma} \quad (4.1.2)$$

where $\dot{\sigma}$ is the line-of-sight rate as before.

Examining Fig. 3.1.1 we note that

$$\tan \sigma = \frac{e_y}{e_x} \quad (4.1.3)$$

and differentiating Eq.(4.1.3) with respect to time we obtain

$$\dot{\sigma} = \frac{\dot{e}_y e_x - \dot{e}_x e_y}{R^2} \quad (4.1.4)$$

where $R = \sqrt{e_x^2 + e_y^2}$ as defined by Eq.(3.1.31).

We have also experimented with an augmented control law of the form

$$\delta_{z_{el}} = P_N \dot{\sigma} + P_\alpha \dot{\alpha} \quad (4.1.5)$$

where $\dot{\alpha}$ is the rate of angle of attack and P_α is angle of attack stabilization coefficient ($P_\alpha < 0$). α is a measure of incidence and it is desirable to keep α as small as possible since the aerodynamic effectiveness of the missile decreases with increasing angle of attack (e.g., the drag force increases as α increases). By adding a term into the control law we can obtain a smaller α throughout without a significant decrease in missile performance. Differentiating Eq.(3.1.5) respect to time we obtain

$$\dot{\alpha} = \frac{\dot{w}u - \dot{u}w}{v_m^2} \quad (4.1.6)$$

where $v_m^2 = u^2 + w^2$ as before.

4.2 The General State-Regulator Problem

In this section we consider the so-called state regulator problem which forms the basis of the optimal control law applied in this present analysis.¹ Basically, the solution of the state-regulator problem leads to an optimal feedback system with the property that the components of the state vector $\underline{x}(t)$ are kept near zero without excessive expenditure of control energy. In solving this problem we are going to make use of the celebrated minimum principle of Pontryagin (3).

Let us consider the linear time-varying system,

$$\dot{\underline{x}}(t) = \underline{A}(t) \underline{x}(t) + \underline{B}(t) \underline{u}(t) \quad (4.2.1)$$

and the cost functional J is given by

$$J = \frac{1}{2} \underline{x}^T \underline{S} \underline{x} + \frac{1}{2} \int_{t_0}^{T_f} \underline{x}^T \underline{Q} \underline{x} + \underline{u}^T \underline{R} \underline{u} \, dt \quad (4.2.2)$$

where the below assumptions are satisfied.

The terminal time T_f is specified.

S is a constant $n \times n$ positive semidefinite matrix.

$Q(t)$ is an $n \times n$ positive semidefinite matrix.

$R(t)$ is an $r \times r$ positive definite matrix.

The physical interpretation of J is this: we want to keep the state near zero without excessive control-energy expenditure.

We shall show in this section that the optimal control is a linear function of the state, i.e., is of the form

$$\underline{u}(t) = \underline{G}(t) \underline{x}(t) \quad (4.2.3)$$

where $\underline{G}(t)$ is an $r \times n$ matrix-valued function called the "gain matrix".

¹ We have relied almost exclusively on the results of Falb and Athans (4) in this section.

Let us assume that an optimal control exists for any initial state. We can use the minimum principle to obtain the necessary conditions for the optimal control and so derive the extremal controls. The Hamiltonian for the system (4.2.1) and cost J of Eq.(4.2.2) is

$$H = \frac{1}{2} \underline{x}^T \underline{Q} \underline{x} + \frac{1}{2} \underline{u}^T \underline{R} \underline{u} + \underline{\lambda}^T (\underline{A} \underline{x} + \underline{B} \underline{u}) \quad (4.2.4)$$

The adjoint (or costate) vector $\underline{\lambda}(t)$ is the solution of the vector differential equation

$$\dot{\underline{\lambda}}(t) = - \frac{\partial H}{\partial \underline{x}(t)} \quad (4.2.5)$$

which reduces to

$$\dot{\underline{\lambda}}(t) = -\underline{Q}(t) \underline{x}(t) - \underline{A}^T(t) \underline{\lambda}(t) \quad (4.2.6)$$

Along the optimal trajectory, we must have

$$\frac{\partial H}{\partial \underline{u}(t)} = \underline{0} \quad (4.2.7)$$

which implies that

$$\frac{\partial H}{\partial \underline{u}(t)} = \underline{R}(t) \underline{u}(t) + \underline{B}^T(t) \underline{\lambda}(t) = \underline{0} \quad (4.2.8)$$

From Eq. (4.2.8) we deduce that

$$\underline{u}(t) = -\underline{R}^{-1}(t) \underline{B}^T(t) \underline{\lambda}(t) \quad (4.2.9)$$

The assumption that $R(t)$ is positive definite² for all $t \in [t_0, T_f]$ guarantees the existence of $\underline{R}^{-1}(t)$ for all $t \in [t_0, T_f]$.

² A necessary condition that \underline{M} be positive definite is $\det \underline{M} > 0$ and so \underline{M} is nonsingular.

We know that the optimal control must minimize the Hamiltonian. The necessary condition $\partial H / \partial \underline{u}(t) = \underline{0}$ yields only an extremum of H with respect to $\underline{u}(t)$. In order for the extremum of H to be a minimum with respect to $\underline{u}(t)$, the rxr matrix $\partial^2 H / \partial \underline{u}^2(t)$ must be positive definite. But, from Eq. (4.2.8), we find that

$$\frac{\partial^2 H}{\partial \underline{u}^2(t)} = \underline{R}(t) \quad (4.2.10)$$

and, hence, since $\underline{R}(t)$ was assumed to be positive definite, it follows that the control $\underline{u}(t)$ given by Eq.(4.2.9) does indeed minimize H .

The next step is to obtain the reduced canonical equations, to do that, we substitute Eq.(4.2.9) into Eq.(4.2.1) to obtain the relation

$$\dot{\underline{x}}(t) = \underline{A}(t) \underline{x}(t) - \underline{B}(t) \underline{R}^{-1}(t) \underline{B}^T(t) \underline{\lambda}(t) \quad (4.2.11)$$

Eqs. (4.2.11) and (4.2.6) are the reduced canonical equations. Define the matrix $\underline{F}(t)$ by setting

$$\underline{F}(t) = \underline{B}(t) \underline{R}^{-1}(t) \underline{B}^T(t) \quad (4.2.12)$$

Note that $\underline{F}(t)$ is a symmetric nxn matrix. Using the matrix $\underline{F}(t)$, we can combine the canonical Eqs. (4.2.11) and (4.2.6) in the form

$$\begin{bmatrix} \dot{\underline{x}}(t) \\ \dot{\underline{\lambda}}(t) \end{bmatrix} = \begin{bmatrix} \underline{A}(t) & -\underline{F}(t) \\ -\underline{Q}(t) & -\underline{A}^T(t) \end{bmatrix} \begin{bmatrix} \underline{x}(t) \\ \underline{\lambda}(t) \end{bmatrix} \quad (4.2.13)$$

Eq. (4.2.13) is a system of $2n$ linear time-varying homogenous differential equations. We know that we can obtain a unique solution of this system of differential equations provided that we know a total of $2n$ boundary

conditions. A total of n boundary conditions is provided by the transversality conditions, which require (since $\underline{x}(T_f)$ is not specified) that, at the terminal time T_f , the costate $\underline{\lambda}(T_f)$ must satisfy the relation

$$\underline{\lambda}(T_f) = \frac{\partial}{\partial \underline{x}(T_f)} \left[\frac{1}{2} \underline{x}^T(T_f) \underline{S} \underline{x}(T_f) \right] \quad (4.2.14)$$

Thus, we deduce that

$$\underline{\lambda}(T_f) = \underline{S} \underline{x}(T_f) \quad (4.2.15)$$

Let $\phi(t; t_0)$ be the $2n \times 2n$ fundamental matrix for the system (4.2.13). If we let $\underline{\lambda}(t_0)$ be the (unknown) initial costate, then the solution of Eq.(4.2.13) is of the form

$$\begin{bmatrix} \underline{x}(t) \\ \underline{\lambda}(t) \end{bmatrix} = \underline{\phi}(t; t_0) \begin{bmatrix} \underline{x}(t_0) \\ \underline{\lambda}(t_0) \end{bmatrix} \quad (4.2.16)$$

where $\underline{\phi}(t; t_0)$ is the $2n \times 2n$ fundamental matrix for the system (4.2.13).

Therefore, at $t = T_f$, we must have the relation

$$\begin{bmatrix} \underline{x}(T_f) \\ \underline{\lambda}(T_f) \end{bmatrix} = \underline{\phi}(T_f; t) \begin{bmatrix} \underline{x}(t) \\ \underline{\lambda}(t) \end{bmatrix} \quad (4.2.17)$$

Next, we partition the $2n \times 2n$ matrix $\underline{\phi}(T_f; t)$ into four $n \times n$ submatrices as follows:

$$\underline{\phi}(T_f; t) = \left[\begin{array}{c|c} \underline{\phi}_{11}(T_f, t) & \underline{\phi}_{12}(T_f; t) \\ \hline \underline{\phi}_{21}(T_f; t) & \underline{\phi}_{22}(T_f; t) \end{array} \right] \quad (4.2.18)$$

Then Eq.(4.2.17) can be written in the form [using $\underline{\lambda}(T_f) = \underline{S} \underline{x}(T_f)$]

$$\underline{x}(T_f) = \underline{\phi}_{11}(T_f; t) \underline{x}(t) + \underline{\phi}_{12}(T_f; t) \underline{\lambda}(t) \quad (4.2.19)$$

$$\underline{\lambda}(T_f) = \underline{\phi}_{21}(T_f; t) \underline{x}(t) + \underline{\phi}_{22}(T_f; t) \underline{\lambda}(t) \quad (4.2.20)$$

From Eqs. (4.2.19) and (4.2.20) we find, after some algebraic manipulations, that

$$\underline{\lambda}(t) = [\underline{\phi}_{22}(T_f; t) - \underline{S} \underline{\phi}_{12}(T_f; t)]^{-1} \cdot \quad (4.2.21)$$

$$[\underline{S} \underline{\phi}_{11}(T_f; t) - \underline{\phi}_{21}(T_f; t)] \underline{x}(t)$$

provided the indicated inverse exists. Eq.(4.2.21) suggests that the costate $\underline{\lambda}(t)$ and the state $\underline{x}(t)$ are related by an equation of the form

$$\underline{\lambda}(t) = \underline{K}(t) \underline{x}(t) \quad (4.2.22)$$

for all $t \in [t_0, T_f]$. The matrix $\underline{K}(t)$ is an $n \times n$ time varying matrix which depends upon the terminal time T_f and the matrix \underline{S} but does not depend upon the initial state. In fact,

$$\underline{K}(t) = [\underline{\phi}_{22}(T_f; t) - \underline{S} \underline{\phi}_{12}(T_f; t)]^{-1} \cdot \quad (4.2.23)$$

$$[\underline{S} \underline{\phi}_{11}(T_f; t) - \underline{\phi}_{21}(T_f; t)]$$

It can be shown that (see Kalman(5)) the required inverse matrix exists for all $t \in [t_0, T_f]$ so that the relation provided by Eq.(4.2.22) is valid.

Let us now comment on evaluating the matrix $\underline{K}(t)$. If the matrices $\underline{A}(t)$, $\underline{F}(t)$, and $\underline{Q}(t)$ are time-varying, then it is impossible, in general, to obtain an analytical expression for the $2n \times 2n$ fundamental matrix $\underline{\phi}(T_f; t)$. In this case, one must evaluate $\underline{K}(t)$ by using (say) a digital

computer. If however, the matrices $\underline{A}(t)$, $\underline{F}(t)$, and $\underline{Q}(t)$ are time-invariant then the matrix $\underline{\phi}(T_f; t)$ can be evaluated analytically by using, for example, Laplace transforms; nonetheless, even in that case, the evaluation of the inverse matrix in Eq.(4.2.21) is an extremely laborious task, especially if the order of the system is high, i.e., if n is a large number.

Thus, we have devised mathematical tools for finding an optimal control for our intercept problem. In Section 4.4 we will show how the fundamental matrix $\underline{\phi}(T_f; t)$ can be evaluated using Pade approximations.

4.3 Apparent Linearization

In the previous section we have shown how to obtain an optimal control for linear time-varying systems. However, considering Eqs.(3.4.1) through (3.4.9), which we propose to use as our system equations for the intercept problem, we see that they are highly nonlinear. Thus, if we want to use the results of the previous section we have to, somehow, linearize the state equations.

Here we are going to use a method, proposed by Pearson (6), in which the original system equations are used although the selection of the matrices is not unique. This method of linearization is called "apparent linearization" and is preferred over linearization by Taylor series expansion by Weber and Lapidus (7). An example briefly illustrates the method. Consider the equation

$$\dot{x}_i(t) = g_i(\underline{x}, \underline{u}, t) + h_i(\underline{x}, \underline{u}, t) + p_i(\underline{x}, \underline{u}, t) \quad (i = 1, 2) \quad (4.3.1)$$

with at least one term nonzero. We now rewrite (4.3.1) as

$$\dot{x}_i(t) = \begin{bmatrix} g_i \\ x_1 \end{bmatrix} x_1 + \begin{bmatrix} h_i \\ x_2 \end{bmatrix} x_2 + \begin{bmatrix} p_i \\ u \end{bmatrix} u \quad (4.3.2)$$

where

$$\lim_{x_1 \rightarrow 0} \left| \frac{g_i}{x_1} \right| < \infty, \quad \lim_{x_2 \rightarrow 0} \left| \frac{h_i}{x_2} \right| < \infty \quad \text{and} \quad (4.3.3)$$

$$\lim_{u \rightarrow 0} \left| \frac{p_i}{u} \right| < \infty \quad (i = 1, 2)$$

The coefficient matrices of the linearized system

$$\dot{\underline{x}} = \underline{A}(\underline{x}, \underline{u}, t) \underline{x} + \underline{B}(\underline{x}, \underline{u}, t) \quad (4.3.4)$$

would be

$$\underline{A} = \begin{bmatrix} g_1/x_1 & h_1/x_2 \\ g_2/x_1 & h_2/x_2 \end{bmatrix} \quad \text{and} \quad \underline{B} = \begin{bmatrix} p_1/u \\ p_2/u \end{bmatrix} \quad (4.3.5)$$

By providing state and control trajectories the dependence of \underline{A} and \underline{B} on \underline{x} and \underline{u} can be eliminated to yield $\underline{A}(t)$ and $\underline{B}(t)$.

Following the procedure outlined above, we write equations (3.4.1) through (3.4.6) and equations (3.4.8) and (3.4.9) in the form

$$\dot{x}_1 = \left(\frac{\hat{q}C_z}{mx_2} \right) x_2 + (x_2) x_3 \quad (4.3.6)$$

$$\dot{x}_2 = \left(\frac{T \pm \hat{q}C_x}{mx_2} \right) x_2 + (-x_3) x_1 \quad (4.3.7)$$

$$\dot{x}_3 = \left(\frac{L_R \hat{q}C_m}{I_y x_2} \right) x_2 + \left(\frac{L_R^2 C_{mq}}{I_y v m} \right) x_3 \quad (4.3.8)$$

$$\dot{x}_4 = x_3 \quad (4.3.9)$$

$$\dot{x}_5 = \left(-\frac{1}{\tau_g}\right) x_5 + \left(\frac{K}{q} \frac{L}{R} \frac{\hat{q}C}{m}\right) x_2 + \left(\frac{K}{q} \frac{L}{R} \frac{C}{mq}\right) x_3 \quad (4.3.10)$$

$$\dot{x}_6 = \left(-\frac{1}{\tau_s}\right) x_5 + \left(-\frac{1}{\tau_s}\right) x_6 + \left(\frac{1}{\tau_s}\right) u \quad (4.3.11)$$

$$\dot{x}_8 = \left(\frac{v_T \cos x_7 - v_m \cos(x_4 - \alpha)}{2x_8}\right) x_8 + \quad (4.3.12)$$

$$\left(\frac{v_T \cos x_7 - v_m \cos(x_4 - \alpha)}{2x_9}\right) x_9$$

$$\dot{x}_9 = \left(\frac{v_T \sin x_7 - v_m \sin(x_4 - \alpha)}{2x_8}\right) x_8 +$$

$$\left(\frac{v_T \sin x_7 - v_m \sin(x_4 - \alpha)}{2x_9}\right) x_9 \quad (4.3.13)$$

By the way there are a few points that should be noted concerning Eqs. (4.3.6)-(4.3.13). First, the equation for \dot{x}_7 (i.e., $\dot{\Omega}_T$) is no more included among the system equations. This is justified by the fact that both a_T and v_T as mere perturbations and cannot be affected by whatever control we apply to our system. a_T and v_T are determined by the target escape policy. Therefore, we compute $\Omega_T(x_7)$ as a parameter and use it in Eqs.(4.3.12) and (4.3.13). With exclusion of the equation for x_7 the order of our system is reduced to eight even though there seem to be nine state variables.

Secondly, Eqs.(4.3.12) and (4.3.13) seem to contradict the necessary conditions represented by Eq.(4.3.3) which state that the coefficients of the matrices \underline{A} and \underline{B} should have a finite limit as the state variables go to zero. In Eqs.(4.3.12) and (4.3.13) if we let x_8 and x_9 go to zero the corresponding coefficients would become infinite. Since we are trying to minimize the miss distance, which is dependent on x_8 (e_x) and x_9 (e_y),

ideally x_8 and x_9 should equal zero at final time. However, this difficulty can be overcome easily. As is going to be considered in the next section, we are going to use a suboptimal adaptive control for the intercept problem. So, we simply stop adaptation when x_8 or x_9 become small enough. The terms with x_2 in the denominator will cause no trouble because x_2 (which represents the velocity component of the missile along the roll axis) is always nonzero.

4.4 Suboptimal Adaptive Control

In Section 4.2 we have devised mathematical tools for obtaining control laws for linear time-varying systems. However, the actual implementation and computation of such a control law is very difficult unless some simplifying assumptions are made. As mentioned in Section 4.2 the evaluation of the Riccati matrix $\underline{K}(t)$ is quite cumbersome. Luckily using a method suggested by Davison and Maki (8) the matrix $\underline{K}(t)$ can be computed quite fastly on a computer. A generalised version of Davison and Maki's method is used in this thesis to determine $\underline{K}(t)$.

In Section 4.2 we have considered a linear time-varying system with a cost functional J where the state and control penalization matrices $\underline{Q}(t)$ and $\underline{R}(t)$ are functions of time also. There we had found that the control can be easily computed if the fundamental matrix $\underline{\phi}(t;t_0)$ is evaluated. The evaluation of the fundamental matrix $\underline{\phi}(t;t_0)$ is in general, a hard task. However, it is a well-known fact that the fundamental matrix for a linear time-invariant system of the form

$$\dot{\underline{x}}(t) = \underline{A} \underline{x}(t) \quad (4.4.1)$$

where \underline{A} is an $n \times n$ constant matrix is

$$\underline{\phi}(t) = e^{\underline{A}t} \quad (4.4.2)$$

where e^{-At} may be viewed as the infinite series sum

$$e^{-At} = \sum_{k=0}^{\infty} \frac{A^k}{k!} t^k \quad (4.4.3)$$

Bearing the above mentioned points in mind, in the remainder of this section we will develop a suboptimal adaptive control scheme following the method suggested by Kuzucu and Roth (9).

We rewrite the quadratic cost functional as

$$J = \frac{1}{2} \underline{x}^T \underline{S} \underline{x} + \frac{1}{2} \int_{t_0}^{T_f} (\underline{x}^T \underline{Q} \underline{x} + \underline{u}^T \underline{R} \underline{u}) dt \quad (4.4.4)$$

where this time

\underline{S} is a constant positive semi-definite matrix

\underline{Q} is a constant positive semi-definite matrix

\underline{R} is a constant positive definite matrix

Instead of the control law formulated by Eqs. (4.2.9), (4.2.22) and (4.2.23), which is very hard to determine and implement, we suggest the following suboptimal feedback control

$$\underline{u}(t) = \underline{G}_k \underline{x}(t) \quad (4.4.5)$$

where \underline{G}_k is a linear feedback matrix, constant between two corrections made at adaptation times t_k and t_{k+1} . The optimal control is approximated by a linear control with constant gain.

The nonlinear state equations are approximated by a linear model of the form

$$\dot{\underline{x}} = \underline{A}_k(\underline{x}_k, \underline{u}_k) \underline{x} + \underline{B}_k(\underline{x}_k, \underline{u}_k) \underline{u} \quad (4.4.6)$$

valid at the correction time t_k and \underline{A}_k and \underline{B}_k are the coefficient matrices of the linearized system formed through apparent linearization.

A linear time-invariant problem is defined by Eqs. (4.4.4) and (4.4.6) if \underline{A}_k and \underline{B}_k are considered as constants as long as the nonlinear system state stays in the validity domain of the linear model (4.4.6). The solution of this problem yields:

$$\underline{u}(t) = -\underline{R}^{-1} \underline{B}_k^T \underline{K}(t) \underline{x}(t) \quad t \in [t_0, T_f] \quad (4.4.7)$$

where $\underline{K}(t)$ is the Riccati matrix.

Further approximation is introduced by taking $t_0 = t_k$, considering only $\underline{K}_k = \underline{K}(t_k)$ and keeping it constant until another correction. The final time T_f can be redefined at each correction. Using this approximation we will have in the suboptimal control law (4.4.5)

$$\underline{G}_k = -\underline{R}^{-1} \underline{B}_k^T \underline{K}_k \quad t \in [t_k, t_{k+1}] \quad (4.4.8)$$

Our interest is now focused on the computation and adaptation of the gain matrix \underline{G}_k .

Canonical equations for the linear problem formulated in Eqs.(4.4.4) and (4.4.6) have the form

$$\begin{bmatrix} \dot{\underline{x}}(t) \\ \dot{\underline{\lambda}}(t) \end{bmatrix} = \begin{bmatrix} \underline{A}_k & -\underline{B}_k^T \underline{R}^{-1} \underline{B}_k \\ -\underline{Q} & -\underline{A}_k^T \end{bmatrix} \begin{bmatrix} \underline{x}(t) \\ \underline{\lambda}(t) \end{bmatrix} \quad (4.4.9)$$

The solution of this homogeneous equation is given by

$$\begin{bmatrix} \underline{x}(t_2) \\ \underline{\lambda}(t_2) \end{bmatrix} = \underline{\phi}(t_2; t_1) \begin{bmatrix} \underline{x}(t_1) \\ \underline{\lambda}(t_1) \end{bmatrix} \quad (4.4.10)$$

where

$$\underline{\phi}(t_2; t_1) = e^{\underline{D}_k(t_2 - t_1)} \quad (4.4.11)$$

with

$$\underline{D}_k = \begin{bmatrix} \underline{A}_k & -\underline{B}_k^T \underline{R}^{-1} \underline{B}_k \\ -\underline{Q} & -\underline{A}_k^T \end{bmatrix} \quad (4.4.12)$$

and \underline{K}_k is obtained from (4.4.10) as

$$\underline{K}_k = (\underline{\phi}_{22} - \underline{S} \underline{\phi}_{12})^{-1} (\underline{S} \underline{\phi}_{11} - \underline{\phi}_{21}) \quad (4.4.13)$$

where $\underline{\phi}_{ij}$ are $n \times n$ submatrices of $\underline{\phi}(T_f; t_k)$.

If we compare the above method with that in Section 4.2 (Eqs.(4.2.9) through (4.2.23)); we see that the method outlined above involves an easy way of computing the fundamental matrix $\underline{\phi}(T_f; t_k)$.

The computation of the matrix exponential in Eq.(4.4.11) is performed in the following way to improve the precision and to lessen the computation time.

The fundamental matrix (i.e., $\exp[\underline{D}_k(T_f - T_k)]$) is computed by use of third order Pade approximation (see for example Varga (11)) with a step size s such that

$$T_f - t_k = 2^{kk} s \quad (4.4.14)$$

Van Loan's criterion (12) is adopted to determine the doubling number kk .

$$\frac{\|\underline{D}_k(T_f - t_k)\|}{2^{kk}} \leq \frac{1}{2} \quad (4.4.15)$$

where $\| \underline{D}_{-k} (T_f - T_k) \|$ is the Frobenius norm defined as

$$\| \underline{Y} \| = \left(\sum_i \sum_j |y_{ij}|^2 \right)^{1/2} \quad (4.4.16)$$

After obtaining $\underline{\phi}(s) = e^{\underline{D}_{-k}s}$, $\underline{\phi}(T_f - t_k)$ is computed by use of the doubling formula

$$\underline{\phi}_i = \underline{\phi}_{i-1} \cdot \underline{\phi}_{i-1} \quad (4.4.17)$$

Submatrices of $\underline{\phi}_{-kk}$ are finally replaced in (4.4.13) and \underline{K}_{-k} is computed.

The gain matrix \underline{G}_{-k} is obtained from Eq.(4.4.5).

V. COMPUTER SIMULATION

5.1 Description of Computer Programs Used

A computer program consisting of a main program together with numerous special and general subroutines is used in the computer simulation of the intercept problem. A short description of the main program and the routines follows. Only the simulation variables read in the main program are considered as inputs and similarly only the output of the main program is considered. The input and the output lists of the subroutines can be found in the Appendix (at least for most of the routines) and are not listed here.

The main program reads the necessary constants and tabulated variables used in Eqs.(3.1.1)-(3.1.4) from a data file named MISDAT. The Q and R matrices are formed (Q is set to 0 and R = 1 , R is 1x1). The nonzero values of the S matrix are read in and echo printed if suboptimal control is sought. Then, the simulation variables related to intercept geometry, escape policy of the target etc. are read in. The list of the simulation variables follows:

H ϕ : altitude (km)
 R ϕ : range (initial distance between the missile and the target, m)
 SIGMA : line-of-sight angle σ (deg)
 MM : initial Mach number of the missile
 MT : initial Mach number of the target
 GT : lateral acceleration of the target (in g's)
 PNG ϕ : initial PNG
 ATANGE : tangential acceleration of the target (m/s²)

DELTAT : time at which the target begins to accelerate (s)
 DELTAR : time at which target evasive manoeuvre begins (s)
 AASC : angle of attack stabilization constant
 X(4) : angle of launch θ (deg)

The target lateral acceleration is in g's following conventions used in literature (e.g., we speak of a target manoeuvre of 5 g's). PNG is initially set to the specified value (we have experimented with a variable PNG also). During simulation variable target velocity is taken into consideration as well. The simulation variable ATANGE enables us to study the case in which the target has tangential acceleration as well as undergoing a manoeuvre. DELTAT and DELTAR specify the time at which the accelerating and turning manoeuvres of the target starts. They are usually set to 2.2 secs since the missile control is activated after 2.2 seconds following launch.

After the intercept geometry and initial conditions are specified the state equations (Eqs.(3.4.1)-(3.4.9) and the trajectory equations (3.1.29)-(3.1.30) are numerically integrated using Runge-Kutta IV with a time increment of 0.01 seconds initially. To improve resolution time step, is set to 0.005 when R becomes less than 50 m and to 0.001 when R is less than 5 m. If we note that the velocity of the missile is about 500 m/s we see that in one time step of 0.01 secs. the position of the missile changes about 5 m. Since we are after obtaining miss distances of the order 1 m, a time step of 0.001 is more suitable when R becomes small.

While the differential equations are being integrated the necessary control is evaluated using either PNG or suboptimal adaptive control schemes. The value of the state variables is printed out every 0.1 seconds. The simulation continues till the missile and target separation begins to increase. When R begins to increase the program stops after printing the final values of the state variables and the below listed output.

RMIN : miss distance (m)
TIME : time the simulation ended (seconds)
ALPAV : time average of angle of attack (deg)
ALPMAX : maximum value of angle of attack (deg)

Below is a list and brief description of the subroutines used.

SUBROUTINE RUNGE:

This subroutine is a general purpose subroutine which integrates a system of differential equations using Runge-Kutta IV method.

SUBROUTINE MISNL:

In this subroutine the right hand side of the state and trajectory equations are evaluated. The coefficients used in these equations (C_c , C_N and C_m) are computed as well. The necessary inputs and outputs are supplied by mainly COMMON blocks in this and the following three subroutines.

SUBROUTINE MISVAR:

The center of gravity, mass and inertia variation of the missile are evaluated in this routine.

SUBROUTINE MISCOEF:

In this routine various tabulated coefficients used in the differential equations are evaluated using three different interpolation subroutines. Also, some error checks are performed to ensure that the missile is within its operation limits (e.g., it is checked that the Mach number for the missile is between 0.8 and 3.2).

SUBROUTINE MISFIN:

In this subroutine four checks are performed. It is checked that: 1) the missile is statically stable (in fact the missile turns out to be statically unstable for the first 0.14 seconds after launch); 2) the velocity of the missile is greater than that of the target; 3) the Mach

number of the missile is greater than 0.8; and 4) the missile is approaching the target and not moving away from it. This last check is of primary importance since it constitutes the criterion for ending the simulation.

SUBROUTINE'S INTPOL, INTER and INTER1 :

These all perform linear interpolation for tabulated variables and the only difference between them is the manner the tabulated variables are placed in arrays.

SUBROUTINE MATRIX :

In this subroutine the coefficient matrices \underline{A}_k and \underline{B}_k are computed at each adaptation. The state equations are linearized via apparent linearization technique and are scaled as discussed in a later section.

SUBROUTINE GAIN1 :

This subroutine computes the linear optimal feedback gains corresponding to a given horizon time T_f with a prescribed degree of stability (in our case we did not prescribe stability). In essence this routine numerically performs the task outlined in Section 4.4.

This routine is the backbone of the suboptimal control scheme. It is a general purpose subroutine which together with the following routines can be used to solve any linear quadratic optimal control problem of the form specified by Eqs.(4.4.4) and (4.4.6).

SUBROUTINE CANM :

This subroutine forms the Hamilton matrix of the linear quadratic optimal control problem defined for the system \underline{A} , \underline{B} with the penalization matrices \underline{Q} , \underline{R} in the form

$$F = \begin{bmatrix} \underline{A} & -\underline{B}^T \underline{R}^{-1} \underline{B} \\ -\underline{Q} & -\underline{A}^T \end{bmatrix} \quad (5.1.1)$$

SUBROUTINE KKH :

This subroutine computes the largest convergent step size to be used in the computation of the exponential of a matrix. It also indicates the number of doubling operations necessary to cover the desired time interval.

SUBROUTINE EXPF3 :

This subroutine computes the exponential of a general square matrix for a given time interval using third order Pade approximation.

SUBROUTINE PART :

This subroutine partitions a nxm general matrix into four submatrices A, B, C and D delimited at the np'th line and the mp'th column in the following way

$$\underline{F} = \begin{bmatrix} \underline{A} & \underline{B} \\ \underline{C} & \underline{D} \end{bmatrix} \quad \begin{matrix} np \\ mp \end{matrix} \quad (5.1.2)$$

SUBROUTINE GMPRD :

This subroutine is used to multiply two general matrices to form a resultant matrix.

SUBROUTINE MINV :

This subroutine inverts a matrix. The standard Gauss-Jordan method is used. The determinant is also calculated. The input matrix is destroyed in computation and is replaced by the resultant inverse.

5.2 Scaling

A closer inspection of Eqs.(4.3.6) through (4.3.13) reveals that the coefficients of the \underline{A}_k and \underline{B}_k matrices are not of the same order. While some are of the order 1000 some are of the order 10^{-3} . Thus, some sort of scaling seems necessary. We scaled the equations as follows. Consider a differential equation of the form

$$\dot{x}_i = a_{i1} x_1 + a_{i2} x_2 + \dots + a_{in} x_n + b_i u \quad (i = 1, \dots, n) \quad (5.2.1)$$

Define new variables as

$$X_i = \frac{x_i}{x_{imax}} \quad (i = 1, \dots, n) \quad (5.2.2)$$

$$\dot{X}_i = \frac{\dot{x}_i}{\dot{x}_{imax}} \quad (i = 1, \dots, n) \quad (5.2.3)$$

where x_{imax} is the maximum value of x_i , and \dot{x}_{imax} is the maximum value of \dot{x}_i . We can rewrite Eq.(5.2.1) as

$$\dot{x}_{imax} \dot{X}_i = a_{i1} x_{imax} X_1 + a_{i2} x_{2max} X_2 + \dots + a_{in} x_{nmax} X_n \quad (5.2.4)$$

or equivalently

$$\dot{X}_i = a_{i1} \frac{x_{imax}}{\dot{x}_{imax}} X_1 + a_{i2} \frac{x_{2max}}{\dot{x}_{imax}} X_2 + \dots + a_{in} \frac{x_{nmax}}{\dot{x}_{imax}} X_n + \frac{b_i}{\dot{x}_{imax}} u \quad (5.2.5)$$

Thus, a_{ij} and b_i are transformed to a'_{ij} and b'_i such that

$$a'_{ij} = a_{ij} \frac{x_{jmax}}{\dot{x}_{imax}} \quad (5.2.6)$$

$$b'_i = \frac{b_i}{\dot{x}_{imax}} \quad (5.2.7)$$

This scaling has the desired effect. The elements of the matrices \underline{A}_k and \underline{B}_k become closer to each other orderwise. Another advantage is that large order terms are no longer present. Hence, the doubling number kk , which is dependent on the norm of Hamilton matrix, is smaller and less computation is necessary in evaluating the suboptimal control.

However, one has to be careful in integrating the scaled equations.

Since

$$x_i = \int \dot{x}_i dt \quad (5.2.8)$$

and

$$x_{imax} X_i = \int \dot{x}_{imax} X_i dt \quad (5.2.9)$$

We obtain

$$X_i = \frac{\dot{x}_{imax}}{x_{imax}} \int X_i dt \quad (5.2.10)$$

In scaling the state equations the following maximum values for x_i and \dot{x}_i were used

	i =	1	2	3	4	5	6	8	9
x_{imax}	:	50	820	0.1	1	0.002	0.1	5000	5000
\dot{x}_{imax}	:	30	230	0.1	0.08	0.01	0.01	500	500

5.3 Scenarios

The computer simulation was repeated for several different scenarios. Different initial conditions were investigated. Scenarios differed in launch conditions as well as target escape manoeuvres. The launch conditions used in simulations are shown in Fig.5.3.1.

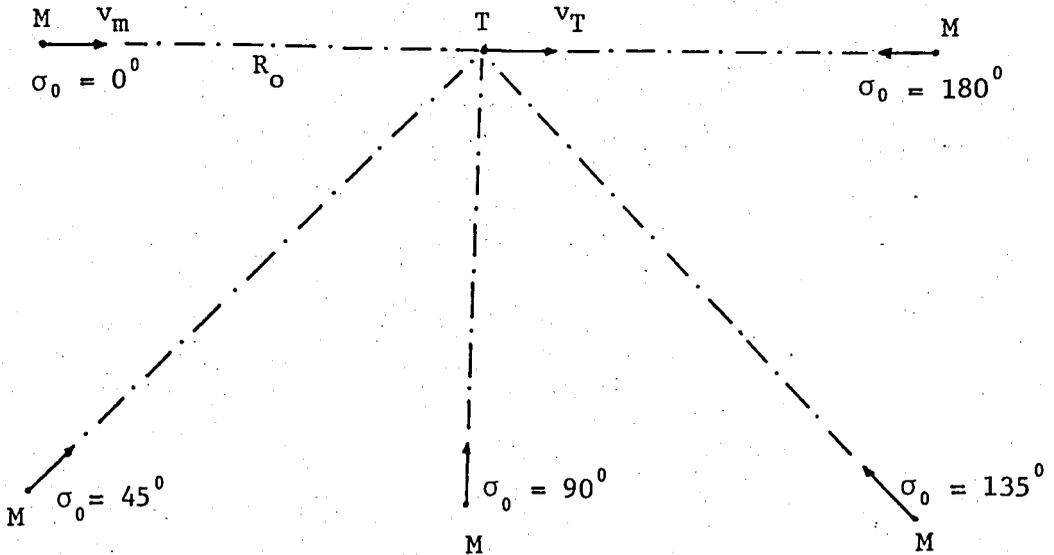


Figure 5.3.1 Launch scenarios

The escape policy of the target can be two fold. It may execute a (specified number of g) turn either towards the missile or away from it and the target may also accelerate; i.e., increase its speed (an escape policy based on target deceleration is not effective). A more detailed account of scenarios and escape policies can be found in the section on results obtained by simulation with PNG law.

5.4 Time-to-go Estimation

The horizon time T_f has to be approximated before suboptimal adaptive control scheme can be applied. The usual procedure in estimating the homing time remaining before a missile intercepts the target involves the quotient of the instantaneous range by the range rate and is of the form (see Fig.5.4.1)

$$T_f = -\frac{R}{\dot{R}} \quad (5.4.1)$$

Noting that

$$R = \sqrt{e_x^2 + e_y^2} \quad (5.4.2)$$

and differentiating Eq.(5.4.2) with respect to time we obtain

$$T_f = \frac{-R^2}{e_x \dot{e}_x + e_y \dot{e}_y} \quad (5.4.3)$$

Eq.(5.4.3) is exact only if the missile is on a perfect collision course, i.e., $\dot{\sigma} = 0$. If the missile suffers from a heading error, $\dot{\sigma}$ is not zero, and Eq.(5.4.3) is only an approximation, sometimes quite bad, depending on the size of missile heading error deviation from the true homing course. Also Eq.(5.4.3) implies that some sort of ranging method exists, either onboard the missile or ground based. Ranging equipment carried on board the missile is usually of the active type, e.g., radar. This involves both weight and power penalties as well as possible detection by the target prior to interception. Passive sensors, e.g., infrared or optical, of either line-of-sight angle or angular measurement type are lighter and are not readily detected. However, they lack range-finding ability and consequently cannot be used to provide a direct estimation of the homing time by the preceding technique.

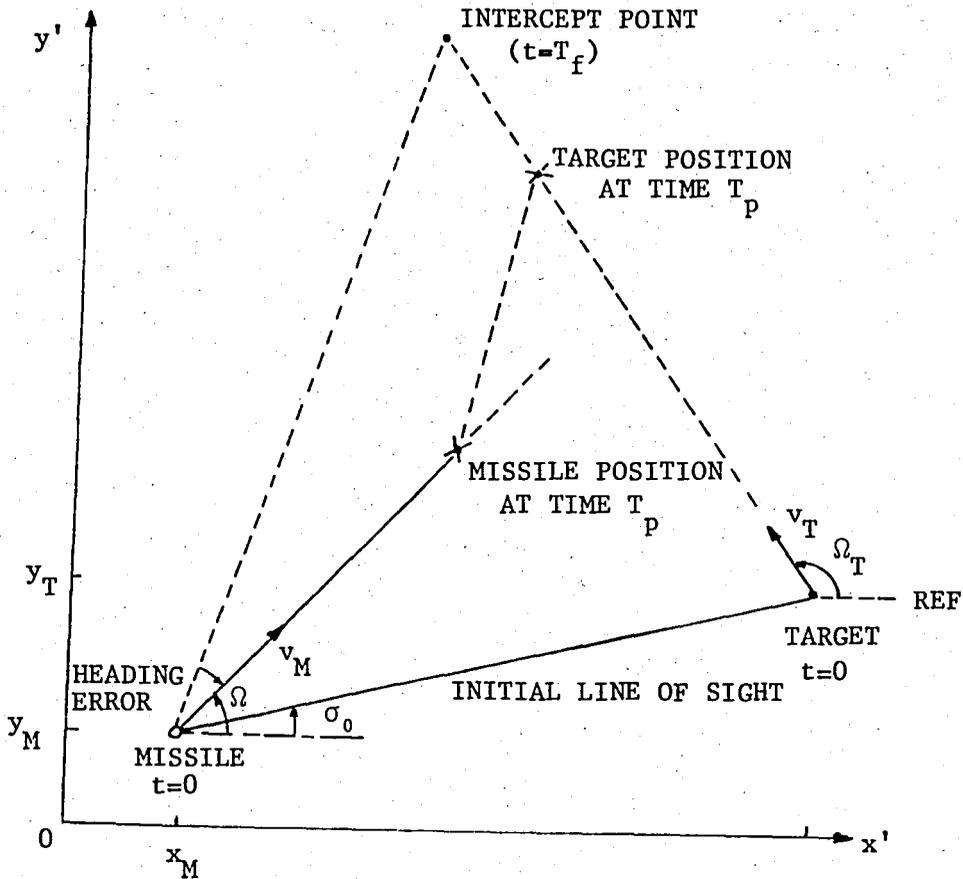


Figure 5.4.2 Homing time estimation

In a technical note Rawling (12) develops a homing time relation which is applicable to passive, angle/rate measuring sensors and is valid when the missile is not on a perfect intercept course. His relation, instead of representing the time-to-go until interception; provides the time to the point of minimum separation between missile and target, since there will be no interception if $\dot{\sigma} \neq 0$. This time-to-minimum separation is denoted by T_p , and is referred to as "time-to-pass" in order to distinguish it from T_f , the time-to-go. A fundamental property T_p is that it reduces to T_f whenever $\dot{\sigma} = 0$, i.e., whenever the missile is homing perfectly. The relation developed by Rawling is

$$T_p = \frac{2 \dot{\sigma} \ddot{\sigma}}{\ddot{\sigma}^2 \pm 4\sigma^4} \quad (5.4.4)$$

Eq.(5.4.4) is a completely passive expression for the homing time remaining from t_0 to the point of minimum separation, involving neither range nor range rate; furthermore, it is valid for large missile heading errors as long as the missile is approaching the target ($\dot{R} < 0$).

Throughout this study Eq.(5.4.3) was used for approximating the time-to-go since heading error is sufficiently small during a great portion of flight.

VII. SIMULATION RESULTS AND DISCUSSION

6.1 General Results

As mentioned earlier the missile accelerates during the first 2.2 seconds after launch and then slowly decelerates due to drag forces. The velocity profile of the missile (for zero angle of attack) is shown in Fig.6.1.1. This profile, which clearly exhibits the boost-coast character of velocity, is for an altitude of 10 km and an initial Mach number of 1.2.

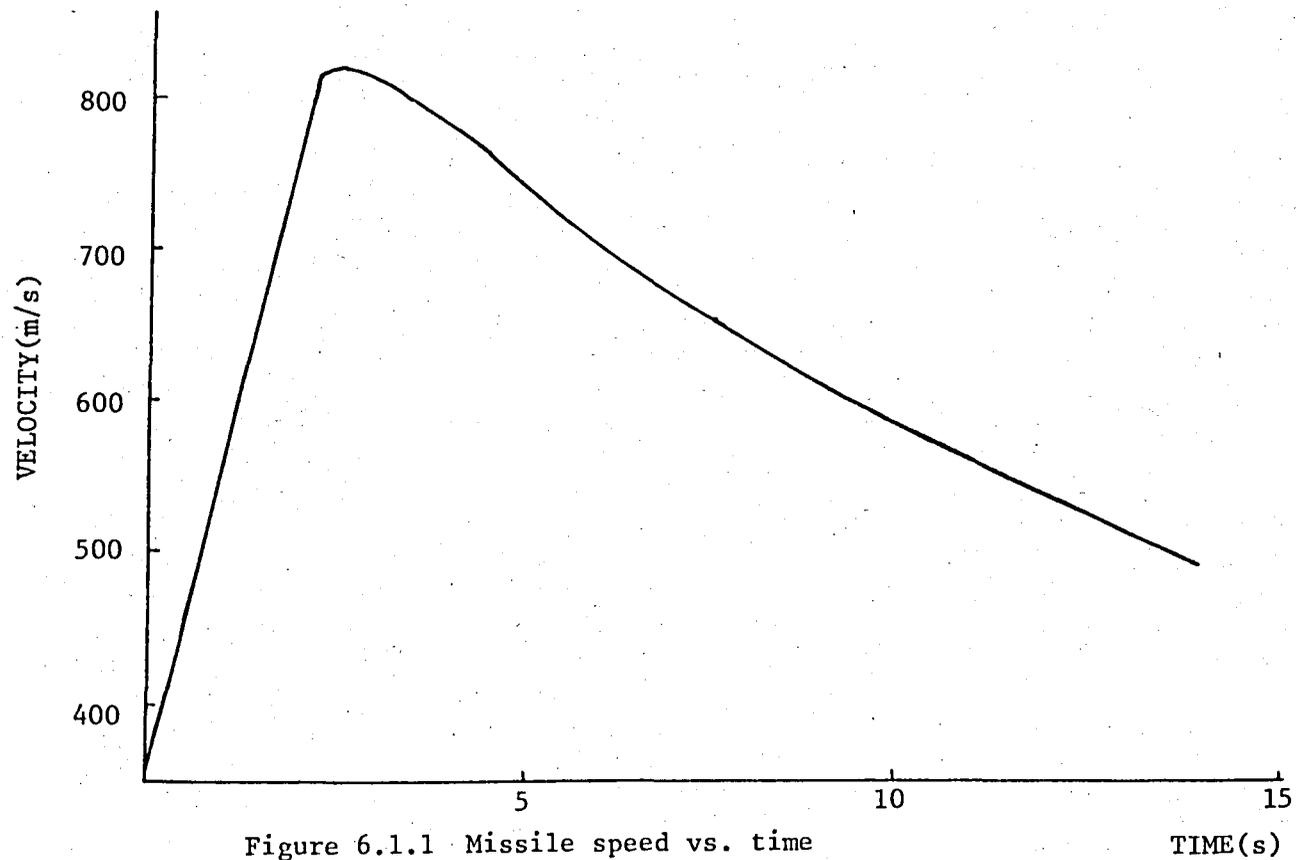


Figure 6.1.1 Missile speed vs. time

TIME(s)

At this altitude the velocity of sound is about 300 m/s. As can be seen from Fig.6.1.1 the maximum velocity of the missile is about 2.73 M (or equivalently 820 m/s). In a more realistic simulation where the missile is trying to intercept a target the missile would decelerate faster due to drag forces which increase with incidence (angle of attacks). As an extreme case we performed an experimental simulation. We set flipper deflection (δ_z) equal to 26° at $t = 2.2$ seconds, and kept it constant later. As a result velocity decreased much faster than that in Fig.6.1.1. At $t = 10$ seconds velocity decreased to 425 m/s compared to 600m/s for zero angle of attack.

The flight path both for the missile and the target are plotted in Figs. 6.1.2 to 6.1.6 for each of the launch conditions shown in Fig.5.3.1. These example plots show typical scenarios with simple PNG laws. The simulation variables that are common in all are listed below and those that differ in value are stated after the relevant figure.

H ϕ : 10 km (altitude)
 MM : 1.2 M (Mach number of the missile)
 MT : 0.95 M (Mach number of the target)
 ATANGE : 10 m/s² (tangential acceleration of the target)
 DELTAT : 2.2 secs
 DELTAR : 2.2 secs
 AASC : 0.

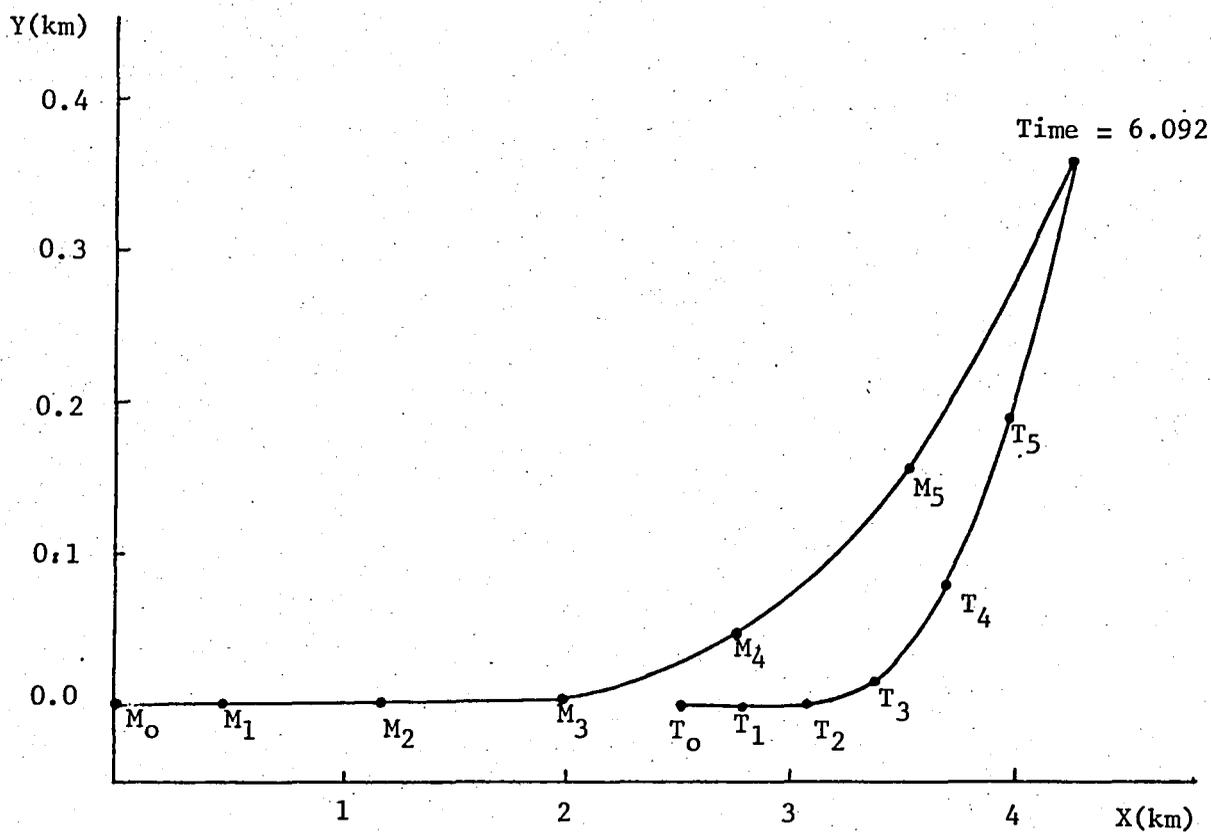


Figure 6.1.2 Trajectories for $\sigma_0 = 0.0^\circ$

For Fig.6.1.2

$$R\phi = 2.5 \text{ (km)}$$

$$GT = 5 \text{ (g's)}$$

$$PNG = 6$$

$$\theta_0 = 0^\circ$$

In Fig.6.1.2 $M_0, M_1 \dots$ shows the position of the missile at $t = 0, 1, 2, \dots$ seconds and similarly $T_0, T_1 \dots$ shows the position of the target at $t = 0, 1, 2, \dots$ seconds. Also the intercept time is shown.

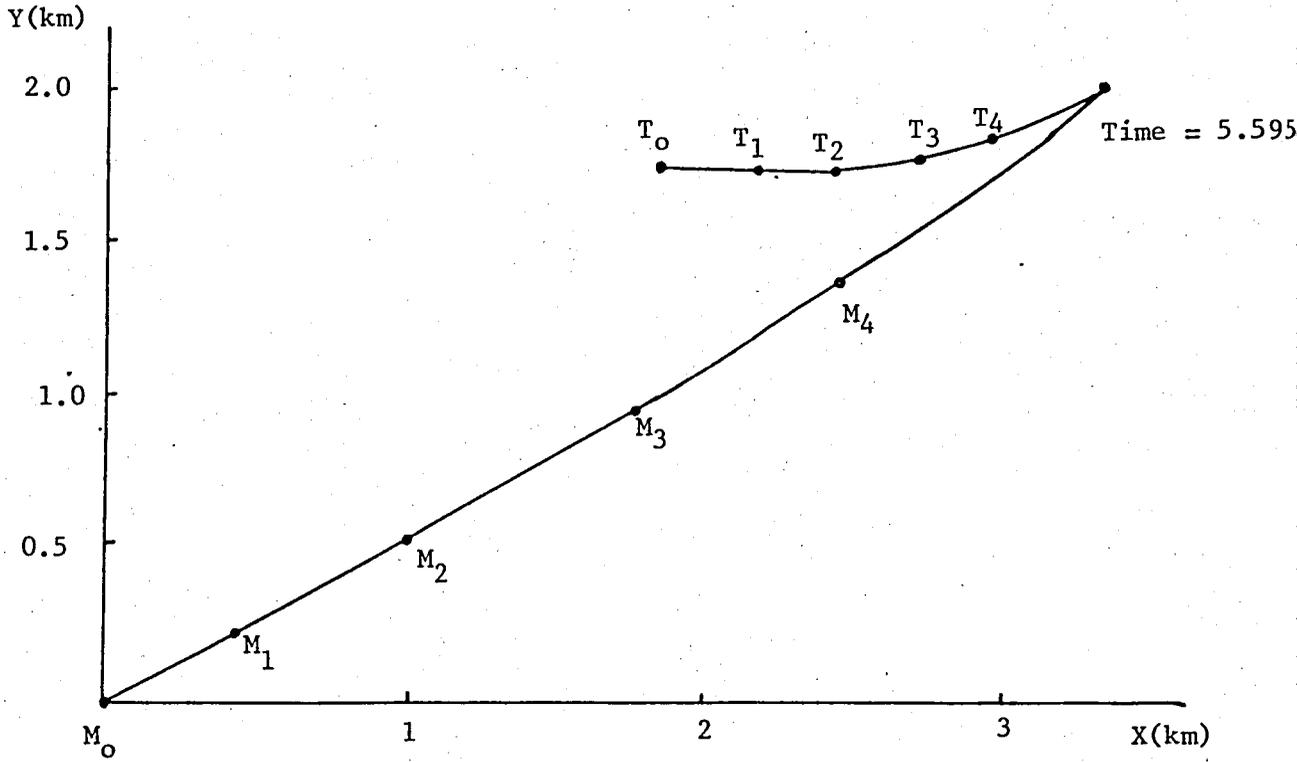


Figure 6.1.3 Trajectories for $\sigma_0 = 45^\circ$

For Fig.6.1.3

$$GT = 5 \quad (\text{g's})$$

$$R\phi = 2.5 \quad (\text{km})$$

$$PNG = 6$$

$$\theta_0 = 28.8^\circ$$

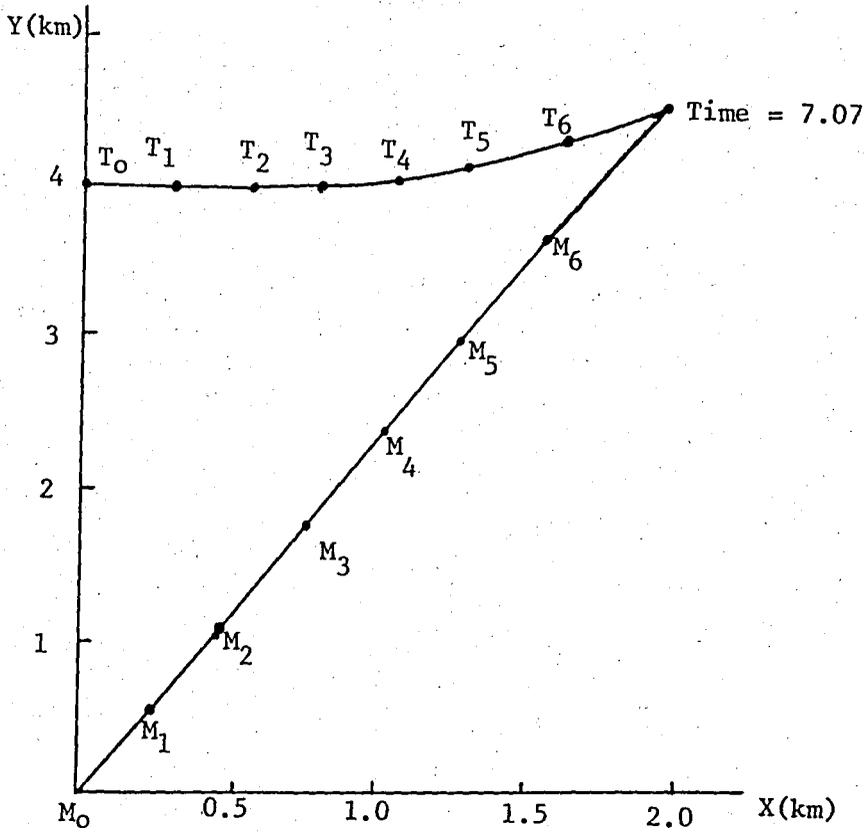


Figure 6.1.4 Trajectories for $\sigma_0 = 90^\circ$

For Fig. 6.1.4

$$R\phi = 4 \text{ (km)}$$

$$GT = 5 \text{ (g's)}$$

$$PNG = 6$$

$$\theta_0 = 67.2^\circ$$

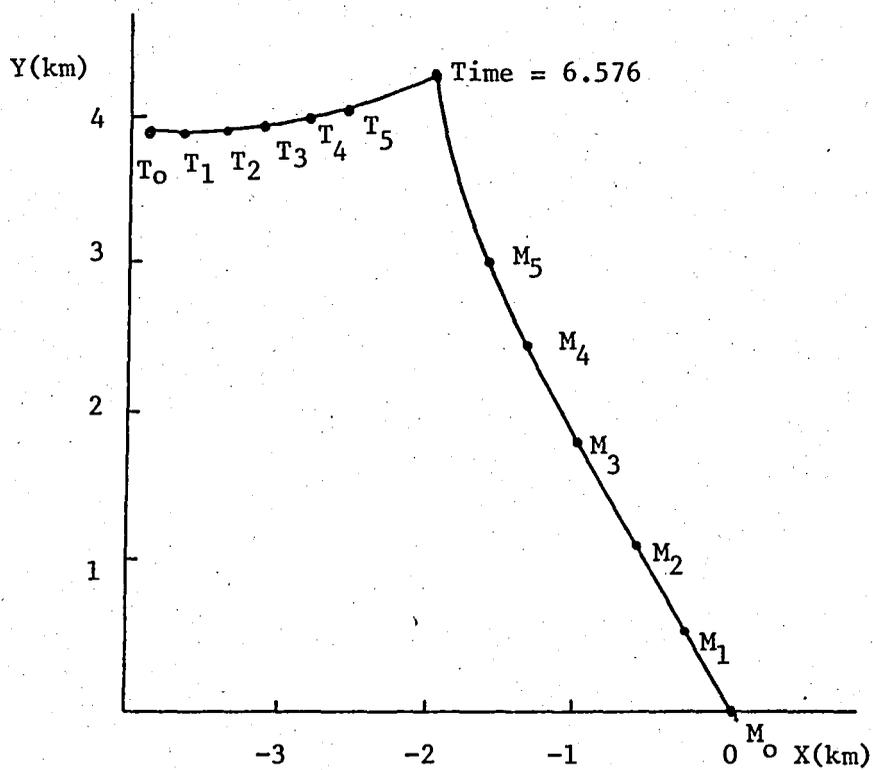


Figure 6.1.5 Trajectories for $\sigma_0 = 135^\circ$

For Fig.6.1.5

$$R\phi = 5.5 \text{ (km)}$$

$$GT = 3 \text{ (g's)}$$

$$PNG = 10$$

$$\theta_0 = 119.1^\circ$$

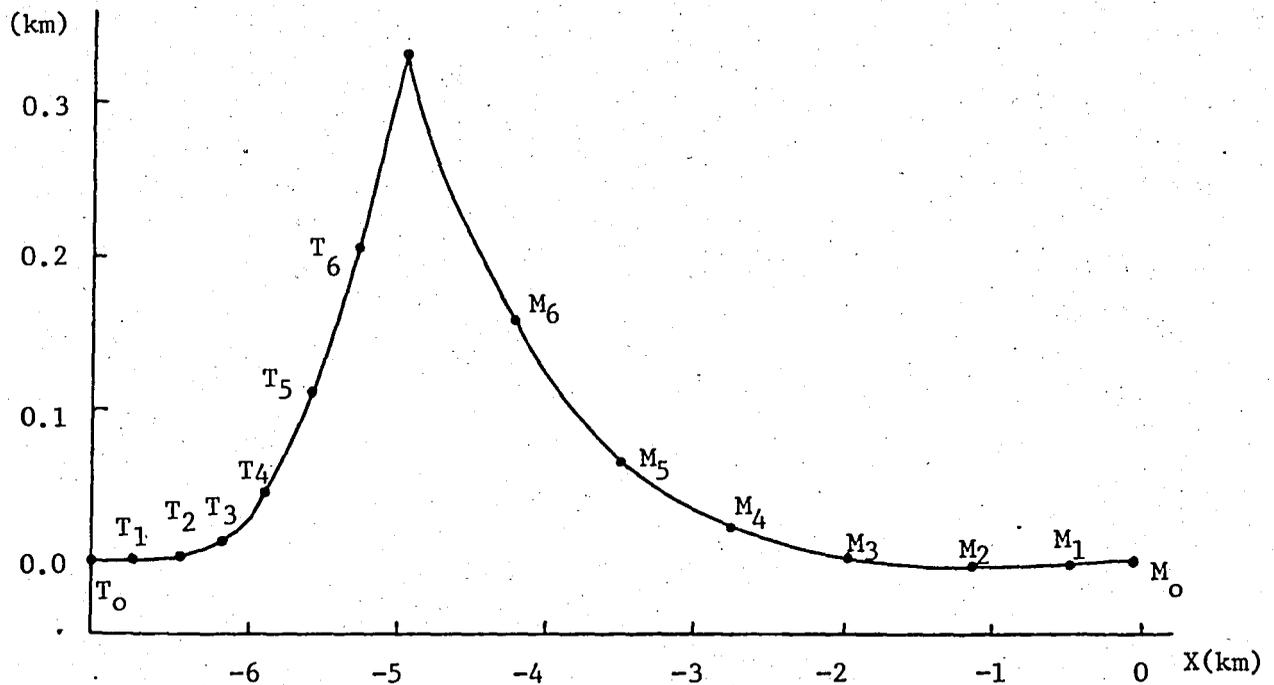


Figure 6.1.6 Trajectories for $\sigma_0 = 180^\circ$

For Fig.6.1.6

$$R\phi = 7 \text{ (km)}$$

$$GT = 3 \text{ (g)}$$

$$PNG = 10$$

$$\theta_0 = 180^\circ$$

6.2 Simulation with PNG Law

First let us consider the launch conditions. In section 3.3 we had discussed two approaches to launch conditions and concluded that lead collision approach was likely to yield more successful results compared to lead pursuit. However, during computer simulation it was observed that for scenarios in which σ_0 (initial LOS angle) is 45° , 90° , or 135° lead collision approach also necessitated quite large controls and resulted in big changes in the lateral component of missile velocity when the control was initially activated (i.e., just after 2.2 seconds). For a typical scenario with the simulation variables

$H\phi$	=	10 km
$R\phi$	=	2500 m
σ	=	45°
MM	=	1.2
MT	=	0.95
GT	=	5
PNG	=	6
ATANGE	=	10 m/s^2
DELTAT	=	2.2 s
DELTAR	=	2.2 s
AASC	=	0.0

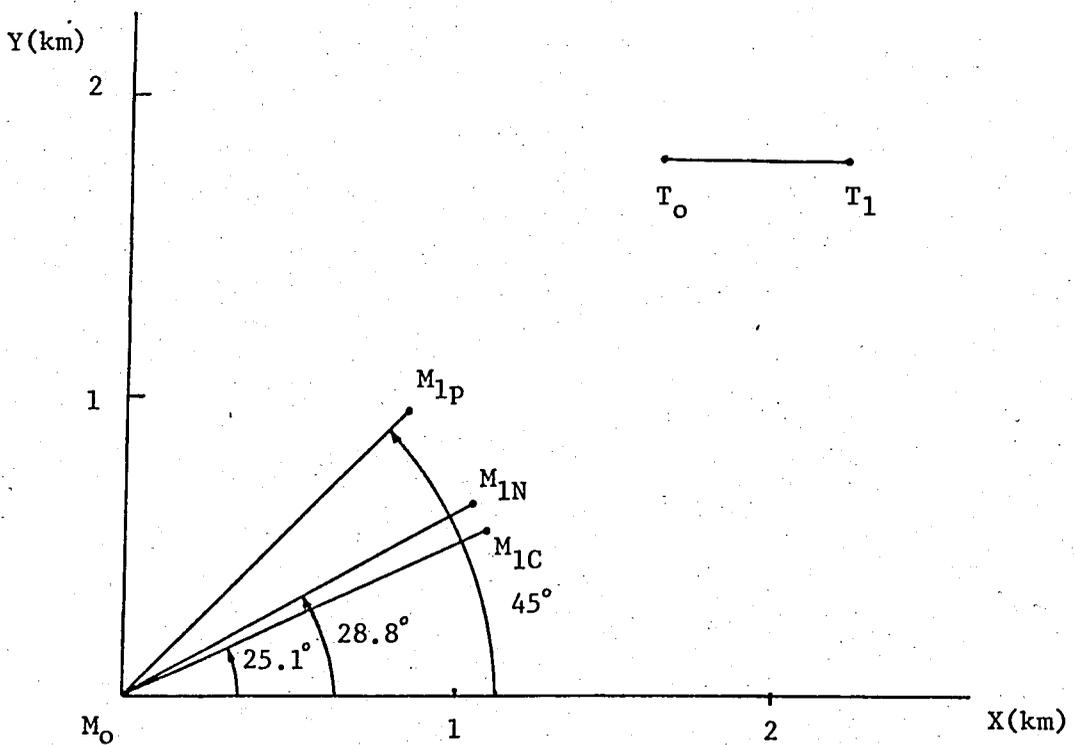
the lateral velocity of the missile for launch angles, θ , of 45° (lead pursuit), 25.185 (lead collision) and 28.8° (a value found by trial and error in numerical experiments) in the time interval 2.3-3.0 seconds is listed in Table 6.2.1.

For convenience these three launch conditions and the positions of the missile and target at $t = 0$ and $t = 2.2$ seconds are shown in Fig.6.2.1. T_0 and T_1 show the position of the target at $t = 0$ and $t = 2.2$ seconds respectively. Similarly, M_0 is the position of the missile at $t = 0$ and M_{1p} is the position of the missile for lead pursuit approach, M_{1c} for lead collision and finally M_{1N} for $\theta = 28.8^\circ$ at $t = 2.2$ secs.

The missile intercepts the target for all three launch angles but examining the above listed values for w (lateral velocity of missile) we see that a launch angle of 28.8° is superior to both lead collision and lead pursuit approaches. In lead pursuit approach the missile is aimed at a point too backwards the target is at $t = 2.2$ secs and this necessitates a large manoeuvre on behalf of the missile. Lead collision approach results in missile being ahead of the target. Best launch angle is

$t(\text{sec}) \backslash \theta_0$	45°	25.185°	28.8°
2.3	-140.5	61.3	0.7
2.4	-303.7	136.2	6.2
2.5	-115.7	127.0	14.8
2.6	-60.9	112.6	27.5
2.7	-250.5	107.9	27.8
2.8	-206.6	104.0	31.4
2.9	-63.1	99.6	34.2
3.0	-167.7	95.8	36.7

Table 6.2.1 Lateral velocity variation with time

Figure 6.2.1 Missile and target positions at $t=0$, $t=2.2$ secs

seen to be $\theta_0 = 28.8^\circ$ which is in between the launch angles for lead collision and lead pursuit. Also the average value of angle of attack (i.e., $\text{atan } w/u$) turns out to be 10.97° for lead pursuit; 6.16 for lead collision and only 4.25° for $\theta_0 = 28.8^\circ$. This is another result which strengthens our claim that $\theta_0 = 28.8^\circ$ is superior to other launch approaches.

In Table 6.2.2 the best launch angles, determined by trial and error, for different initial launch scenarios are listed.

We repeated the computer simulation for different scenarios and tried to determine the limits for successful intercept. The results that follow are for simulation variables $H\phi = 10 \text{ km}$, $MM = 1.2$, $MT = 0.95$, $ATANGE = 10 \text{ m/s}^2$, $DELTAT = 2.2 \text{ s}$, $DELTAR = 2.2 \text{ s}$ and $AASC = 0$. Also the launch angles for the scenarios are those listed in Table 6.2.2 where relevant. The limits for the successful operation of the missile in terms of initial range between target and missile and target evasive manoeuvres and the underlying reasons for those limits are below. We try to investigate each launch condition separately.

First let us consider tail chase (i.e., $\sigma_0 = 0^\circ$, see Fig.6.1.2). In this scenario the initial range $R\phi$ must not be excessively large. ($R_{\text{max}} \sim 5000 \text{ m}$). For ranges between $2000\text{--}4500 \text{ m}$ the missile functions successfully for target manoeuvres in between $\pm 10 \text{ g}'\text{s}$.

For $\sigma_0 = 45^\circ$ (see Fig.6.1.3) and $R\phi = 2500 \text{ m}$ intercept is ensured for $\pm 10 \text{ g}$ manoeuvres by the target. But if $R\phi = 4000 \text{ m}$ the miss distance is acceptable for target manoeuvres in between ± 6 and -10 g . For $R\phi = 5500 \text{ m}$ the missile cannot catch the target if the manoeuvre is away from the missile.

For $\sigma_0 = 90^\circ$ (see Fig.6.1.4) and $R\phi = 2500 \text{ m}$ the successful range is again $\pm 10 \text{ g}'\text{s}$ and for $R = 4000 \text{ m}$ this range slightly differs (± 9 and

σ (deg)	$R\phi$ (m)	Best Launch Angle (deg)
45°	2500	28.8°
45°	44000	29.5°
45°	5500	29.8°
90°	2500	65.75°
90°	4000	67.2°
90°	5500	67.8°
135°	4000	118.5°
135°	5500	119.1°
135°	7000	119.4°

Table 6.2.2 Best launch angles for different scenarios

-12 g's). Again with $R\phi = 5500$ m intercept is not possible with a positive target manoeuvre.

For $\sigma_0 = 135^{\circ}$ (see Fig.6.1.5) and target manoeuvres of ± 7 g's intercept occurs with $R\phi = 4000$ m and $R\phi = 5500$ m.

For head-on collision ($\sigma_0 = 180^{\circ}$, see Fig.6.1.6), $GT = \pm 7$ g's and $R\phi = 5000$ m or $R\phi = 7000$ m the miss distance is about 2 m (which is acceptable) with $PNG = 15$. For $R\phi = 8500$ m and $GT = \pm 5$ g's the miss distance turns out to be about 4 m.

From above limits we see that the missile is effective in different ranges for different LOS angles (σ_0). The missile performs quite well in shorter ranges for $\sigma_0 = 0^{\circ}$ and $\sigma_0 = 45^{\circ}$. For $\sigma_0 = 135^{\circ}$ and $\sigma_0 = 180^{\circ}$ the range must be longer ($R > 4000$ m). Also it turns out that scenarios with $\sigma_0 = 135^{\circ}$ and $\sigma_0 = 180^{\circ}$ are the more difficult ones in terms of allowable target manoeuvres. However, even for those cases the performance is acceptable since a target manoeuvre of ± 7 g's and a linear acceleration of 10 m/s^2 is an appreciable manoeuvre.

The success of the missile is affected by two factors; the speed of the missile and its manoeuvrability. The speed of missile decreases in time due to drag forces and since we consider accelerating targets very large ranges are not allowable. Also the manoeuvring capacity of the missile is limited. The flipper deflection is limited to $\pm 26^\circ$ mechanically and for large g manoeuvres of the target saturation of flipper deflection is observed.

Intercept time varies between about 5 and 13 seconds depending on the range and attack geometry. Larger intercept times are not possible for accelerating targets since the velocity of the missile decreases quite fastly after burnout. For example, with a target acceleration of 10 m/s^2 and initial target velocity of 0.95 M, the velocity of the target is about 400 m/s after 13 seconds while the velocity of the missile is about 500 m/s at the same time (see Fig.6.1.1). The velocity difference is not sufficient anymore. Smaller intercepts times do not allow enough time for the missile to manoeuvre properly.

Also an augmented PNG law, of the form discussed in Section 4.1 (i.e., $\delta_{ze1} = P_N \dot{\sigma} + P_\alpha \dot{\omega}$), with the control signal depending on $\dot{\sigma}$ as well as $\dot{\omega}$ was used in simulation. This guidance law does have some stabilizing effect for especially lead pursuit launches, where appreciable oscillation in both control and lateral velocity is observed initially; but its effect is not of much importance for launch angles listed in Table 6.2.2 for lead collision approach.

Another experimentation was with variable PNG. PNG was increased as range decreased. However, this approach does not improve performance much, since as range decreases LOS rate $\dot{\sigma}$ increases ($\dot{\sigma}$ is inversely proportional to R^2). Also, for large target evasive manoeuvres the flipper deflection is saturated (i.e., reaches its maximum or minimum allowable value) close

to intercept and any further increase in the control cannot improve manoeuvrability of the missile.

The effect of launch angle on miss distance was investigated for two scenarios. An easy case with $GT = 3$, $PNG = 5$ and a relatively more difficult case with $GT = 5$, $PNG = 6$ were considered. In both cases $H\phi = 10$ km, $R\phi = 2500$ m, $MM = 1.2$, $MT = 0.95$, $SIGMA = 45^\circ$, $DELTAT = DELTAR = 2.2$ secs, $ATANGE = 10$ m/s² and $AASC = 0$. The variation of miss distance is plotted in Fig.6.2.2 as a function of launch angle. From Fig.6.2.2 it is seen that the miss distance is zero between launch angles 11.8° and 45.8° for the easy case; and for the difficult case between 13.8° and 49.8° (best launch angle for this attack geometry being 28.8°). The miss distance increases almost "linearly" outside these ranges. It is understandable that in the difficult case the missile is more successful in intercepting the target for larger angles of launch if we note that this specific manoeuvre is such that the target is turning towards the left and also a larger angle of launch means that the missile is aimed towards a point leftwards of the target.

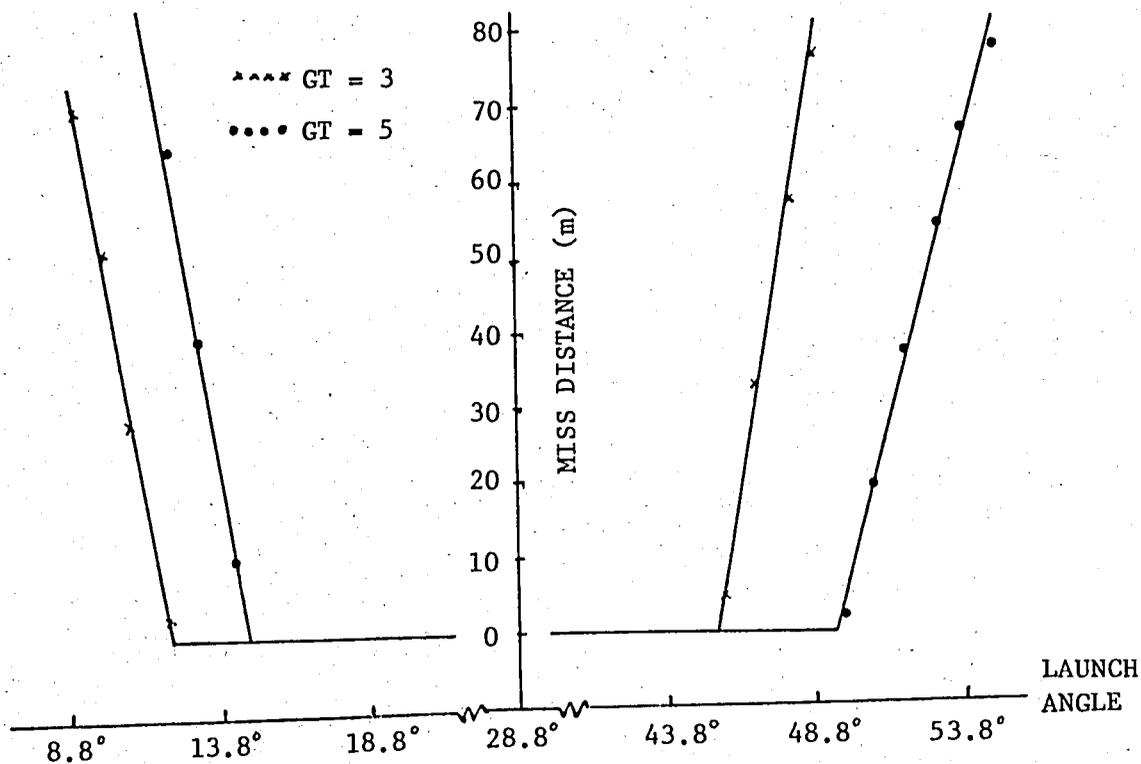


Figure 6.2.2 Miss distance vs launch angle

6.3 Suboptimal Guidance

The suboptimal control for this intercept problem proved to be a formidable task. The difficulties encountered are two-fold. First difficulty is with regards to apparent linearization of the system equations. As was also noted in Section 4.3 the apparent linearization technique does not yield a unique solution to the selection of the coefficient matrices. The most profitable way to linearize the system is hard to guess.

Secondly, the selection of the penalization matrices \underline{S} , \underline{Q} and \underline{R} is an "art rather than science." Here \underline{R} was chosen to be unity and \underline{Q} identically equal to zero, since we are trying to minimize at final time and not continuously during flight. It turns out that the sixth column and row of the matrix \underline{S} is of primary importance and the other elements of \underline{S} have negligible effect.

The variation of control as a result of both PNG and suboptimal adaptive control is plotted in Fig.6.3.1 and the variation of lateral velocity of the missile is plotted in Fig.6.3.2. These plots are both for a typical scenario with $H\phi = 10$ km, $R\phi = 2500$ m, $\sigma_0 = 45^\circ$, $\theta_0 = 28.8^\circ$, $MM = 1.2$, $MT = 0.95$, $ATANGE = 10$ m/s², $DELTAT = DELTAR = 2.2$ s, $GT = 3$. The plots are for $PNG = 5$ and $S(6,6) = 1$, $S(7,6) = S(6,7) = S(8,6) = S(6,8) = 10^4$ respectively.

We observe that the lateral velocity variation closely follows that of the control signal (as would be expected) for both PNG and suboptimal control. Another point is that both $\delta_{z_{e1}}$ and w due to PNG law exhibits a smooth character while those due to suboptimal control is of an oscillatory character. This oscillatory behaviour explains why it is difficult to choose the elements of \underline{S} matrix approximately so that intercept is ensured.

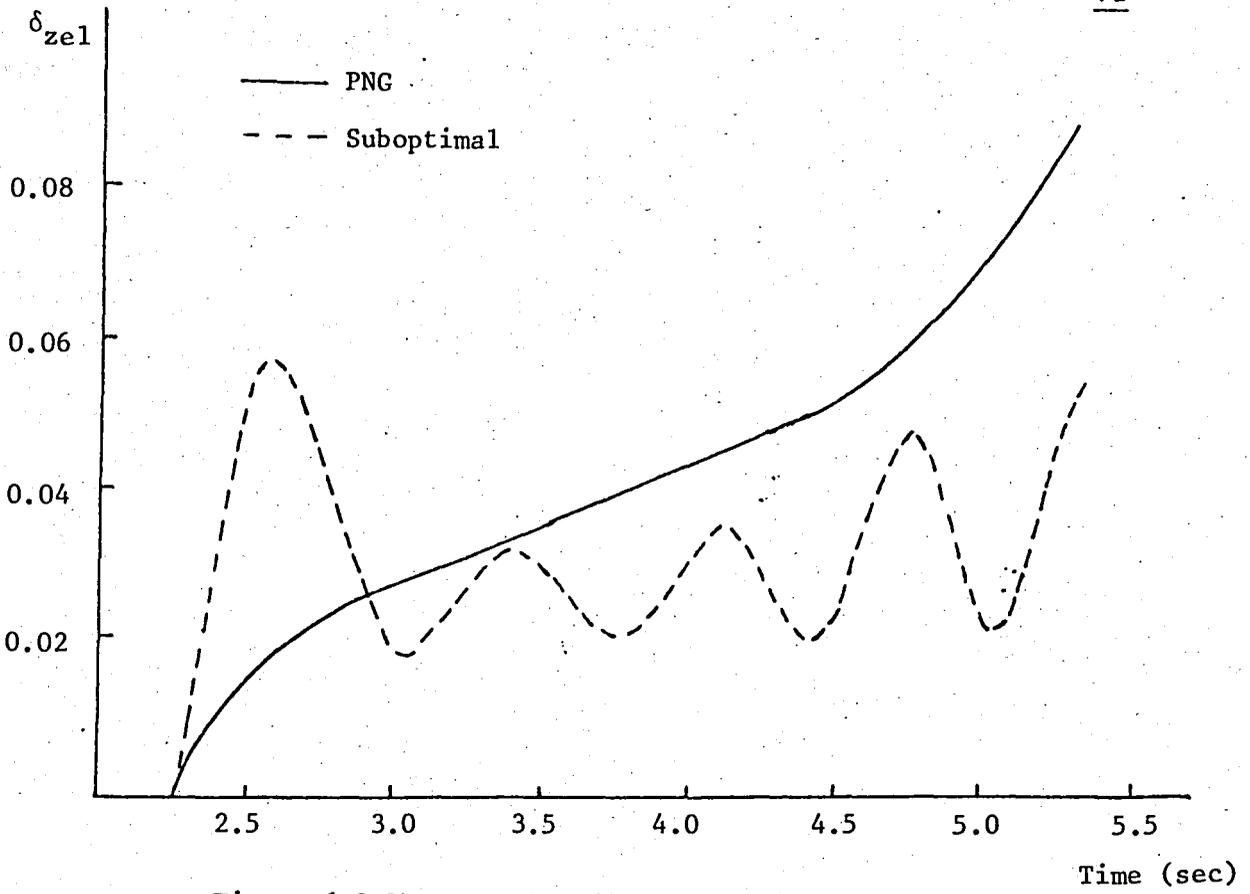
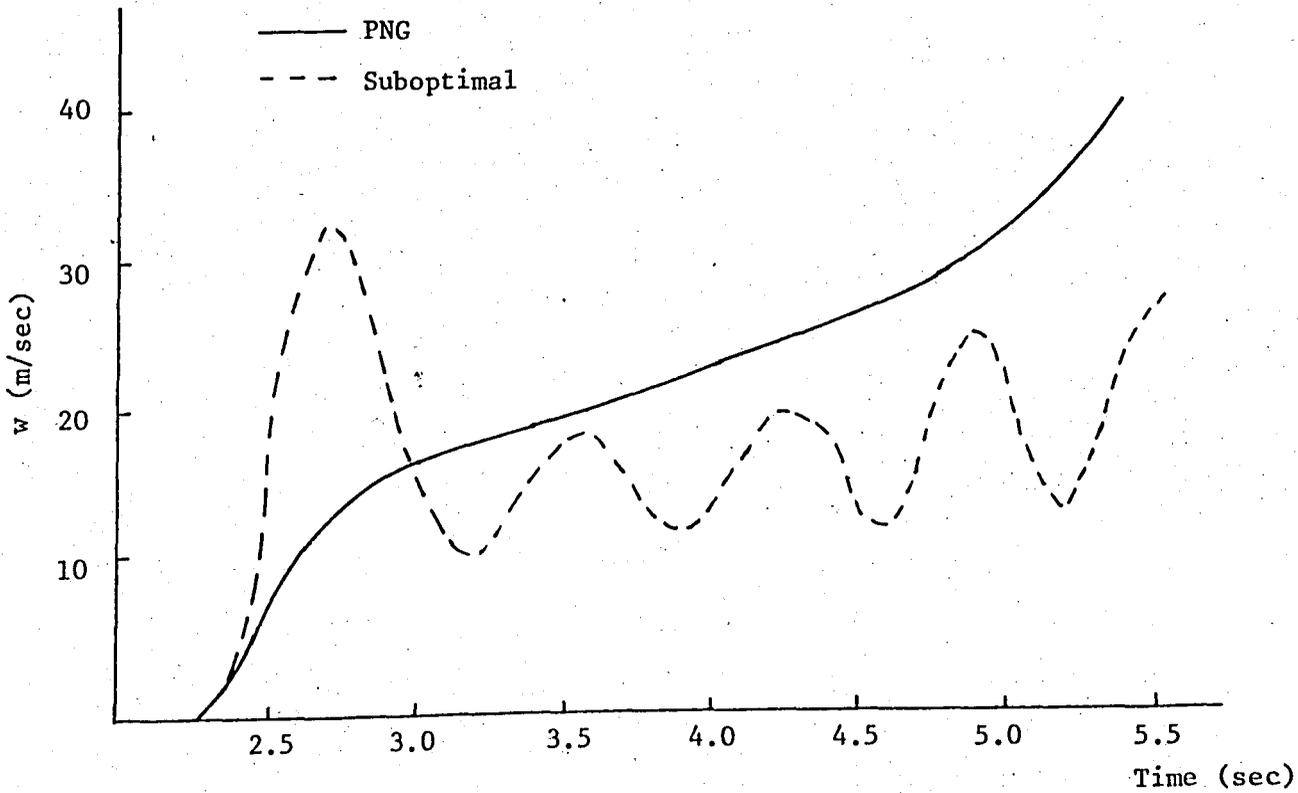


Figure 6.3.1 Control variation with time



It is interesting that quite different values for the \underline{S} matrix can produce similar results. For example the values $S(6,6) = 1$, $S(6,7) = S(7,6) = 750$ and $S(6,8) = S(8,6) = 3$ yields a control very similar to that of Fig.6.3.1. A more stable control was obtained for values $S(6,6) = 1$, $S(6,7) = S(7,6) = 500$ and $S(8,6) = S(6,8) = 20$. This control is plotted in Fig.6.3.3 and the lateral velocity variation produced by it in Fig.6.3.4. For all these three \underline{S} matrices the miss distance is about 1 m. For all, the state matrices were corrected at every 0.1 seconds.

The determination of the best \underline{S} matrix for any scenario seems to require numerous trials and a deep insight to the problem. In a real life application predetermined \underline{S} matrices could be used to depending upon the initial conditions and the target manoeuvre.

While experimenting with suboptimal control it was seen that the control was mainly dependent on e_x and e_y (i.e., x_8 and x_9), as would be expected. Thus, we decided to use only x_6 , x_8 and x_9 as the state variables and evaluate the control using the reduced system. We wrote Eq.(4.3.11) in the form

$$\dot{x}_6 = \left(-\frac{1}{\tau_S} \frac{x_5}{x_6} - \frac{1}{\tau_S} \right) x_6 + \frac{1}{\tau_S} u \quad (6.3.1)$$

and used Eqs.(6.3.1), (4.3.12) and (4.3.13) as the new state equations. Thus the order of the system is reduced to three while it was eight before.

This reduced order system is quite useful. The performance obtained using the reduced order system does not differ much from that obtained using the eighth order system. However, the computation time varies drastically. For example, in a specific scenario the computation time using the eighth order system is about 30 CPU seconds while for the third order system it is only 10 CPU seconds (these computation times are on CDC Cyber 170 Model 815 system).

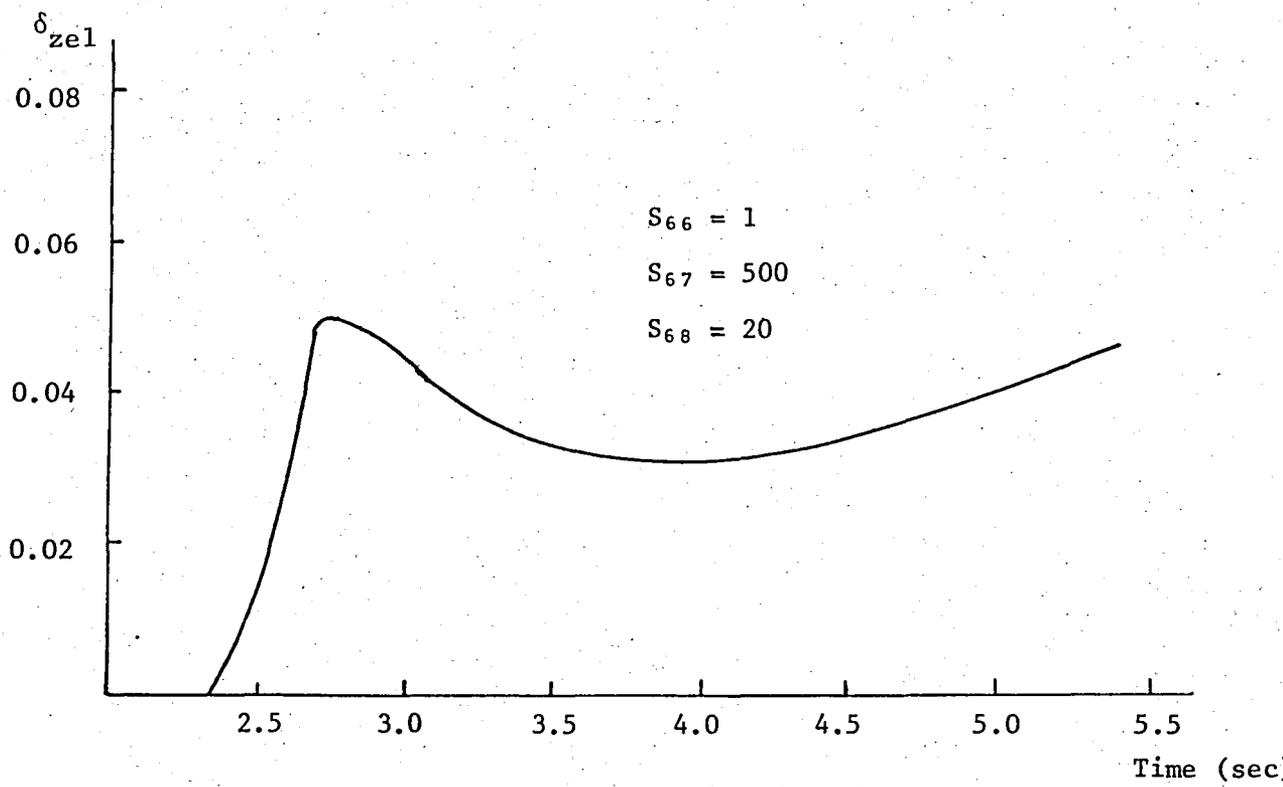


Figure 6.3.3 Control variation with time

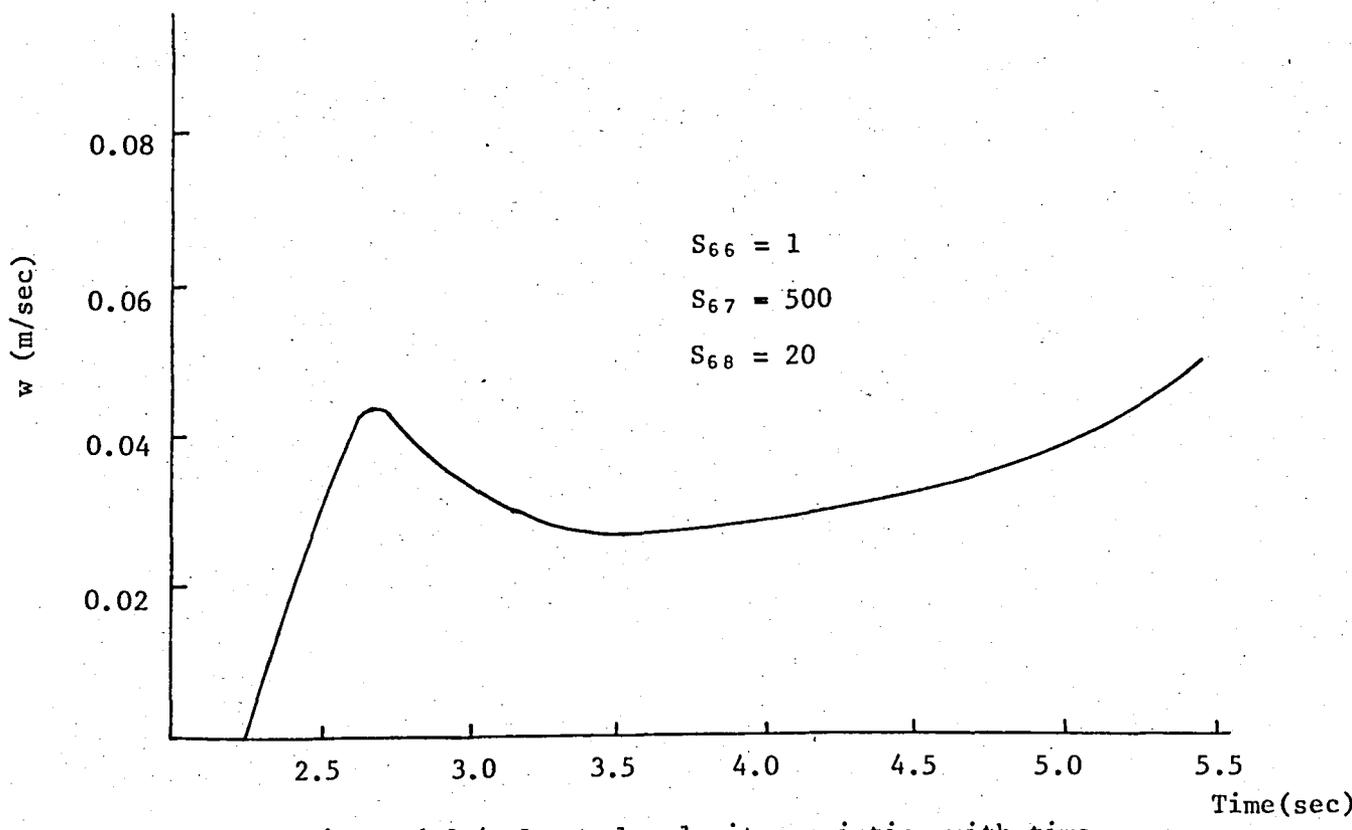


Figure 6.3.4 Lateral velocity variation with time

Another observation is that the performance does not improve much if the correction time for the system matrices is reduced below 0.1 seconds. For instance, correction times of 0.1, 0.05 and 0.01 seconds were used in a specific scenario and it was seen that the final performance, in terms of miss distance and average angle of attack, was about the same (although the instantaneous value of the control changed quite much between different correction times).

Also, suboptimal control is sensitive to initial conditions. The effect of launch angle (among other initial conditions) turned out to be quite important. Its effect was investigated for the scenario $H\phi = 10$ km, $R\phi = 2500$ m, $SIGMA = 90^\circ$, $MM = 1.2$, $MT = 0.95$, $GT = 3$, $PNG = 5$, $ATANGE = 10$, $DELTAT = 2.2$ s, $DELTAR = 2.2$ s and $AASC = 0$. For a launch angle $\theta_0 = 65.2$ the final performance turned out to be

$$RMIN = 5.134 \text{ m}$$

$$ALPMAX = 4.005^\circ$$

$$ALPAV = 1.335^\circ$$

For a launch angle $\theta_0 = 65^\circ$ same values turned out to be

$$RMIN = 1.852 \text{ m}$$

$$ALPMAX = 2.913^\circ$$

$$ALPAV = 0.694^\circ$$

The above values are obtained using the reduced order system with final state penalization matrix

$$\underline{S} = \begin{bmatrix} 1 & 500 & 20 \\ 500 & 1 & 0 \\ 20 & 0 & 1 \end{bmatrix} \quad (6.3.2)$$

From Table 6.2.2 we see that the best launch angle for this scenario is 65.75° for PNG control. However, for suboptimal control it is seen that performance varies drastically with launch angle and a new table of best launch angles together with final state penalization matrices S has to be formed for each scenario.

VII. CONCLUSION

In Chapter II we explain the task of a missile control system and briefly mention aerodynamic lateral control and thrust vector control. In the same chapter we introduce notation and conventions standardised in the guided missile literature.

Later we introduce the differential equations of motion for missile flight dynamics, the measurement and control system and the intercept error. The intercept equations are put into a form of system state equations suitable to application of optimal control.

In this work two different guidance techniques are used: proportional navigation guidance and suboptimal control. Each technique is briefly explained in Chapter IV.

Computer simulations are repeated for various scenarios differing in both initial conditions and target escape policies. Several problems are associated with the implementation of suboptimal guidance. Suboptimal guidance appears to be sensitive to initial conditions. Allied to this is the importance of selecting correct numerical quantities for the elements of the weighing matrices in the chosen performance index and the requirement to model the system accurately.

The intercept equations have to be scaled and linearized before suboptimal control can be applied. A reduced order system, including only the intercept error equations and the equation for the control system, is considered in evaluating the suboptimal control as well as

the full eighth order system. The results obtained using the third order system does not differ much from those obtained using the eighth order system, at least in terms of miss distance and average incidence. The computation time for the reduced order system is much shorter. A serious drawback of the third order system is that the resulting control does not vary in an orderly fashion and appears to be quite random. If this problem of control stability is overcome the third order system can be used.

Although modern tendency in missile guidance is towards implementation of optimal guidance; in this work it is seen that classical PNG law is quite successful in engagement against highly manoeuvring targets. An obvious advantage of PNG is simplicity of implementation.

Suboptimal control laws possess the capability of including other criteria in the specification of the cost functional in addition to miss distance in an engagement scenario.

B I B L I O G R A P H Y

1. Garnell, P. and D.J. East. Guided Weapon Control Systems. Oxford: Pergamon Press, 1977.
2. Pastrick, H.L., S.M. Seltzer, and M.E. Warren. "Guidance Laws for Short-Range Tactical Missiles," *Journal of Guidance and Control*, V.4, No.2, pp. 98-108, 1981.
3. Pontryagin, L.S., V. Boltyanskii, R. Gamkrelidze, and E. Mishchenko. The Mathematical Theory of Optimal Processes. New York: Interscience Publishers, Inc., 1962.
4. Athans, m. and P.L. Falb. Optimal Control. New York: Mc-Graw Hill, 1966.
5. Kalman, R.E., "Contribution to the Theory of Optimal Control," *Boletin Sociedad Matematica Mexicano*, V.5, pp.102-119, 1960.
6. Pearson, J.D., "Approximation Methods in Optimal Control," *Journal of Electronics and Control*, V.13, pp.453, 1962.
7. Weber, A.P.J. and L. Lapidus, "Suboptimal Control of Nonlinear Systems," *AIChE. Journal*, V.17, No.3, pp. 641-648, 1971.
8. Davison, E.J. and M.C. Maki. "The Numerical Solution of the Matrix Riccati Differential Equation," *IEEE Transactions on Automatic Control*, V.AC-18, pp. 71-73, 1973.
9. Kuzucu, A. and A. Roth, "Suboptimal Adaptive Control of Nonlinear Systems," *Lecture Notes in Control and Information Sciences* V.24, pp. 131-140, 1980.

10. Varga, R.S. Matrix Iterative Analysis. New Jersey: Prentice Hall, Inc., 1962.
11. Van Lvan, O.F., "Computing Integrals Involving the Matrix Exponential," *IEEE Transactions on Automatic Control*, V.AC-23, pp. 395-404, 1978.
12. Rawling, A.G., "Passive Determination of Homing Time," *AIAA Journal*, V.6, No.8, pp. 1604-1606, 1968.

APPENDIX

SIMULATION PROGRAM LISTING

PROGRAM MISSIL4 (OUTPUT, OUTM, INPUT, MISDAT, TAPE1=OUTPUT,
 TAPE2=OUTM, TAPE5=INPUT, TAPE7=MISDAT)
 PARAMETER (N=8, IR=1)

EXTERNAL MISNL

REAL LR, MO, MB, IYO, IYB, KQG, MMIN, MMAX
 REAL MM
 REAL MOD, IYD, M, IY
 REAL MT
 CHARACTER*1 ANSWER

DIMENSION A(N,N), B(N), Q(N,N), RI(IR, IR), S(N,N), GKQ(IR, N),
 1XDM(11), XM(11)
 COMMON /CONST/PI, G, SR, LR, MO, MB, XCGO, XCSB, IYO, IYB, TVENT, TB, TAUS,
 &TAUG, KQG, DZLIM, AE, AB, FINTIM, MMIN, MMAX
 COMMON /COEF/FBASE, FFRIC, FCNF, FCHF, FCMQ, FCFE, FCNA, FXCP, FS, FR, FA,
 COMMON /VAR/X(11), DX(11), M, XCG, IY
 COMMON /MISS/QTILD, CZ, CX, CA, AL, ALPHA, MM, HDZ, SCALE, DZEL, VIM
 COMMON /TARG/VIT, HS, ATANGE, DELTAT, DELTAR, AT, R
 COMMON /MISC/TT(34), FTHR(34), FSTAT(16), FRC(16), FAA(16),
 &FX(20, 20), FXXT(20, 20), FTT(16), ZFACF(3), ZFCNA(7), ZFXCP(5)

DATA XDM/30., 230., .1., .08., .01, 100., .1, 500., 500., 1., 1./
 DATA XM/50., 520., .1, 1., .002, 1., .3, 5000., 5000., 1., 1./
 DATA ZFACF/0., 10., 25., /ZFCNA/0., 2., 5., 10., 15., 20., 25./
 DATA ZFXCP/0., 2.5, 10., 15., 25./
 DATA X, DX/22*0./
 DATA A, Q/128*0., /S/64*0./
 DATA RI, GKQ, B/17*0./

MISCONST DATA

SR : REFERENCE AREA
 LR : REFERENCE LENGTH
 MO : INITIAL MASS
 MB : BURNOUT MASS
 XCGO : INITIAL CENTER OF GRAVITY LOCATION
 XCSB : CENTER OF GRAVITY LOCATION AT BURNOUT
 IYO : INITIAL MOMENT OF INERTIA
 IYB : MOMENT OF INERTIA AT BURNOUT
 TVENT : GUIDANCE ACTIVATION TIME
 TB : THRUST DURATION
 TAUS : SERVO TIME CONSTANT
 TAUG : RATE GYRO TIME CONSTANT
 KQG : RATE GYRO COEFFICIENT
 DZLIM : MAXIMUM FLIPPER DEFLECTION
 AE : NOZZLE EXIT AREA
 AB : BASE AREA
 FINTIM : POWER LIMIT (ELECTRICAL POWER SUPPLY DURATION)
 MMIN : MINIMUM MACH NUMBER
 MMAX : MAXIMUM MACH NUMBER

READ(7, 1000)PI, G, SR, LR, MO, MB, XCGO, XCSB, IYO, IYB, TVENT, TB, TAUS,
 &TAUG, KQG, DZLIM, AE, AB, FINTIM, MMIN, MMAX

FTHR : ROCKET MOTOR THRUST (KP) VS. TIME (S)

READ(7,1100)(TT(I),FTHR(I),I=1,34)

ATMOSPHERIC DATA

FSTAT : STATIC PRESSURE
(KP/M**2) VS. ALTITUDE (KM)
FR0 : ICAD STANDARD ATMOSPHERE AIR
DENSITY (KP/M**4*G**2 VS. ALTITUDE (KM)
FAA : IDEM, SPEED OF SOUND (M/S) VS. ALTITUDE (KM)

READ(7,1110)(FSTAT(I),FR0(I),FAA(I),FTT(I),I=1,16)

COEFDEF DATA

FBASE : BASE DRAG COEFFICIENT VS. MACH NUMBER
FFRIC : FRICTION DRAG COEFFICIENT VS. MACH NUMBER
FCNF : TRIM NORMAL FORCE EFFECTIVENESS (PER DEG)
VS. MACH NUMBER
FCMF : TRIM PITCHING MOMENT EFFECTIVENESS (PER DEG)
VS. MACH NUMBER
FCMQ : PITCH DAMPING MOMENT COEFFICIENT (PER RAD)
VS. MACH NUMBER
FACF : AXIAL FORCE COEFFICIENT VS. MACH NUMBER
AND INCIDENCE (DEG)
FCNA : NORMAL FORCE SLOPE COEFFICIENT (PER DEG)
VS. MACH NUMBER AND INCIDENCE (DEG)
FXCP : CENTER OF PRESSURE LOCATION (M) FROM NOSE
VS. MACH NUMBER AND ANGLE OF ATTACK (DEG)

DO 10 I=1,20

READ(7,1120)(FXX(I,J),FXXT(I,J),J=1,20)

CONTINUE

CHECK WHETHER SUBOPTIMAL CONTROL IS SOUGHT.

PRINT*, 'DO YOU WANT SUBOPTIMAL CONTROL ? (Y/N)'

READ(5,11)ANSWER

FORMAT(A1)

IF(ANSWER.EQ.'Y')THEN

FORMING Q AND R MATRICES

RI(1,1)=1.

READ NON-ZERO TERMS OF S MATRIX AND ECHO PRINT S

PRINT*, 'ENTER S(1,1)'

READ(5,*)S(1,1)

PRINT*, 'ENTER S(3,3)'

READ(5,*)S(3,3)

PRINT*, 'ENTER S(5,5)'

READ(5,*)S(5,5)

PRINT*, 'ENTER S(6,6)'

READ(5,*)S(6,6)

PRINT*, 'ENTER S(7,7)'

READ(5,*)S(7,7)

PRINT*, 'ENTER S(8,8)'

READ(5,*)S(8,8)

PRINT*, 'ENTER S(7,8) AND S(8,7)'

READ(5,*)S78

S(8,7)=S78

S(7,8)=S87

PRINT*, 'ENTER S(6,7) AND S(7,6)'

READ(5,*)S67

304

```

S(6,7)=SM67
S(7,6)=SM67
PRINT*, 'ENTER S(6,8) AND S(8,6)'
READ(5,*)SM68
S(6,8)=SM68
S(8,6)=SM68
WRITE(2,304)S
FORMAT(//, ' ***** S MATRIX ***** ', //, 8(1X, G9.3))
ENDIF
PRINT*, 'ENTER THE SIMULATION VARIABLES'
PRINT*, ' HU (KM) : ', ' RO (M) : '
READ(5,*)H,RO
PRINT*, ' SIGMA (DEG) : '
READ(5,*)SIGMA
PRINT*, ' MM : MT : '
READ(5,*)MM,MT
PRINT*, ' GT ; PNGO : '
READ(5,*)GT,PNGO
PRINT*, 'TANGENTIAL ACC : '
READ(5,*)ATANGE
PRINT*, 'DELTAT : DELTAR : '
READ(5,*)DELTAT,DELTAR
PRINT*, 'AASC : ANGLE OF ATTACK STABILIZATION COEFFICIENT'
READ(5,*)AASC
PRINT*, 'LAUNCH ANGLE (DEG) : '
READ(5,*)X(4)

```

```

INITIALIZE THE NECESSARY VARIABLES

```

```

SIG=SIGMA*PI/180.
X(8)= R*COS(SIG)
X(9)= R*SIN(SIG)

```

```

X(4)=X(4)*PI/180.
PNG=PNGO

```

```

R=RO
RMIN=100000
TIM=0.
ALPHA=0.
SIGDOT=0.
ALPDOT=0.
DZEL=0.
DZ=0.
RDOT=0.
RDOTDE=0.
ALPMAX=0.
SUMALP=0.
DZELOP=0.
CALL MISCOEF(H,MM,MT,ALPHA,1ERR)
VIM=MM*FA
X(1)=VIM*SIN(ALPHA)
X(2)=VIM*COS(ALPHA)
VIT=MT*FA

```

```

PRINT INITIAL CONDITIONS AND TARGET ESCAPE POLICY

```

```

WRITE(2,2010)H,R,SIGMA,MM,MT,GT,PNG,ATANGE,DELTAT,DELTAR,CVRC
1,AASC,(X(I),I=1,11)

```

```

TIMM=0.01
TIMT=0.
HS=0.01
HSX=HS
LMAX=FITIM/HS+100

```

START SIMULATION

DO 500 L=1,LMAX

EVALUATE THE CONTROL

IF(TIM.GT.2.2)THEN

RDOT=(X(3)*DX(3)+X(7)*DX(7))/R

SIGDOT=(X(3)*DX(9)-X(7)*DX(2))/R**2

ALPDOT=(X(2)*DX(1)-X(1)*DX(2))/VIT**2

SUBOPTIMAL CONTROL

IF(ANSWER.EQ.'Y')THEN

IF(TIMT.GE.HS*9.)THEN

IF(RDOT.NE.0.)TTOGO=ABS(R/RDOT)

IF(TTOGO.LE.0.05)GOTO 152

CALL MATRIX(A,R,N,TIM)

CALL GAIN1(A,B,Q,RI,S,O.,TTOGO,N,IR,GKD)

DZELOP=0.

DO 131 IKK=1,6

DZELOP=DZELOP+GKD(1,IKK)*X(1KK)/XM(1KK)

131 CONTINUE

DZELOP=DZELOP+GKU(1,7)*X(3)/XM(8)+GKU(1,8)*X(9)/XM(9)

WRITE(2,141)DZELOP,GKD

141 FORMAT(/,'DZELOP = ',G13.6,/, 'GKU = ',8(1X,G13.6))

152 DZEL=DZELOP

ELSE

IF(TIM.LE.2.303)DZEL=PNG*SIGDOT-AASC*ALPDOT

ENDIF

CLASSICAL PNG LAW

ELSE

DZEL=PNG*SIGDOT-AASC*ALPDOT

ENDIF

ENDIF

HSX=HS

TIM=TIMM-HS

CALL MISVAR(TIM)

CALL MISCOEF(H,MM,TIM,ALPHA,IERR)

SUM ALPHA TO EVALUATE ITS AVERAGE

IF(TIM.GT.2.2)THEN

SUMALP=SUMALP+ABS(ALPHA)

IF(ALPMAX.LT.ABS(ALPHA))ALPMAX=ABS(ALPHA)

ENDIF

TIN=TIMM-HS

CALL RUNGEK(MISNL,X,DX,11,TIM,HS,1)

DECREASE TIME STEP WHEN RANGE BECOMES SMALL ENOUGH

IF(R.LT.50.)HS=0.005

IF(R.LT.5.)HS=0.001

TIMM=TIMM+HS

TIMT=TIMT+HS

SET DZ TO ITS MAX OR MIN VALUE IF NECESSARY

IF(X(6).GT.0.4537856)X(6)=0.4537856

IF(X(6).LT.-.4537856)X(6)=-.4537856

IF(R.LT.RMIN)RMIN=R

PRINT THE STATE VARIABLES AND THE CONTROL EVERY TEN TIME STEPS

```

IF(TIME.GE.10.*HS)THEN
TIMT=0.
WRITE(2,1010)TIM,(X(I),I=1,11),DZEL
WRITE(2,121)R
121 FORMAT('R = ',F9.2)
ENDIF
DZ=X(6)*180./PI
IF (DZ.GT.DZLIM) DZ=DZLIM
IF (DZ.LT.-DZLIM) DZ=-DZLIM

CHECK FOR STOPPING CRITERIA

CALL MISFIN(IERR)
IF(TIM.GT.35.OR.IERR.EQ.4)GOTO 600
IF(IERR.EQ.0) GOTO 500
WRITE(1,2020)IERR,TIM,L
500 CONTINUE
600 CONTINUE
ALPAV=SUMALP/(L-220.)

PRINT SIMULATION RESULTS

WRITE(1,2030)L,TIM,RMIN,ALPMAV,ALPAV,(X(I),I=1,11)
WRITE(2,2030)L,TIM,RMIN,ALPMAV,ALPAV,(X(I),I=1,11)

FORMATS FOR DATA
1000 FORMAT(21(F12.6,/,))
1010 FORMAT(7(F11.4,1X),/,6(F11.4,1X))
1100 FORMAT(2(F12.6))
1110 FORMAT(3F12.6,F3.0)
1120 FORMAT(5(F6.4,F6.2))

FORMATS FOR OTHER I/O S
2010 FORMAT(///,15X,'PROGRAM MISSILE4',/,15X,'*****',///,
15X,'INITIAL CONDITIONS : ',/,15X,'HO : ',F6.2,' KM',
2/,15X,'RO : ',F9.3,' M',/,11X,'SIGMAO : ',F6.2,' DEG',/,
315X,'ML : ',F6.2/,15X,'MT : ',F6.2/,13X,'ATAN : ',F6.2/,
414X,'PNG : ',F6.2/,11X,'ATANGE : ',F6.2/,13X,'DELT : ',F6.2/,
513X,'DELR : ',F6.2/,13X,'CVRC : ',F6.0/,13X,'AASC : ',F6.3/,
65X,'X(I) : ',6(1X,F9.4),/,5X,'X(I) : ',5(1X,F9.4),///)
2015 FORMAT(//,15X,'PROGRAM MISSILE4',/,15X,'*****',///,
15X,'INITIAL CONDITIONS : ',/,15X,'HO : ',F6.2,' KM',
2/,15X,'RO : ',F9.3,' M',/,11X,'SIGMAO : ',F6.2,' DEG',/,
315X,'ML : ',F6.2/,15X,'MT : ',F6.2/,13X,'ATAN : ',F6.2/,
414X,'PNG : ',F6.2/,11X,'ATANGE : ',F6.2/,13X,'DELT : ',F6.2/,
513X,'DELR : ',F6.2/,13X,'CVRC : ',F6.0/,13X,'AASC : ',F6.3)
2020 FORMAT(5X,'ERROR ',15,5X,'TIME = ',F6.2,5X,'PAS = ',15)
2030 FORMAT(///,5X,'PROGRAM TERMINATED AT PAS NO : ',15,///,5X,
5,' FINAL TIME ',F6.3,///,5X,'MISS DISTANCE : ',F9.3,' M',///,
35X,'ALPMAV ',F6.3,' ALPAV ',F8.3,/,
35X,'X(I) : ',6(1X,F9.4),///,5X,'X(I) : ',5(1X,F9.4),/)

STOP
END
SUBROUTINE MISNL(X,TIM,DX)

THE INTERCEPT EQUATIONS ARE EVALUATED AND THE NECESSARY COEFFICIEN
USED IN THE EQUATIONS ARE COMPUTED.

DIMENSION X(11),DX(11)

REAL LR,MO,MB,IYO,IYB,KQG,MMIN,MMAX
REAL MM
REAL MOD,IYD,M,IY

```

```

COMMON /CONST/PI,SR,LR,MD,MR,XCGO,XCGB,IYO,IYB,TVENT,TB,TAUS,
STAUG,KQG,DZLIM,AE,AB,FIHTIM,MMIN,MMAX
COMMON /COEF/FBASE,FFRIC,FCNF,FCMF,FCM2,FACF,FCNA,FXCP,FS,FR,FA
COMMON /VAR/XX(11),DXX(11),H,XCG,IY
COMMON /MISS/QTILD,CZ,CX,CM,AL,ALPHA,MM,H,DZ,SCALE,DZEL,VIM
COMMON /TARG/VIT,HS,ATANG,DELTAT,DELTAR,AT,R
COMMON /MISC/TT(34),FTHM(34),FSTAT(16),FRO(16),FAA(16),
1FX(20,20),FXXT(20,20),FTT(16),ZFCF(3),ZFCNA(7),ZFXCP(5)

```

EVALUATE THE COEFFICIENTS

```

VIM=SQRT(X(1)**2+X(2)**2)
AL=ATAN(X(1)/X(2))
ALPHA=AL*180./PI

```

```
MM=VIM/FA
```

```

QTILD=FR/2.*VIM**2*SR
CH=FCNA*ALPHA-FCNF*DZ
AESAB=AE/AB
IF (TIM.GT.TB) AESAB=0.
CC=FACF+FFRIC*(H-3.096)/3.048+FBASE*(1-AESAB)
CM=FCNA*ALPHA*(XCG-FXCP)/LR+(FCMF-FCNF*(XCG-XCGB)/LR)*DZ

```

```
CX=-CC
```

```
CZ=-CH
```

```
IF (TIM.LT.TB) DZEL=0.
```

```
IF (TIM.GE.DELTAT) VIT=VIT+ATANG*HS/4.
```

MISSILE FLIGHT DYNAMICS

```

DX(1)=X(3)*X(2)+QTILD/M*CZ
DX(2)=-1*X(3)*X(1)+QTILD*CX/M+T/M
DX(3)=QTILD*LR/IY*(CH-LR/VIM*500*FCM2*X(3))
DX(4)=X(3)

```

MEASUREMENT AND CONTROL SYSTEM

```

DX(5)=-1./TAUG*X(5)+KQG*PI/180.*DX(3)
DX(6)=-1./TAUS*(X(6)+X(5)-DZEL)

```

INTERCEPT ERROR

```

DX(8)=VIT*COS(X(7))-VIM*COS(X(4)-AL)
DX(9)=VIT*SIN(X(7))-VIM*SIN(X(4)-AL)
IF (TIM.GE.DELTAR) DX(7)=AT/VIT

```

TRAJECTORY EQUATIONS

```

DX(10)=VIM*COS(X(4)-AL)
DX(11)=VIM*SIN(X(4)-AL)
RETURN
END

```

SUBROUTINE MISVAR(TIM)

CG, MASS AND INERTIA VARIATION DURING BOOST

```

REAL LR,NO,MR,IYO,IYB,KQG,MMIN,MMAX
REAL MM
REAL MD,MYD,M,IY
REAL MT

```

```

COMMON /CONST/PI,S,SR,LR,MO,MB,XCGO,XCGB,IYO,IYB,TVENT,TB,TAUS,
1TAUG,KQG,DZLIN,AE,AB,FINTIN,MMIN,MMAX
COMMON /COEF/EBASE,FFRIC,FCNF,FCMF,FCMQ,FACF,FCNA,FXCP,FS,FR,FA,
COMMON /VAR/X(11),DX(11),H,XCG,IY
COMMON /MISS/QTILD,CZ,CX,CM,AL,ALPHA,MM,H,DZ,DZEL,VIM
COMMON /TARG/VIT,HS,ATANCE,DELTAT,DELTAR,ATR
COMMON /MISCO/TT(34),FTHR(34),FSTAT(16),FRO(16),FAA(16),
1FXX(20,20),FXXT(20,20),FTT(16)

```

```

C
C
TSWI=TIM
IF (TIM.GT.TB) TSWI=TB
MOD=(MO-MB)/TB
XCGP=(XCGO-XCGB)/TB
IYD=(IYO-IYB)/TB

```

```

C
M=(MO-MOD*TSWI)/S
XCG=XCGO-XCGP*TSWI
IY=(IYO-IYD*TSWI)/S

```

```

C
RETURN
END
SUBROUTINE MISCOEF(H,MM,TIM,ALPHA,IERR)

```

```

C
C
C
IN THIS ROUTINE NUMEROUS CHECKS ARE PERFORMED AND THE VALUES
OF TABULATED VARIABLES ARE EVALUATED USING INTERPOLATION.

```

```

C
REAL LR,MO,MB,IYO,IYB,KQS,MMIN,MMAX
REAL MM
REAL MOD,IYD,H,IY
REAL HT

```

```

C
COMMON /CONST/PI,S,SR,LR,MO,MB,XCGO,XCGB,IYO,IYB,TVENT,TB,TAUS,
1TAUG,KQG,DZLIN,AE,AB,FINTIN,MMIN,MMAX
COMMON /COEF/EBASE,FFRIC,FCNF,FCMF,FCMQ,FACF,FCNA,FXCP,FS,FR,FA,
COMMON /VAR/X(11),DX(11),H,XCG,IY
COMMON /MISS/QTILD,CZ,CX,CM,AL,ALPHA,MM,H,DZ,DZEL,VIM
COMMON /TARG/VIT,HS,ATANCE,DELTAT,DELTAR,ATR
COMMON /MISCO/TT(34),FTHR(34),FSTAT(16),FRO(16),FAA(16),
1FXX(20,20),FXXT(20,20),FTT(16)

```

```

C
IERR=0

```

```

C
IF (TIM.GE.0.) GOTO 50
IERR=1
TIM=0.
WRITE(1,3000) IERR,TIM
50 CALL INTPOL(T,FTHR,TIM,TT,34)

```

```

C
IF (H.GE.0.) GOTO 250
IERR=2
H=0.
WRITE(1,3000) IERR,TIM
250 IF (H.LT.15) GOTO 260
IERR=3

```

```

H=14.999
WRITE(1,3000) IERR,TIM
260 CALL INTPOL(FS,FSTAT,H,FTT,16)
CALL INTPOL(FR,FRO,H,FTT,16)
CALL INTPOL(FA,FAA,H,FTT,16)

```

```

C
IF (MM.GE.MMIN) GOTO 310
IERR=4
MM=MMIN
WRITE(1,3000) IERR,TIM
310 IF (MM.LE.MMAX) GOTO 320

```

```

IERR=5
MM=MMAX
WRITE(1,3000) IERR,TIM
320 IF (ALPHA.GE.-25.) GOTO 330
IERR=6
ALPHA=-25.
WRITE(1,3000) IERR,TIM
350 IF (ALPHA.LE.25.) GOTO 340
IERR=7
ALPHA=25.
WRITE(1,3000) IERR,TIM
340 CONTINUE

```

```

C
A,3ALPHA=ABS(ALPHA)
CALL INTER1(FBASE,FXX,MM,FXXT,1,17)
CALL INTER1(FFRIC,FXX,MM,FXXT,2,10)
CALL INTER1(FCNF,FXX,MM,FXXT,3,9)
CALL INTER1(FCMF,FXX,MM,FXXT,4,9)
CALL INTER1(FCMQ,FXX,MM,FXXT,5,10)
CALL INTER(FXX,FXXT,2,FACT,12,6,8,3,FCNF,MM,ABALPHA)
CALL INTER(FXX,FXXT,2,FCNA,12,9,15,7,FCNA,MM,ABALPHA)
CALL INTER(FXX,FXXT,2,FXCP,12,16,20,5,FXCP,MM,ABALPHA)
600 CONTINUE

```

```

C
3000 FORMAT (5X,'ERROR IN MISCOEF = ',I3,5X,'TIME = ',F8.3)
C
RETURN
END
SUBROUTINE MISFIN(IERR)

```

IN THIS ROUTINE VARIOUS CHECKS ARE PERFORMED.

```

REAL LR,MO,MB,IYO,IYB,KQG,MMIN,MMAX
REAL MM
REAL MOD,IYD,M,IY
REAL MT

```

```

COMMON /CONST/PI,S,SR,LR,MO,MB,XCGO,XCGB,IYO,IYB,TVENT,TB,TAUS,
1TAUG,KQG,DZLIM,AE,AB,FINTIN,MMIN,MMAX
COMMON /COEF/FBASE,FFRIC,FCNF,FCMF,FCMQ,FACF,FCNA,FXCP,FS,FR,FA
COMMON /VAR/X(11),DX(11),Y,XCG,IY
COMMON /MISS/OTILD,CZ,CX,CM,AL,ALPHA,MM,H,DZ,DZEL,VIM
COMMON /TARG/VIT,HS,ATANGE,DELTAT,DELTAR,AT,R
COMMON /MISCO/TT(34),FTHR(34),FSTAT(16),FRO(16),FAA(16),
&FXX(20,20),FXXT(20,20),FTT(16)

```

```

IERR=0
IS STATIC MARGIN POSITIVE ?
SSM=(FXCP-XCG)/LR
IF (SSM.LT.0.) IERR=1
IS SPEED OF THE MISSILE GREATER THAN THAT OF THE TARGET ?
IF (VIM.LE.VIT) IERR=2
IS MISSILE MACH NUMBER GREATER THAN 0.3 ?
IF (MM.LT.MMIN) IERR=3
IS RDOT < 0. ?

```

```

RR=SQRT(X(8)**2+X(9)**2)
IF ((R-RR).LT.0.) IERR=4
R=RR

```

```

C      RETURN
C      END
C      SUBROUTINE INTER(X,Y,Z,NX,NY1,NY2,NZ,X1,Y1,Z1)
C
C      X   : DEPENDENT VARIABLE
C      Y   : FIRST INDEPENDENT VARIABLE
C      Z   : SECOND INDEPENDENT VARIABLE
C      NX  : UPPER BOUND OF THE SECOND INDEX OF THE ARRAY
C      NY1 : LOWER BOUND OF THE FIRST INDEX OF THE ARRAY
C      NY2 : UPPER BOUND OF THE FIRST INDEX OF THE ARRAY
C      NZ  : NY2-NY1+1
C      X1  : VALUE OBTAINED BY INTERPOLATION X(Y1,Z1)
C      Y1  : VALUE X1 IS TO BE EVALUATED FOR
C      Z1  : VALUE X1 IS TO BE EVALUATED FOR
C
C      DIMENSION X(20,20),Y(20,20),Z(7)
C      NXX=NX-1
C      NZZ=NZ-1
C      DO 10 I=1,NZZ
C      IF(Z1.GE.Z(I).AND.Z1.LE.Z(I))GOTO 20
10    CONTINUE
C      PRINT*, 'ERROR IN INTERPOLATION ! Z VALUE NOT IN RANGE !'
20    NYY1=NY1+I-1
C      NYY2=NY1+I
C      DO 30 J=1,NXX
C      IF(Y1.GE.Y(NYY1,J).AND.Y1.LE.Y(NYY1,J+1))GOTO 40
30    CONTINUE
C      PRINT*, 'ERROR IN INTERPOLATION ! Y VALUE NOT IN RANGE !'
40    CONTINUE
C      DO 50 K=1,NXX
C      IF(Y1.GE.Y(NYY2,K).AND.Y1.LE.Y(NYY2,K+1))GOTO 60
50    CONTINUE
C      PRINT*, 'ERROR IN INTERPOLATION ! Y VALUE NOT IN RANGE (K+1) !'
60    X11=X(NYY1,J)-(X(NYY1,J)-X(NYY1,J+1))*(Y(NYY1,J)-Y1)/(Y(NYY1,J)
C      -Y(NYY1,J+1))
C      X12=X(NYY2,K)-(X(NYY2,K)-X(NYY2,K+1))*(Y(NYY2,K)-Y1)/(Y(NYY2,K)
C      -Y(NYY2,K+1))
C      X1=X11-(X11-X12)*(Z(I)-Z1)/(Z(I)-Z(I+1))
C      RETURN
C      END
C      SUBROUTINE INTER1(X0,X,Y0,Y,N,M)
C
C      X0 : VALUE OBTAINED BY INTERPOLATION X(Y0)
C      X  : DEPENDENT VARIABLE X=X(Y)
C      Y0 : VALUE FOR WHICH X IS TO BE EVALUATED
C      Y  : INDEPENDENT VARIABLE
C      N  : INDICATES ROW OF THE TWO DIMENSIONAL ARRAY
C      M  : UPPER BOUND OF THE SECOND INDEX OF THE ARRAY
C           I.E. X(N,1:M)
C
C      DIMENSION X(20,20),Y(20,20)
C      X0=0.
C      M4=M-1
C      DO 100 I=1,MM
C      IF(Y0.GE.Y(N,I).AND.Y0.LE.Y(N,I+1))GO TO 200
100   CONTINUE
C      PRINT*, 'ERROR IN INTERPOLATION ! Y0 =',Y0,' N =',N,' M =',M
C      X0=0.
C      RETURN
200   X0=X(N,I)+(X(N,I+1)-X(N,I))*(Y0-Y(N,I))/(Y(N,I+1)-Y(N,I))
C      RETURN
C      END
C      SUBROUTINE INTPOL(X0,X,Y0,Y,M)
C
C      X0 : VALUE OBTAINED BY INTERPOLATION X(Y0)

```

X : DEPENDENT VARIABLE X=X(Y)
 Y0 : VALUE FOR WHICH X IS TO BE EVALUATED
 Y : INDEPENDENT VARIABLE
 M : UPPER BOUND OF THE ARRAY

```

DIMENSION X(40),Y(40)
XD=0.
MM=M-1
DO 100 I=1,MM
IF(Y0.GE.Y(I).AND.Y0.LE.Y(I+1))GO TO 200
100 CONTINUE
PRINT*, 'ERROR IN INTERPOLATION ! Y0 =',Y0,' M =',M
XD=0.
RETURN
200 XD=X(I)+(X(I+1)-X(I))*(Y0-Y(I))/(Y(I+1)-Y(I))
RETURN
END
SUBROUTINE MATRIX(A,B,N,TIM)

```

IN THIS ROUTINE THE 8*8 A MATRIX AND THE 8*1 B MATRIX USED IN STATE SPACE REPRESENTATION ARE FORMED USING APPARENT LINEARIZATION THE MATRICES ARE SCALED ALSO.

```

DIMENSION A(8,8),B(8),D(8),F(8),XM(8),XDM(8)
REAL LR,MO,MB,IYO,IYB,KQG,MXIN,MMAX
REAL MM
REAL MOD,IYD,M,IY
REAL MT

```

```

COMMON /CONST/PI,G,SR,LR,MO,MB,XCGG,XCGB,IYO,IYB,TVENT,TB,TAUS,
1TAUG,KQG,DZLIM,AE,AB,FINTIM,MMIN,MMAX
COMMON /COEF/FBASE,FFRIC,FCNF,FCNF,FCM,FACE,FCNA,FXCP,FS,FR,FA,T
COMMON /VAR/X(11),DX(11),M,XCG,IY
COMMON /MISS/QTILD,CZ,CX,CM,AL,ALPHA,MM,H,DZ,SCALE,DZEL,VIM
COMMON /TARG/VIT,HS,ATANGE,DELTA,DELTA,R,AT,R
COMMON /MISCO/TT(34),FTHR(34),FSTAT(16),FRD(16),FAA(16),
1FXX(20,20),FXXT(20,20),FTT(16),ZFACE(3),ZFCNA(7),ZFXCP(5)
DATA XDM/30.,230.,.1,.03,.01,100.,500.,500./
DATA XM/50.,220.,.1,1.,.002,1.,5000.,5000./
A(1,2)=QTILD/M*CZ/X(2)
A(1,3)=X(2)
A(2,2)=(T+QTILD*CX)/M/X(2)
A(2,1)=-X(3)
A(3,2)=LR*QTILD*CM/IY/X(2)
A(3,3)=-QTILD*LR**2*500*FCM/IY/VIM
A(4,3)=1
A(5,2)=A(3,2)*KQG*PI/180.
A(5,3)=A(3,3)*KQG*PI/180.
A(5,5)=-1./TAUG
A(6,5)=-1./TAUS
A(6,6)=A(6,5)
B(6)=1./TAUS
A(7,7)=VIT*COS(X(7))/X(8)
A(7,7)=-VIM*COS(X(4)-AL)/X(8)+A(7,7)
A(7,7)=A(7,7)/2.
A(7,8)=A(7,7)*X(8)/X(9)
A(8,7)=-VIM*SIN(X(4)-AL)/X(8)
A(8,7)=VIT*SIN(X(7))/X(8)+A(8,7)
A(8,7)=A(8,7)/2.
A(8,8)=A(8,7)*X(8)/X(9)
DO 10 I=1,6
10 F(I)=X(I)
F(7)=X(8)
F(8)=X(9)

```

CALL GMPD(A,F,D,3,3,1)

D(6)=D(6)+DZEL/TAUS

DO 41 I=1,3

B(I)=B(I)/XDM(I)

DO 41 J=1,3

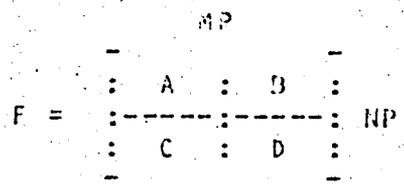
41 A(I,J)=A(I,J)*XM(J)/XDM(I)

RETURN

END

SUBROUTINE PART(F,N,M,NP,MP,A,B,C,D)

THIS PROGRAM PARTITIONS A (N*M) GENERAL MATRIX F TO FOUR SUBMATRICES A,B,C AND D DELIMITED AT THE NP' TH LINE AND THE MP' TH COLUMN, IN THE FOLLOWING WAY:



USAGE: CALL PART(F,N,M,NP,MP,A,B,C,D)

ARGUMENTS:

- F : N*M GENERAL INPUT MATRIX
- N : NUMBER OF ROWS OF F
- M : NUMBER OF COLUMNS OF F
- NP : ROW NUMBER AT WHICH F IS PARTITIONED
- MP : COLUMN NUMBER AT WHICH F IS PARTITIONED
- A : NP*MP SUBMATRIX OF F
- B : NP*(M-MP) SUBMATRIX OF F
- C : (N-NP)*MP SUBMATRIX OF F
- D : (N-NP)*(M-MP) SUBMATRIX OF F

ATTENTION: A,B,C AND D SHOULD BE DECLARED AS MATRICES WITH APPROPRIATE DIMENSIONS IN THE MAIN PROGRAM.

DIMENSION F(1),A(1),B(1),C(1),D(1)

INITIALIZE INCREMENTERS

NN=N-NP
MM=M-MP
KF=-N
KA=-NP
KB=-NP
KC=-NN
KD=-NN

PARTITION THE FIRST MP COLUMNS TO A AND C

DO 3 J=1,MP

KF=KF+N
KA=KA+NP
KC=KC+NN
IK=KF
IA=KA
IC=KC

DO 1 I=1,MP
IK=IK+1
IA=IA+1

1 A(IA)=F(IK)

```

IK=IK+1
IC=IC+1
2 C(IC)=F(IK)
3 CONTINUE

```

PARTITION THE OTHER MM=M-NP COLUMNS INTO B AND D

```

DO 6 J=1,MM
KF=KF+M
KB=KB+NP
KD=KD+NH
IK=KF
IB=KB
ID=KD

DO 4 I=1,NP
IK=IK+1
IB=IB+1
4 B(IB)=F(IK)

DO 5 I=1,NN
IK=IK+1
ID=ID+1
5 D(ID)=F(IK)
6 CONTINUE

```

```

RETURN
END
SUBROUTINE KKH(A,N,T,K,H)
DIMENSION A(1)

```

COMPUTATION OF THE NORM OF (MATRIX A)*(TIME INTERVAL T)

```

N2=N*N
SN=0.
DO 1 I=1,N2
1 SN=SN+(A(I)*T)**2
ISN=SQRT(SN)*2

```

SELECTION OF THE STEP SIZE H AND THE NUMBER OF DOUBLING
ITERATIONS USING VAN LOAN'S CRITERION: $SSN/(2**KK) < 1/2$

```

K=0
H=T
2 K=K+1
H=H/2.
ISN=ISN/2
IF(ISN.GE.1)GO TO 2

```

```

RETURN
END
SUBROUTINE EXPF3(F,N,H,E1,E2,L,M,EFT)

```

THIS PROGRAM COMPUTES THE EXPONENTIAL OF A GENERAL SQUARE MATRIX
FOR A GIVEN TIME INTERVAL (STEP SIZE) H USING 3RD ORDER PADE
APPROXIMATION.

USAGE:

CALL EXPF3(F,N,H,E1,E2,L,M,EFT)
INPUT ARGUMENTS ARE:

F : GENERAL N*N SQUARE MATRIX
N : ORDER N OF THE MATRIX F
H : TIME INTERVAL (STEP SIZE)

OUTPUT IS:

EFT: N*N RESULT MATRIX EXPONENTIAL OF F*H

DIMENSIONS OF MATRICES AND VECTORS SHOULD BE PROPERLY DECLARED IN THE MAIN PROGRAM.

DIMENSION F(1),E1(1),E2(1),L(1),M(1),EFT(1)

MATRIX SIZE AND INTERMEDIATE TIME INTERVALS

N1=N+1
 N2=N*N
 H2=H/2.
 H10=H*H/10.
 H60=H*H/60.

SQUARING F

IJ=0
 IK=-N
 DO 2 K=1,N
 IK=IK+N
 DO 2 J=1,N
 IJ=IJ+1
 JI=J-N
 IB=IK
 SI=0.
 DO 1 I=1,N
 JI=JI+N
 IB=IB+1
 1 SI=SI+F(JI)*F(IB)
 2 EFT(IJ)=SI

CONSTRUCTION OF THE WORK MATRIX E1=I+(FH)**2/10.

DO 3 I=1,N2
 3 E1(I)=EFT(I)*H10
 IJ=-N
 DO 4 I=1,N
 IJ=IJ+N1
 4 E1(IJ)=E1(IJ)+1.

CONSTRUCTION OF THE WORK MATRIX E2=I+(FH)**2/60.

DO 5 I=1,N2
 5 E2(I)=EFT(I)*H60
 IJ=-N
 DO 6 I=1,N
 IJ=IJ+N1
 6 E2(IJ)=E2(IJ)+1

MULTIPLICATION BY (FH/2)

IJ=0
 IK=-N
 DO 8 K=1,N
 IK=IK+N
 DO 3 J=1,N
 IJ=IJ+1
 JI=J-N

```

SI=0.
DO 7 I=1,N
JI=JI+N
IB=IB+1
7 SI=SI+F(JI)*E2(IB)
8 EFT(IJ)=SI*H2

      [I+(FH)**2/10+(FH/2)(I+(FH)**2/60)]=E1+EFT PLACED IN E2

DO 9 I=1,N2
9 E2(I)=E1(I)+EFT(I)

      [I+(FH)**2/10-(FH/2)(I+(FH)**2/60)]=E1-EFT PLACED IN E1

DO 10 I=1,N2
10 E1(I)=E1(I)-EFT(I)

      E1 INVERTED

CALL MINV(E1,N,D,L,M)

      COMPUTATION OF EXPF(FH)=EFT=E1*E2

IJ=0
IK=-N
DO 12 K=1,N
IK=IK+N
DO 12 J=1,N
IJ=IJ+1
JI=J-N
IB=IK
SI=0.
DO 11 I=1,N
JI=JI+N
IB=IB+1
11 SI=SI+E1(JI)*E2(IB)
12 EFT(IJ)=SI

RETURN
END
SUBROUTINE CANM(A,B,Q,RI,N,IR,BT,RT,BRBT,F)

```

THIS SUBROUTINE FORMS THE CONTINUOUS HAMILTON MATRIX OF THE LINEAR QUADRATIC OPTIMAL CONTROL PROBLEM DEFINED FOR THE SYSTEM (A,B) WITH THE PENALIZATION MATRICES (Q,R) IN THE FORM:

$$F = \begin{bmatrix} A & -B*RI*BT \\ -Q & -AT \end{bmatrix}$$

INPUT ARGUMENTS ARE:

A : N*N SYSTEM MATRIX
 B : N*R INPUT MATRIX
 Q : N*N STATE PENALIZATION MATRIX
 RI : R*R INVERSE OF THE CONTROL PENALIZATION MATRIX
 N : ORDER N OF THE SYSTEM
 IR : ORDER R OF THE CONTROL VECTOR

OUTPUTS OF THE SUBROUTINE ARE:

BT : TRANSPOSE OF THE INPUT MATRIX (R*N)

RBT : -RI*BT (R*N)
BRBT : -B*RI*BT (N*N)

F : (2N)*(2N) HAMILTON MATRIX OF THE
OPTIMAL CONTROL PROBLEM

USAGE:

CALL CANM(A,B,Q,RI,N,IR,BT,RBT,BRBT,F)

SUBROUTINES USED: \$KUZUCU.PROGRAMS\$GMPRD

ATTENTION: DIMENSIONS OF MATRICES SHOULD BE PROPERLY DECLARED
IN THE MAIN PROGRAM.

DIMENSION A(1),B(1),Q(1),RI(1),BT(1),RBT(1),BRBT(1),F(1)

VECTOR SIZES

N2=2*N
N1=N+1

TRANSPOSE OF THE INPUT MATRIX B

```
K=0
DO 1 I=1,N
IJ=I-N
DO 1 J=1,IR
IJ=IJ+N
K=K+1
1 BT(K)=B(IJ)
```

COMPUTATION OF -RI*BT AFTER GMPRD

```
IJ=0
IK=-IR
DO 3 K=1,N
IK=IK+1R
DO 3 J=1,IR
IJ=IJ+1
JI=J-IR
IB=IK
S=0.
DO 2 I=1,IR
JI=JI+IR
IB=IB+1
2 S=S-RI(JI)*BT(IB)
3 RBT(IJ)=S
```

COMPUTATION OF BRBT = B*RBT

CALL GMPRD(B,RBT,BRBT,N,IR,N)

FORMATION OF THE MATRIX F USING THE ALGORITHM OF FORM

NNJ=-N2
NJ=-N

FIRST N COLUMNS

```
DO 5 J=1,N
NNJ=NNJ+N2
NJ=NJ+N
DO 4 I=1,N
IJJ=I+NNJ
IJ=I+NJ
4 F(IJJ)=A(IJ)
DO 5 I=N1,N2
```

IJ=I-N+NJ
5 F(IJJ)=-Q(IJ)

SECOND N COLUMNS

NJ=-N
DO 7 J=N1,N2

NNJ=NNJ+N2
NJ=NJ+N

DO 6 I=1,N
IJJ=I+NNJ
IJ=I+NJ
6 F(IJJ)=BRBT(IJ)

IJ=J-N2
DO 7 I=N1,N2
IJ=IJ+N
IJJ=I+NNJ
7 F(IJJ)=-A(IJ)

RETURN
END

SUBROUTINE GAIN1(A,B,Q,RI,S,AL,TF,N,IR,GKO)

THIS SUBROUTINE COMPUTES THE LINEAR OPTIMAL FEEDBACK GAINS
CORRESPONDING TO A GIVEN HORIZON TIME TF WITH A PRESCRIBED
DEGREE OF STABILITY OF AL.

INPUT ARGUMENTS ARE:

A : N*N SYSTEM MATRIX
B : N*R INPUT MATRIX
Q : N*N STATE WEIGHT MATRIX
RI : R*R INVERSE OF THE CONTROL WEIGHT MATRIX
S : N*N FINAL STATE WEIGHT MATRIX
AL : DESIRED DEGREE OF STABILITY
TF : HORIZON TIME
N : ORDER N OF THE SYSTEM
IR : ORDER R OF THE CONTROL VECTOR

THE OUTPUT OF THE SUBROUTINE IS:

GKO : R*N FEEDBACK MATRIX

USAGE:

CALL GAIN1(A,B,Q,RI,S,AL,TF,N,IR,GKO)

SUBROUTINES USED:

\$KUZUCU, PROGRAMS OGMPRD, MINV, CANM, KKH, EXPF3, PART

* THIS PROGRAM USES INTERMEDIATE MATRICES WHICH CAN BE *
* USEFUL FOR FURTHER COMPUTATIONS IN THE LINEAR SYSTEM *

* STUDY. DIMENSIONS OF THOSE MATRICES SHOULD BE GIVEN
* WITHIN THIS SUBROUTINE. THOSE MATRICES MAY FIGURE
* AMONG THE INPUT ARGUMENTS IF NECESSARY.
*
* EDIT THE NECESSARY MODIFICATIONS BEFORE USAGE!

DIMENSION A(1),B(1),G(1),RI(1),S(1),GKO(1)

INTERMEDIATE MATRICES AND DIMENSIONS FOR CANM

BT : TRANSPOSE OF B (R*N)
RBT : -RI*BT (R*N)
BRBT : -B*RI*BT (N*N)
F : HAMILTON MATRIX (2N*2N)

DIMENSION BT(1,8),RBT(1,8),BRBT(8,8),F(16,16)

INTERMEDIATE MATRICES AND DIMENSIONS FOR EXPF3

E1,E2 : WORK MATRICES (2N*2N) (FOR 2N*2N F MATRIX)
LW,MW : WORK VECTORS (2N) (FOR 2N*2N F MATRIX)
EF : EXPONENTIAL OF THE MATRIX F*H (2N*2N)

DIMENSION E1(16,16),E2(16,16),LW(16),MW(16),EF(16,16)

INTERMEDIATE MATRICES AND DIMENSIONS FOR THIS SUBROUTINE

F1 : TRANSITION MATRIX (2N*2N)
F11,F12,F21,F22 : WORK MATRICES FOR PART (N*N)
P : RICCATI MATRIX (N*N)

DIMENSION F1(16,16),F11(8,8),F12(8,8),F21(8,8),F22(8,8),P(8,8)

INTERMEDIATE MATRICES AND DIMENSIONS FOR MINV

DIMENSION L(8),M(8)

INCREMENTORS AND VECTOR SIZES

N1=N+1
N2=2*N
NH=N*N

INTRODUCTION OF THE PRESCRIBED DEGREE OF STABILITY

IF(AL.LE.0.) GO TO 3

IJ=-N
DO 1 I=1,N
IJ=IJ+N1
1 A(IJ)=A(IJ)+AL

ALTF=-AL*TF*2
ALTF=EXP(ALTF)
DO 2 I=1,NH
2 S(I)=S(I)*ALTF

3 CONTINUE

CONSTRUCTION OF THE HAMILTON MATRIX

```

C
C
C
C100 WRITE(6,100)
C   FORMAT(1X,'*** CANONICAL MATRIX F ***',/)
C   WRITE(6,101)((F(I,J),J=1,N2),I=1,N2)
C   *****
C101 FORMAT(1X,12F10.4)
C
C
C   SELECTION OF THE STEP SIZE AND THE DOUBLING NUMBER
C
C   CALL KKH(F,N2,TF,KK,H)
C
C   IMPRESSION OF THE STEP SIZE AND THE STEP SIZE
C
C   WRITE(6,102)TF,H,KK
C102 FORMAT(1X,' H =',F10.4,' TF=',F10.4,' KK=',I3)
C
C   EXPONENTIAL OF F*H
C
C   CALL EXPF3(F,N2,H,L1,E2,L4,M4,EF)
C
C   IMPRESSION OF THE INITIAL TRANSITION MATRIX
C
C   WRITE(6,103)
C103 FORMAT(1X,'*** TRANSITION MATRIX FOR H ***',/)
C   WRITE(6,101)((EF(I,J),J=1,N2),I=1,N2)
C
C   DOUBLING KK TIMES
C
C   DO 5 K=1,KK
C
C   CALL GMPRD(EF,EF,FI,N2,N2,N2)
C
C   IMPRESSION OF THE TRANSITION MATRIX
C
C   WRITE(6,104)K
C104 FORMAT(1X,'*** TRANSITION MATRIX AT DOUBLING KK=',I3)
C   WRITE(6,101)((F1(I,J),J=1,N2),I=1,N2)
C
C
C   IF(K.EQ.KK)GO TO 5
C
C   DO 4 J=1,N2
C   DO 4 I=1,N2
C   4 EF(I,J)=FI(I,J)
C
C   5 CONTINUE
C
C   PARTITION FI=EXP(F*TF)
C
C   CALL PART(FI,N2,N2,N,N,F11,F12,F21,F22)
C
C   COMPUTE F22=F22-S*F12
C
C   CALL GMPRD(S,F12,P,N,N,N)
C
C   DO 6 J=1,N
C   DO 6 I=1,N
C   6 F22(I,J)=F22(I,J)-P(I,J)
C
C   COMPUTE F21=-(F21-S*F11)=S*F11-F21
C
C   CALL GMPRD(S,F11,P,N,N,N)

```

```

DO 7 J=1,N
DO 7 I=1,N
7 F21(I,J)=P(I,J)-F22(1,J)

```

```

      INVERT F22

```

```

CALL MINV(F22,N,D,L,M)

```

```

      COMPUTE THE RICCATI MATRIX P(O)=F22*F21

```

```

CALL GMPRD(F22,F21,P,N,H,N)

```

```

      IMPRESSION OF THE RICCATI MATRIX P

```

```

WRITE(6,105)

```

```

105 FORMAT(1X,'*** THE RICCATI MATRIX P(O) ***')

```

```

WRITE(6,106)((P(I,J),J=1,H),I=1,N)

```

```

106 FORMAT(1X,6F10.4)

```

```

      COMPUTE THE FEEDBACK GAIN MATRIX G(O)=RBT*P(O)

```

```

CALL GMPRD(RBT,P,GK,IR,N,N)

```

```

RETURN

```

```

END

```

```

SUBROUTINE MINV

```

```

PURPOSE

```

```

      INVERT A MATRIX

```

```

USAGE

```

```

      CALL MINV(A,N,D,L,M)

```

```

DESCRIPTION OF PARAMETERS

```

```

      A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY
          RESULTANT INVERSE.

```

```

      N - ORDER OF MATRIX A

```

```

      D - RESULTANT DETERMINANT

```

```

      L - WORK VECTOR OF LENGTH N

```

```

      M - WORK VECTOR OF LENGTH N

```

```

REMARKS

```

```

      MATRIX A MUST BE A GENERAL MATRIX

```

```

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

```

```

      NONE

```

```

METHOD

```

```

      THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT
      IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT
      THE MATRIX IS SINGULAR.

```

```

SUBROUTINE MINV(A,N,D,L,M)

```

```

DIMENSION A(1),L(1),M(1)

```

```

IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE
C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION
WHICH FOLLOWS.

```

THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS ROUTINE.

THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS. ABS IN STATEMENT 10 MUST BE CHANGED TO DABS.

.....

SEARCH FOR LARGEST ELEMENT

```

D=1.0
NK=-N
DO 30 K=1,N
  NK=NK+N
  L(K)=K
  H(K)=K
  KK=NK+K
  BIGA=A(KK)
  DO 20 J=K,N
    IZ=N*(J-1)
    DO 20 I=K,N
      IJ=IZ+I
10  IF(ABS(BIGA)-ABS(A(IJ))) 15,20,20
15  BIGA=A(IJ)
    L(K)=I
    M(K)=J
20  CONTINUE

```

INTERCHANGE ROWS

```

J=L(K)
IF(J-K) 35,35,25
25  KI=K-N
  DO 30 I=1,N
    KI=KI+N
    HOLD=-A(KI)
    JI=KI-K+J
    A(KI)=A(JI)
30  A(JI)=HOLD

```

INTERCHANGE COLUMNS

```

35  I=M(K)
  IF(I-K) 45,45,35
38  JP=N*(I-1)
  DO 40 J=1,N
    JK=NK+J
    JI=JP+J
    HOLD=-A(JK)
    A(JK)=A(JI)
40  A(JI)=HOLD

```

DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS CONTAINED IN BIGA)

```

45  IF(BIGA) 48,46,48
46  D=D.0
  RETURN
48  DO 55 I=1,N
    IF(I-K) 50,55,50
50  IK=NK+I

```

55

C
C
C

REDUCE MATRIX

```
DO 65 I=1,N
IK=NK+I
HOLD=A(IK)
IJ=I-N
DO 65 J=1,N
IJ=IJ+N
IF(I-K) 60,65,60
60 IF(J-K) 62,65,62
62 KJ=IJ-I+K
A(IJ)=HOLD*A(KJ)+A(IJ)
65 CONTINUE
```

C
C
C

DIVIDE ROW BY PIVOT

```
KJ=K-N
DO 75 J=1,N
KJ=KJ+N
IF(J-K) 70,75,70
70 A(KJ)=A(KJ)/B1GA
75 CONTINUE
```

C
C
C

PRODUCT OF PIVOTS

D=D*B1GA

C
C
C

REPLACE PIVOT BY RECIPROCAL

```
A(KK)=1.0/B1GA
30 CONTINUE
```

C
C
C

FINAL ROW AND COLUMN EXCHANGE

```
K=N
100 K=(K-1)
IF(K) 150,150,105
105 I=L(K)
IF(I-K) 120,125,105
108 JQ=N*(K-1)
JR=N*(I-1)
DO 110 J=1,N
JK=JR+J
HOLD=A(JK)
JI=JR+J
110 A(JK)=-A(JI)
A(JI)=HOLD
120 J=M(K)
IF(J-K) 100,100,125
125 KI=K-N
DO 130 I=1,N
KI=KI+N
HOLD=A(KI)
JI=KI-K+J
A(KI)=-A(JI)
130 A(JI)=HOLD
GO TO 100
150 RETURN
END
```

C
C
C
C

SUBROUTINE GMPRO

PURPOSE

MULTIPLY TWO GENERAL MATRICES TO FORM A RESULTANT GENERAL MATRIX

USAGE

CALL GMPRD(A, B, R, N, M, L)

DESCRIPTION OF PARAMETERS

A - NAME OF FIRST INPUT MATRIX
 B - NAME OF SECOND INPUT MATRIX
 R - NAME OF OUTPUT MATRIX
 N - NUMBER OF ROWS IN A
 M - NUMBER OF COLUMNS IN A AND ROWS IN B
 L - NUMBER OF COLUMNS IN B

REMARKS

ALL MATRICES MUST BE STORED AS GENERAL MATRICES
 MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX A
 MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX B
 NUMBER OF COLUMNS OF MATRIX A MUST BE EQUAL TO NUMBER OF ROWS OF MATRIX B

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

NONE

METHOD

THE M BY L MATRIX IS PREMULIPLIED BY THE N BY M MATRIX A AND THE RESULT IS STORED IN THE N BY L MATRIX R

.....
 SUBROUTINE GMPRD(A, B, R, N, M, L)
 DIMENSION A(1), B(1), R(1)

```

IR=0
IK=-M
DO 10 K=1, L
IK=IK+M
DO 10 J=1, N
IR=IR+1
JI=J-N
IB=IK
R(IR)=0
DO 10 I=1, M
JI=JI+N
IB=IB+1
10 R(IR)=R(IR)+A(JI)*B(IB)
RETURN
END
```

6.12.53.UCLP, BU, PG3

1.473KLS.