**INTERNSHIP REPORT ON**

# KINEMATIC ANALYSIS OF SURFACE TO AIR MISSILE

**Submitted to**

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**
**BELGAUM**

**In partial fulfilment of the requirements for the award of degree of**

**BACHELOR OF ENGINEERING**

**In**

**AERONAUTICAL ENGINEERING**

**By**

| | |
|---|---|
| VAISHNAV R | 1NT18AE056 |
| PRAJWAL S RODRIGUES | 1NT18AE032 |
| RITHIK MARC ROBINSON | 1NT18AE042 |


**Internship Is Carried Out At**
**Aeronautical Development Establishment, DRDO, Bangalore**

**Submitted to:**
**Mr. Suraj Chaurasiya**
**Scientist, Aeronautical Development Establishment, DRDO, Bangalore**

**DEPARTMENT OF AERONAUTICAL ENGINEERING**
**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**
**YELAHANKA, BANGALORE-560064**

# INDEX

# ACKNOWLEDGEMENT

Behind every achievement there are people who inspire us to do it. To them we take the opportunity to lay the words of gratitude imprinted not just on the paper but deep in our heart.

I express my deepest thanks to **Dr Shashidhar** and **Mr. Suraj Chaurasiya , FSIM & AI, Aeronautical Development Establishment, DRDO, Bangalore** for allowing us to carry out Internship at your esteemed organisation and supporting us throughout.

I am grateful to **Dr. P K Dash,** HOD, Department of Aeronautical Engineering, NMIT, Bangalore for granting me the permission and guiding me throughout.

I wish to convey my deep sense of gratitude to **Dr. P N Tengli** , Prof & Advisor (Aero / Space Research),Aeronautical Department, NMIT for his valuable guidance for attaining this internship.

# ABSTRACT

This project will provide not only the foundational building blocks of Proportional Navigation but will then move upward from that foundation into an exploration of many of the typical trade-offs and modelling studies required during the execution of modern missile guidance programs.  This coverage of PN basics will provide the reader a general knowledge of concepts and specific examples that can readily be adapted to any number of applications and specific hardware designs.

A matlab$^{TM}$ model of lateral proportional navigation was implemented during the Analysis to demonstrate the impact of such factors as target manoeuvres, timing of manoeuvres, variance of the navigational constant by guidance phase, temporal gaps in the feedback of target information, energy usage, and terminal guidance logic.  Examples found throughout this text are designed to provide insight into the typical problems encountered by modern missile guidance applications.

# INTRODUCTION

As a brief introduction to proportional navigation guidance of modern missile systems let us first consider the familiar situation of a passenger arriving late for a train. That passenger is you, and as you approach the train is pulling away from the (now empty) station. There are four basic methods of reaching (guiding yourself to) the doorway of the train (Rocket Guidance, 2005):

- You could run straight toward the door at all times as it moves
- You could run at some constant angle in front of the door
- You could run in a fixed direction based on an initial line-of-sight and running speed
- You could lead the door by an angle proportional to the train speed and your running speed (this is the “Proportional” Navigation method of guidance control)

This report will focus on aspects of the Proportional Navigation (PN) method. In general, the term Proportional Navigation refers to a determination of your direction based on a comparison of two different angular rates of change. More specifically, PN means that the angular change in the (running passengers) path exceeds the angular change in the line-of-sight (to the door) by a multiplicative factor called the navigation constant (N`) or gain. At some future time, this will correct the runners’ path such that the line-of-sight angle is no longer changing and the runner is on a “collision course” for the door as desired. Without noise considerations, the higher the chosen navigation constant, the quicker the runner will correct his path and the less adjustment will be required later.

Although conceptually simple to understand, the power of this guidance control algorithm was recognized very early and is currently used in many variants over a diverse set of applications. The following section is a brief but critical background of Proportional Navigation.

# PROPORTIONAL NAVIGATION PRINCIPLES

## PN Law Introduction

The Law of Proportional Navigation provides lateral acceleration commands to the target line-of-sight. These commands are proportional (using a navigation constant) to the closing velocity and the rotation rate of the missile line-of-sight. In other words, the steering guidance of missiles using proportional navigation depends primarily on three multiplicative factors which when multiplied together need to yield the minimum possible value of lateral acceleration (latex). Here is a basic definition of the three terms and what can be done to minimize their multiplicative result.

The navigation constant (N`)

The navigation constant is a value that is chosen by the designers based on the specific application. As we will see this value may not actually be "constant" but may vary by phase (midcourse, terminal, etc.) or be designed as a function of one or more parametric (flight time, altitude, missile speed, etc.). The navigation constant is always 1 or greater. A value of 1, in the prior example, is when you are running straight at the door. The higher this multiplier is, the faster you correct your path and consequently the less correction is necessary later in the flight profile (barring target manoeuver). This desire for high gain, however, must be tempered based on the expected noise and other factors. Typical values as shall be determined later are between 2 and 4 for the various phases of guidance.

The closing velocity ($v_c$)

The closing velocity refers to the speed at which the target and missile are approaching each other. The closing velocity between a missile and target cannot be less than zero (never a good sign) or equal to zero if you want to intercept a target.

The rotation rate of the missile line-of-sight to the target ($d\lambda$)

An understanding of this parameter is **KEY** to the foundation of proportional navigation. The rotation rate of the missile line-of-sight to the target is the product factor that we want to make equal to zero. A value of zero will put the missile on a "collision course" with the target and need no further corrections (barring a target manoeuver). Maritime captains knew and understood the ramifications of this rule well. Another ship maintaining the SAME relative angle over time was a sure indication that they were on a COLLISION COURSE .The difficulty with this term arises from the number of factors that can cause this to deviate from zero once established. These factors include noise, acceleration of target and/or missile, manoeuvring of target and/or missile, jamming, and many other factors.

Using these basic parameters, the general law of Proportional Navigation (PN) can now be expressed as:

Equation 1: $n_c = N`v_c \, d\lambda$

## PN Law Derivation

Building from this foundation of "what" the PN law is, we can now expand our understanding to include the mathematical derivation. This derivation will facilitate a complete understanding of the subsequent concepts.

First I will list the basic terms to be used in the derivation. "M" and "T" subscripts will refer to M=missile and T=target parameters (see

Figure 1). The "0" and "F" subscripts will be used to refer to time, with "0" representing the initial condition and "F" representing the time of closest approach. Following the figure, each term used on the Figure and in the subsequent derivation will be briefly defined.

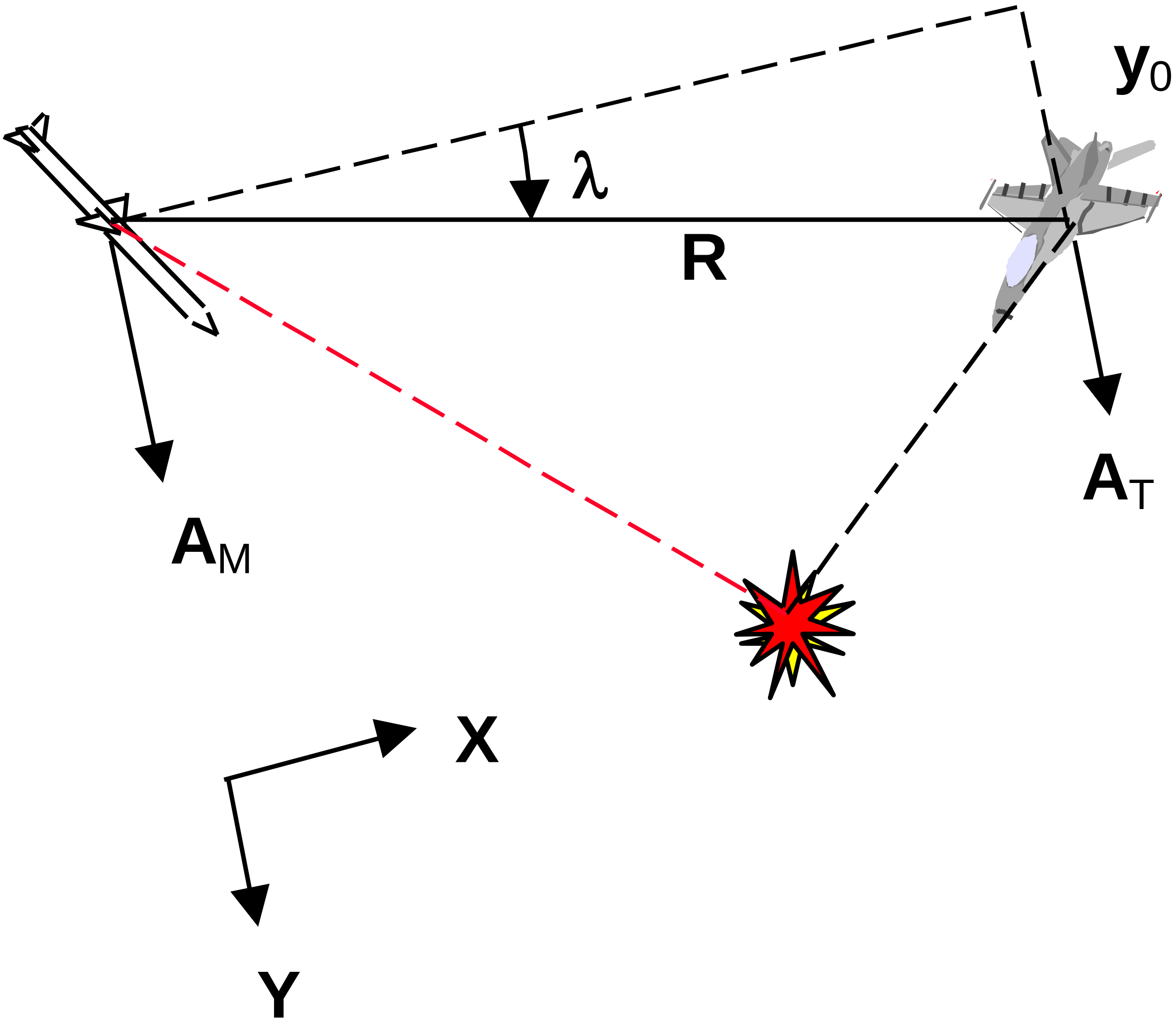



**Figure 1. PN Derivation Schematic**

## LIST OF TERMS

- $A_M$ = missile acceleration normal to closing
- $A_T$ = target acceleration normal to closing
- N` = navigation constant (gain)
- R = missile to target range
- $v_c$ = closing velocity = $d$R
- $\lambda$ = line of sight angle
- t = intercept time-to-go
- y = y-axis missile position – y-axis target position
- $y_0$ = initial y-axis missile position – initial y-axis target position
- $y_F$ = y-position at point of closest approach
- $d$y = $v_M$ – $v_T$ (missile velocity advantage)

Using these terms and

Figure 1. PN Derivation Schematic, the acceleration command required to result in a zero miss distance can now be expressed as the following general Equation 2 (Hanson). In this equation

**Equation 2:** $(\frac{1}{2}A_M)t^2 = y_0 + (dy)t + (\frac{1}{2}A_T)t^2$

Since standard PN does not assume any target manoeuver (i.e. $A_T = 0$) then Equation 2 can be rearranged and solved for $A_M$ :

**Equation 3:** $A_M = 2*(y_0/t^2 + dy/t)$

We have thus solved for the normal missile acceleration $A_M$ in terms of time, initial position, and the difference between the missile and target velocity. Setting aside Equation 3 temporarily, we can use

Figure 1. PN Derivation Schematic and mathematical manipulations to duplicate the right hand side of Equation 3 only this time relating it to the three basic terms in

Equation 1. Using

Figure 1. PN Derivation Schematic and small angle theory we can solve for the missile turning angle ($\lambda$):

**Equation 4:** $\lambda \sim y_0/R = y_0*R^{-1}$

The derivative $d\lambda$ can then be calculated and both sides multiplied by $-dR$:

**Equation 5:** $d\lambda = -y_0*dR/R^2 + dy/R$

**Equation 6:** $dRd\lambda = y_0(dR^2/R^2) - dy(dR/R)$

Now, since R is defined as $t\,d\mathrm{R}$ then $(1/t) = (d\mathrm{R}/\mathrm{R})$ and we can substitute into Equation 6 and get the same second term as in Equation 3:

**Equation 7:** $-d\mathrm{R}d\lambda = y_0/t^2 + dy/t$

Substituting the second term into Equation 3 and realizing that the closing velocity ($v_c$) is defined as the negative rate of change of the missile target separation ($-d\mathrm{R}$) then:

**Equation 8:** $A_M = 2*(-d\mathrm{R}d\lambda) = (2)\, v_c\, d\lambda = N`v_c\, d\lambda$

This (Equation 9) is equal to

Equation 1 as desired. It should be noted that this derivation results in a constant acceleration command (N'=2). N can be modified to result in different acceleration profiles as will be shown later.

**Equation 9:** $n_c = N`v_c\, d\lambda$

The utility of this concept is very broad and has enjoyed tremendous popularity. It should be noted that this popularity is a result not only of its effectiveness in minimizing intercept miss distance, but also of its simplicity to design, implement, and modify.

# PROPORTIONAL NAVIGATION MODEL

The following exercise is designed to demonstrate the concepts involved in the development of critical PN Guidance algorithms in modern missile systems. Many items of complexity, themselves worthy of entire projects or books, have been simplified or omitted. These topics include items such as varying winds aloft, speed variations, turbulence, imperfect fin design, motor variations in thrust power and time, lag time in servo's, etc. in order to clarify the more significant PN concepts. Gravitational and drag effect have also been neglected.

A main priority is that the guidance system must work in almost any circumstance with as much autonomy as possible. To this end, the design engineers must insure a PN guidance algorithm that minimizes processing, maximizes robustness, and adapts to changing situations.

## Scenario

The basic scenario for the model is to simulate the intercept of a target and a missile using PN guidance. A two dimensional scenario was used as there is no real need to go to the added complexity of a three dimensional model. Most all of the fundamental limitations and strengths of PN control can be fully realized and demonstrated in the two dimensional model. The coordinate system used is an ECEF (earth-centered, earth-fixed) reference frame with a flat-earth model. Axis (or Vector) 1 can be considered down-range while Axis 2 can be either cross-range or altitude. This scenario has chosen Axis 2 to represent cross-range.

## Assumptions

- Constant Missile Speed
- Constant Target Speed

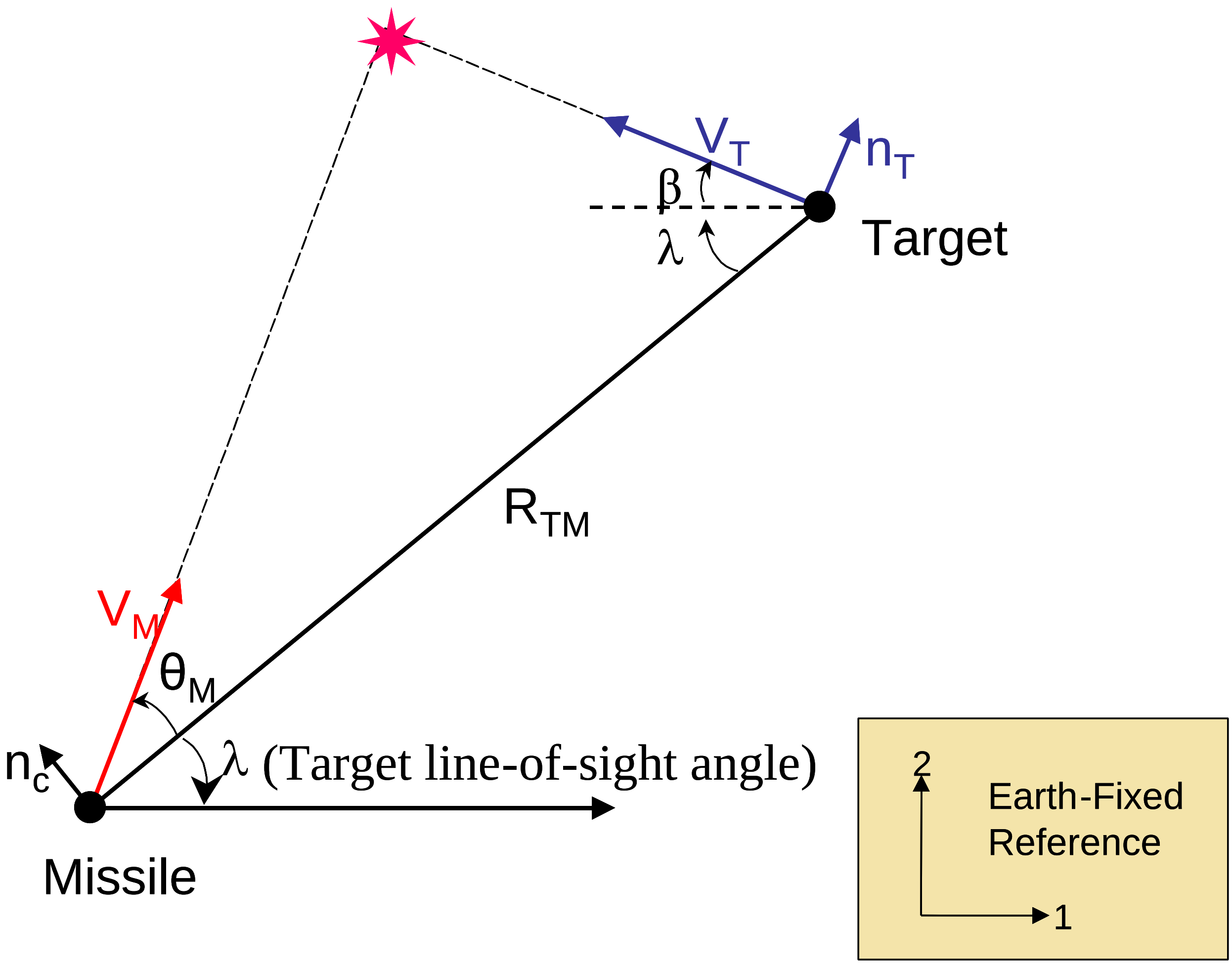



**Figure 2. PN Law “Collision Triangle” Diagram**

## Initial Conditions (t=0)

The initial conditions will be inputs to the missileproject.m file (the run file) of the model for ease of manipulation. It will be made clear throughout the text when any of the initial conditions are modified to suit a specific point or example.

- $V_{T,0}$ = 1000ft/s (for Low speed code) -2900ft/s(for High Speed Code)

(Target Velocity at Time=0)

- $\beta_0$ = 0deg (Angle of Target wrt Vector 1)
- $V_{M1,0}$ = 0 (Vector 1 Missile Velocity at Time=0)
- $V_{M2,0}$ = 3000ft/s (for Low speed code) -3900ft/s(for High Speed Code)

(Vector 2 Missile Velocity at Time=0)

- $R_{TM1,0}$ = 20kft (Vector 1 Target to Missile Range at Time=0)
- $R_{TM2,0}$ = 20kft (Vector 2 Target to Missile Range at Time=0)

## Target Manoeuver Inputs (3g turn-out example)

Any number of potential simple or complex manoeuvers can be included in the simulation. Additionally, start and stop times for each manoeuver or combination of manoeuvers can be specified. As can be seen in Appendix A, the manoeuvers chosen as representative and included in this model were: No Manoeuver, 3g or 5g turn-out (shown in the example below), 3g or 5g turn-in, and a 3g side-to-side manoeuver.

- $t_{manoeuver}$ = 5s (Manoeuver time from t=0)
- $t < t_{manoeuver}$: $n_T = 0$ (Straight and level flight)
- $t >= t_{manoeuver}$: $n_T = -3g$ (3g turn-out)

## Outputs

- $R_{TM}$ = Missile to Target Range (minimize this distance)

## Ordinary Differential Equation Calculations

The PN guidance law for the closing turn rate: $\boxed{\mathbf{n_c = N`v_c}\, d\lambda}$ as presented earlier will now come into play. In this equation, the parameters $v_c$ (closing velocity) and $\lambda$ (turning angle) can be expressed directly in terms of $R_{TM}$ (Figure 2) as detailed in section

ODE Calculations while $n_c$ (commanded normal missile acceleration) and N` (navigation constant chosen by the designer) were defined previously. I will state these two equations now and return to them in the section developing the ODE equations.

**Equation 10** $\mathbf{v_c = -dR_{TM}}$

**Equation 11** $\boldsymbol{\lambda} = \mathbf{tan^{-1}(R_{TM2}/R_{TM1})}$

The next step toward developing a working model is to express the system shown in Figure 2 as a set of representative states. Note that there are many potential state sets that will work. These have been chosen because they are proven to work well. The "trick" in developing a set of states is finding a set from which the time derivatives can be expressed easily in terms of other states as will be shown in the next sections. Proper selection of these states can make the difference between a straightforward or very difficult model to implement.

## Target States

- $R_{T1}$ = Vector 1 Target Position
- $R_{T2}$ = Vector 2 Target Position
- $V_T$ = Target Velocity
- $\beta$ = Target Velocity Angle with respect to Vector 1

## Missile States

- $R_{M1}$ = Vector 1 Missile Position
- $R_{M2}$ = Vector 2 Missile Position
- $V_{M1}$ = Missile Velocity with respect to Vector 1
- $V_{M2}$ = Missile Velocity with respect to Vector 2

## ODE Calculations

Set up the formal set of ordinary differential equations (ODE) for each state in the system. Since the ODE equations are used to solve for the change in the state of the entire system over time, the set of ODE equations are created directly by taking the time (t) derivative of each individual system state and expressing the result in terms of the other system states.

**Target States:** The time derivatives of the target position states $R_{T1}$ and $R_{T2}$ are simply the velocity in the respective axes, which can then be expressed in terms of the target angle β (Figure 2).

**Equation 12** $\boldsymbol{dR_{T1} = V_{T1} = -V_T\cos\beta}$

**Equation 13** $\boldsymbol{dR_{T2} = V_{T2} = V_T\sin\beta}$

The time derivative of the target velocity, given the assumption of constant target velocity (axis 2 to represent cross-range.

**Assumptions**) equals zero.

**Equation 14** $\boldsymbol{dV_T = 0}$

An incremental change in target angle β can be expressed (Figure 2) as an incremental time delta multiplied by the ratio of the $n_T$ and the target velocity.

The time derivative can then be expressed in terms of the target velocity state and the given input value $n_T$ as shown in Target Manoeuver Inputs (3g turn-out example).

**Equation 15** $d\beta \Rightarrow \Delta\beta = n_T \Delta t / V_T \Rightarrow d\beta = n_T / V_T$

**Missile States:** The time derivatives of the missile position states $R_{M1}$ and $R_{M2}$ are simply the velocity in the respective axes.

**Equation 16** $dR_{M1} = V_{M1}$

**Equation 17** $dR_{M2} = V_{M2}$

The time derivatives of the missile velocity vectors $V_{M1}$ and $V_{M2}$ can be expressed (using Figure 2) in terms of the commanded normal missile acceleration and the turning angle.

**Equation 18** $dV_{M1} = - n_c \sin\lambda$

**Equation 19** $dV_{M2} = n_c \cos\lambda$

The next step is to calculate the Equation 18 and Equation 19 parameters $\mathbf{n_c}$ and $\lambda$ in terms of the chosen set of state variables. Combining the missile to target distance in each axis into $R_{TM1}$ and $R_{TM2}$, $\lambda$ can then be calculated as the inverse tangent of the ratio. $\lambda$ has now been expressed in terms of existing state variables.

**Equation 20** $R_{TM1} = R_{T1} - R_{M1}$

**Equation 21** $\mathbf{R_{TM2} = R_{T2} - R_{M2}}$

**Equation 22** $\boldsymbol{\lambda = \tan^{-1}(R_{TM2}/R_{TM1}) = \tan^{-1}[(R_{T2} - R_{M2})/(R_{T1} - R_{M1})]}$

This leaves $\mathbf{n_c}$ as our last variable to put in terms of the chosen state variables.

The PN Guidance Law for the closing turn rate: $\mathbf{n_c = N`v_c}\ \boldsymbol{d\lambda}$ as presented earlier (

Equation 1) will now come into play. Defining the target to missile closing velocity for each axis into $V_{TM1}$ and $V_{TM2}$, we can complete the $v_c$ equation (Equation 10) in terms of state variables.

**Equation 23** $\mathbf{V_{TM1} = V_{T1} - V_{M1}}$

**Equation 24** $\mathbf{V_{TM2} = V_{T2} - V_{M2}}$

**Equation 25** $\mathbf{v_c = dR_{TM} = (R_{TM1}V_{TM1} + R_{TM2}V_{TM2})/R_{TM}}$

With the N` navigation constant chosen by the designer and $\lambda$ calculated in Equation 22 we can at length solve for $d\lambda$ and thus express $\mathbf{n_c}$ in terms of existing state variables. Starting with the prior calculation for $\lambda$ we have:

**Equation 26** $\boldsymbol{d\lambda = d\tan^{-1}(R_{TM2}/R_{TM1})}$

Using calculus and the chain rule applied to this inverse tangent function the function is expanded as follows:

**Equation 27** $\boldsymbol{d\lambda = 1/[1+(R_{TM2}/R_{TM1})^2]\ d(R_{TM2}/R_{TM1})}$

Rearranging by multiplying the top and bottom of the first part by $R_{TM1}^2$ and using the quotient rule to expand the derivative:

**Equation 28**

$$d\lambda = [R_{TM1}^2/( R_{TM1}^2+R_{TM2}^2)]*[R_{TM1}dR_{TM2} - R_{TM2}\, dR_{TM1}]/ R_{TM1}^2$$

Cancelling the resultant $R_{TM1}^2$ terms and creating a variable $R_{TM}$ = SQRT($R_{TM1}^2$+ $R_{TM1}^2$) this can be simplified to:

**Equation 29** $$d\lambda = [1/R_{TM}^2]*[R_{TM1}dR_{TM2} - R_{TM2}\, dR_{TM1}]$$

The final manipulation involves expressing the derivatives of the range terms $dR_{TM1}$ and $dR_{TM1}$ in terms of velocity as shown in Equation 30 and Equation 31and substituting these into Equation 29 to obtain the solution of $d\lambda$ in terms of state variables shown in Equation 32.

**Equation 30** $$dR_{TM1} = V_{TM1}$$

**Equation 31** $$dR_{TM2} = V_{TM2}$$

**Equation 32** $$d\lambda = [1/R_{TM}^2]*[R_{TM1}V_{TM2} - R_{TM2}\, V_{TM1}]$$

Using Equation 25 for $v_c$ Equation 32 for $d\lambda$ we can then substitute into Equation 1 and express $n_c$ in terms of state variables.

**Equation 33** $$n_c=N'*(R_{TM1}V_{TM1}+R_{TM2}V_{TM2})/R_{TM}*[1/R_{TM}^2]*[R_{TM1}V_{TM2}- R_{TM2}\, V_{TM1}]$$

Finally, substituting this result into Equation 18 and Equation 19 completes our set of ordinary differential equations for the model.

**Equation 34** $$dV_{M1} = -N'(R_{TM1}V_{TM1}+R_{TM2}V_{TM2})/R_{TM} *[1/R_{TM}^{2}]* [R_{TM1}V_{TM2}-R_{TM2} V_{TM1}]*\sin[\tan^{-1}(R_{TM2}/R_{TM1})]$$

**Equation 35** $$dV_{M2} = N'(R_{TM1}V_{TM1}+R_{TM2}V_{TM2})/R_{TM} *[1/R_{TM}^{2}]* [R_{TM1}V_{TM2}-R_{TM2} V_{TM1}]*\cos[\tan^{-1}(R_{TM2}/R_{TM1})]$$

The resultant ODE set is presented in **Error! Reference source not found.** and shall be considered as the basic control system used. Modifications to the basic equation set shall be stated as applicable.

# Matlab™ Model

The missile and target ordinary differential equations given in **Error! Reference source not found.** were then used to create a simple 2 dimensional non-linear Matlab™ model using Runge-Kutta numerical integration. The model itself is separated into an PN ODE module (ODE equations, PN Gain constant(s), Target Manoeuvers, Hardware Limitations, etc.) called "ProNav51.m" and a PN Run module (Initial Conditions, PN ODE Module Call, Output Plots, etc.) called ProNavRun51.m". Matlab™ functions ode23 or ode45 were both implemented and these gave comparable results for most applications. The proper order for the Runge-Kutta depends on the specific application and may even change based on the phase of flight for a single application. The code is documented in its entirety below.

# MATLAB™ MODEL CODE

## odeProNav.m (Non-linear ODE Equations)

```
function xdot = odePropNav(t,x)

% INPUT INITIAL STATE CONDITIONS
Rt1 = x(1);
Rt2 = x(2);
Vt = x(3);
Beta = x(4);
Rm1 = x(5);
Rm2 = x(6);
Vm1 = x(7);
Vm2 = x(8);
nc = x(9);
xdot = zeros(size(x));

% GAIN INPUTS TO THE MODEL
Npr = 2.2;                         %PROPORTIONAL NAVIGATION CONSTANT
TermNpr = 4;                     %TERMINAL NAVIGATION CONSTANT
TermRange = 4000;                %TERMINAL RANGE
maxg = 10;                       % MAX G-FORCE MISSILE HARDWARE CAN
 SUSTAIN

% FOUR MANEUVER INPUTS TO THE MODEL
% NO MANEUVER
 nt=0;

% SIDE TO SIDE MANEUVER 5G
% nt=32*5*sin(T);

% LATE 5G TURNOUT
%if t<5
    %nt=0;
%else
    %nt=32*5;
%end

% LATE 3G TURN- IN
%if t<0
%    nt=0;
%else
%    nt=-32*3;
%end



%STATE EQN CALCULATONS
Rt1dot = -Vt*cos(Beta);
Rt2dot = Vt*sin(Beta);
Vtdot = 0;
Betadot = nt/Vt;
Rm1dot = Vm1;
Rm2dot = Vm2;
```

```
%CALCULATION OF LAMBDA
Rtm1 = Rt1-Rm1;
Rtm2 = Rt2-Rm2;
lambda = atan2(Rtm2,Rtm1);

%CALCULATION OF  nc
Vt1 = -Vt*cos(Beta);
Vt2 = Vt*sin(Beta);
Vtm1 = Vt1-Vm1;
Vtm2 = Vt2- Vm2;
RtmSq =Rtm1^2 + Rtm2^2;
Rtm = sqrt(RtmSq);

if Rtm <= -3 %LAMDA LOGIC FOR TERMINAL RANGE ZERO HANDLING
    lambdadot = 0;
    Vc = 0;

else
    lambdadot = (Rtm1*Vtm2-Rtm2*Vtm1)/RtmSq;
    Vc= -(Rtm1*Vtm1 + Rtm2*Vtm2)/Rtm;
end

% ENTER TERMINAL PHASE (HIGHER GAIN) IF WITHIN TERMINAL RANGE
if Rtm < TermRange
    Npr = TermNpr;                  %REPLACING THE MIDCOURSE GAIN WITH
 TERMINAL GAIN
end
nc = Npr*Vc*lambdadot;

if abs(nc) >32*maxg
    nc = 32*maxg*sign(nc);
end

Vm1dot = -nc*sin(lambda);
Vm2dot = nc*cos(lambda);

xdot= [Rt1dot; Rt2dot; Vtdot; Betadot; Rm1dot; Rm2dot; Vm1dot; Vm2dot;
 nc;];
```

*Published with MATLAB® R2021a*

**missileproject.m** (Run Script for odeProNav.m)

**(high speed with no manoeuver)**

```
xlabel('Time (sec)')                    % X Coordinate Label
ylabel('Separation (feet)')             % Y Coordinate Label
title('MISSILE TO TARGET DISTANCE')

% PLOTTING THE GODS-EYE TRAJECTORIES OF MISSILE AND AIRCRAFT
% FIGURE(2)
subplot(2,2,2)
plot(Rm1,Rm2,'r',Rt1,Rt2,'g')
xlabel('DownRange (feet)')                    % X Coordinate Label
ylabel('CrossRange (feet)')             % Y Coordinate Label
title('GODS-EYE TRAJECTORY')
text(1000,5000,'Missile')
text(16000,19000,'Target')


fprintf('\n THE MISS DISTANCE FOR GAIN = 2.2 , TERMINAL GAIN 4 IS %G
 FEET.',miss_distance)

function xdot = ProNav(t,x)


% INPUT INITIAL STATE CONDITIONS
Rt1 = x(1);
Rt2 = x(2);
Vt = x(3);
Beta = x(4);
Rm1 = x(5);
Rm2 = x(6);
Vm1 = x(7);
Vm2 = x(8);
nc = x(9);
xdot = zeros(size(x));

% GAIN INPUTS TO THE MODEL
Npr = 2.2;                        %PROPORTIONAL NAVIGATION CONSTANT
TermNpr =4 ;                    %TERMINAL NAVIGATION CONSTANT
TermRange = 4000;               %TERMINAL RANGE
maxg = 10;                      % MAX G-FORCE MISSILE HARDWARE CAN
 SUSTAIN

% FOUR MANEUVER INPUTS TO THE MODEL
% NO MANEUVER
nt=0;

% SIDE TO SIDE MANEUVER 5G
%nt=32*5*sin(t);

% LATE 5G TURNOUT
%if t<5
    %nt=0;
%else
    %nt=32*5;
%end
```

```
% LATE 3G TURN- IN
%if t<0
    %nt=0;
%else
    %nt=-32*3;
%end



%STATE EQN CALCULATONS
Rt1dot = -Vt*cos(Beta);
Rt2dot = Vt*sin(Beta);
Vtdot = 0;
Betadot = nt/Vt;
Rm1dot = Vm1;
Rm2dot = Vm2;

%CALCULATION OF LAMBDA
Rtm1 = Rt1-Rm1;
Rtm2 = Rt2-Rm2;
lambda = atan2(Rtm2,Rtm1);

%CALCULATION OF  nc
Vt1 = -Vt*cos(Beta);
Vt2 = Vt*sin(Beta);
Vtm1 = Vt1-Vm1;
Vtm2 = Vt2- Vm2;
RtmSq =Rtm1^2 + Rtm2^2;
Rtm = sqrt(RtmSq);

if Rtm <= -3 %LAMDA LOGIC FOR TERMINAL RANGE ZERO HANDLING
    lambdadot = 0;
    Vc = 0;

else
    lambdadot = (Rtm1*Vtm2-Rtm2*Vtm1)/RtmSq;
    Vc= -(Rtm1*Vtm1 + Rtm2*Vtm2)/Rtm;
end

% ENTER TERMINAL PHASE (HIGHER GAIN) IF WITHIN TERMINAL RANGE
if Rtm < TermRange
    Npr = TermNpr;                    %REPLACING THE MIDCOURSE GAIN WITH
 TERMINAL GAIN
end
nc = Npr*Vc*lambdadot;

if abs(nc) >32*maxg
    nc = 32*maxg*sign(nc);
end

Vm1dot = -nc*sin(lambda);
Vm2dot = nc*cos(lambda);
```

```
xdot= [Rt1dot; Rt2dot; Vtdot; Betadot; Rm1dot; Rm2dot; Vm1dot; Vm2dot;
 nc;];
end
```

*Published with MATLAB® R2021a*

**lsmissileproject.m (Run Script for odeProNav.m)**

**(low speed with no manoeuver)**

```
%INITIAL CONDITIONS
Rt10 = 20000;       %ft
Rt20 = 20000;       %ft
Vt0 = 1000;         %ft/sec
Beta0 = 0;           %deg
Rm10 = 0;           %ft
Rm20 = 0;           %ft
Vm10 = 0;           %ft/sec in vector 2 direction
Vm20 = 3000;        %ft/sec in vector 2 direction
nc = 0;

%TIME CONDITIONS
gaptime = 0.08;     %Time Between Which We Have No Information On The
 Target
endtime = 13.5;     %sec
time = [0:gaptime:endtime];

%EXECUTION LINE FOR SOLVING THE ORDINARY DIFFERENTIAL EQUATION (ODE23)
%MATRIX INPUTS ARE TIME (0 TO RUNTIME) AND INITIAL CONDITIONS

[t,x] = ode23(@odePropNav,time,
[Rt10;Rt20;Vt0;Beta0;Rm10;Rm20;Vm10;Vm20;nc]);

%EXTRACTING LINE FROM ODE23 EXECUTION
Rt1 = x(:,1);
Rt2 = x(:,2);
Rm1 = x(:,5);
Rm2 = x(:,6);
Vm1 = x(:,7);
Vm2 = x(:,8);
nc2 = x(:,9);
```

*Published with MATLAB® R2021a*

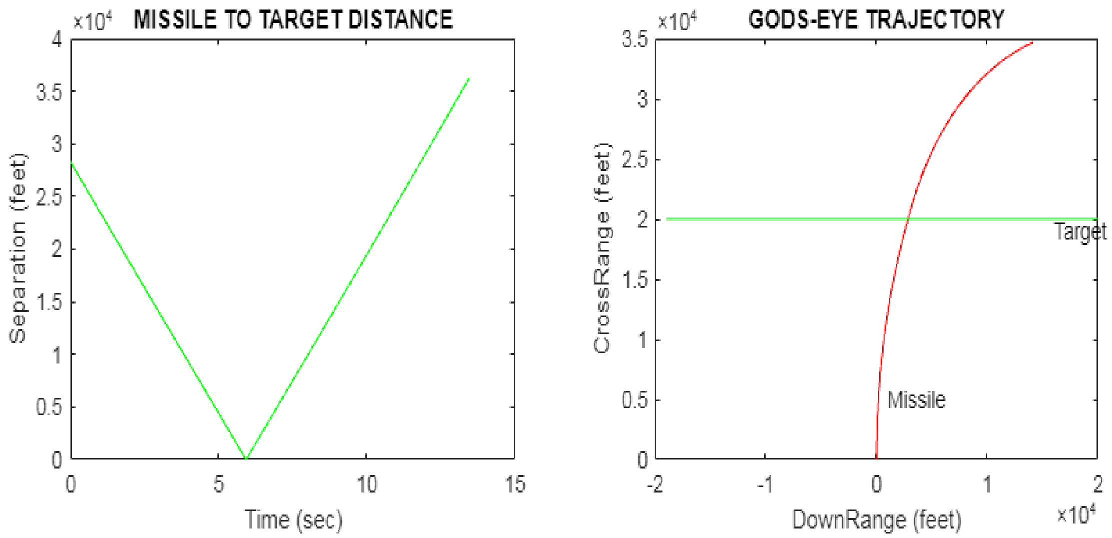
# 2-DIMENSIONAL MODEL RESULTS

During this chapter, it is our privilege to play the role of "System Designer". At last we will have a chance to observe the impact that a chosen navigational gain will have upon different target manoeuvers and heading angles. Initial model runs were executed using the scenario given before and the Initial Conditions (t=0) in order to observe general performance using the Matlab™ "ode23" function which uses 2nd and 3rd order Runge Kutta integration.

With the 2 dimensions chosen as downrange and cross range, a typical plot for addressing general performance is commonly referred to as a God's-Eye plot.

It should be noted that the calculated miss distance is calculated as the minimum missile to target range from the set of points and is thus an estimate.

## 1. High Speed with No Manoeuver

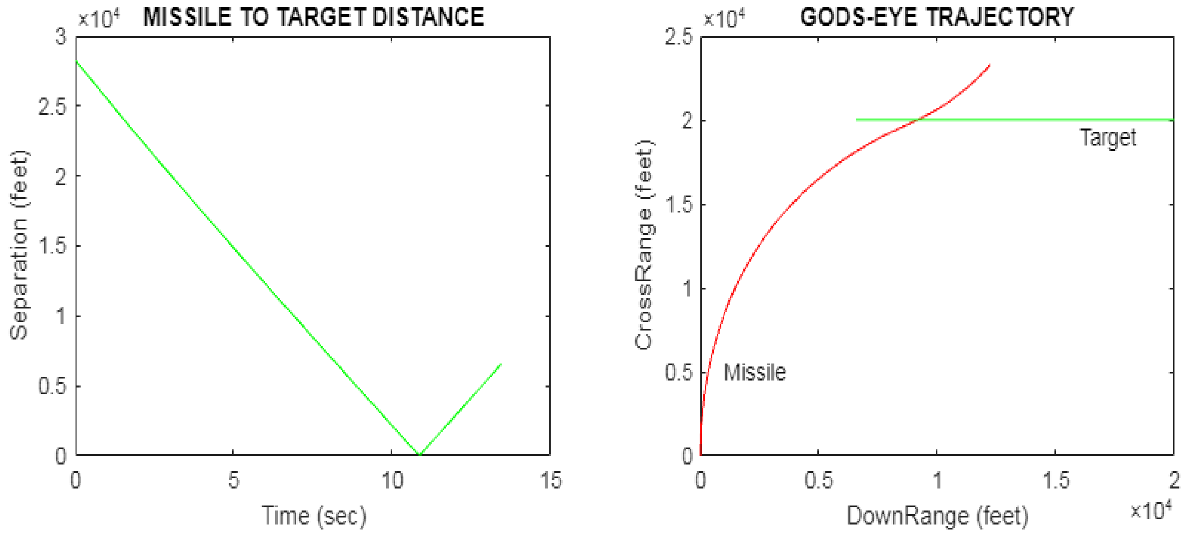



The Miss Distance for Midcourse Gain **2.2**, Terminal Gain **4** is **0.981338 feet**.

The gains for this particular scenario was highly optimized over a number of iterations.

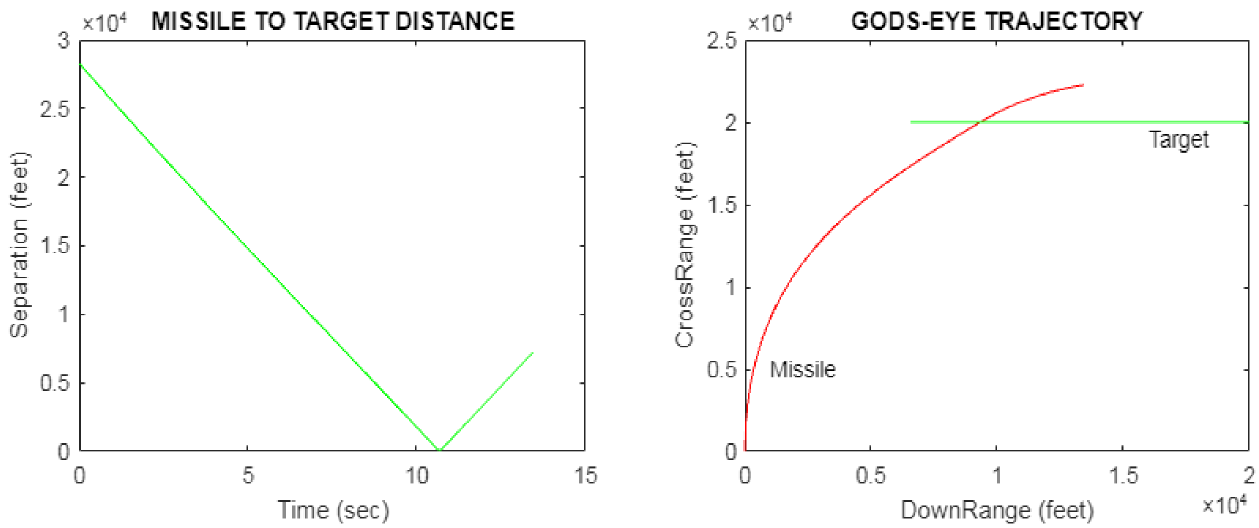
## 2. Low Speed with No Manoeuver

## I.



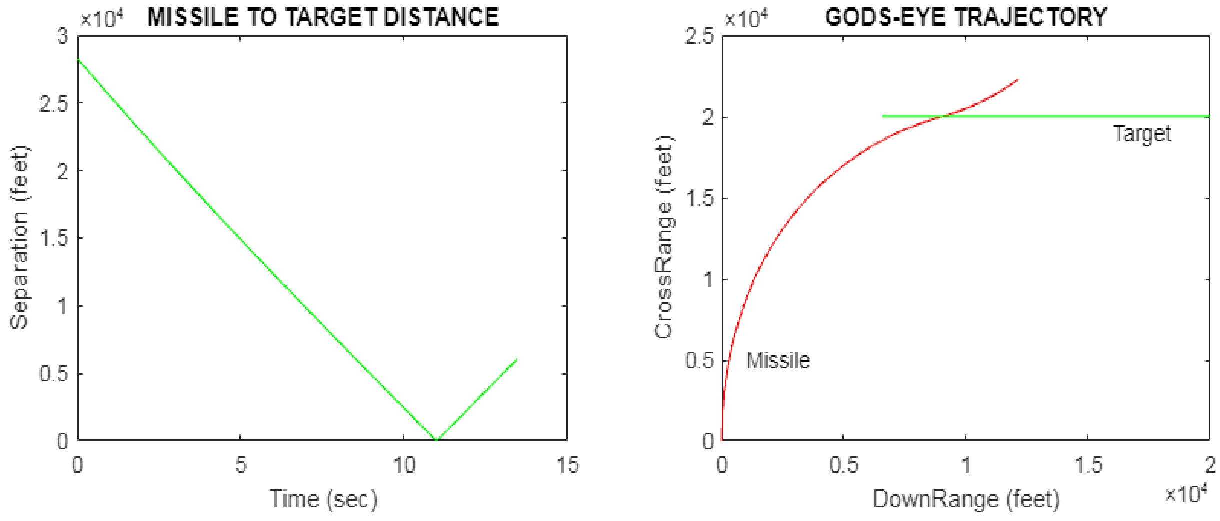

The Miss Distance for Midcourse Gain = **2.2**, Terminal Gain **4** Is **81.2901 feet**.

## II.



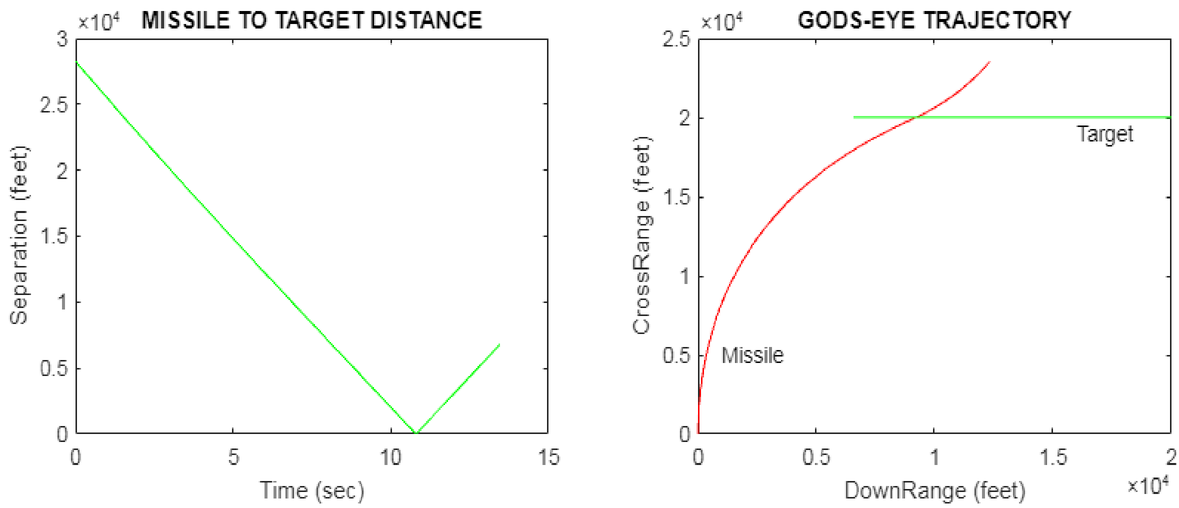

The Miss Distance for Midcourse Gain = **3**, Terminal Gain **4** Is **91.6456 feet**.

# III.



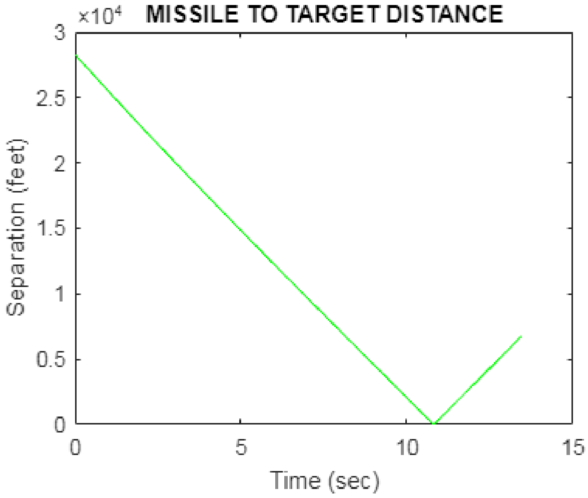

The Miss Distance for Midcourse Gain = **2**, Terminal Gain **2** Is **65.719 feet**.

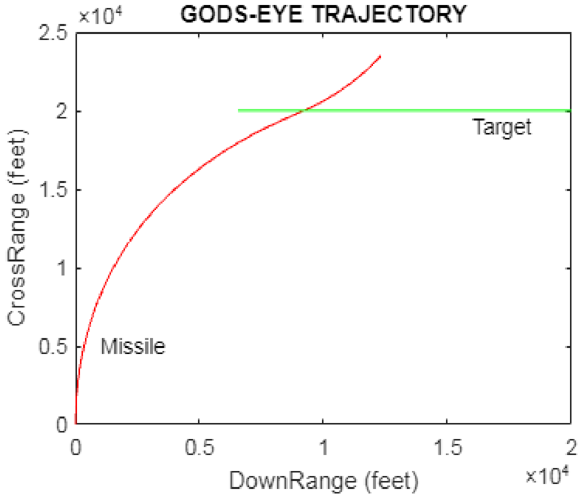
# IV.




The Miss Distance for Midcourse Gain = **2.5**, Terminal Gain **3** Is **15.2257 feet**.

## V.




The Miss Distance for Midcourse Gain = **2.287**, Terminal Gain **3** Is **0.26773 feet**.

These were the best gains for this scenario.

## Target Maneuvers

Repeating the steps in the prior section can be used on other target manoeuvers and will give a good feel for how robust the PN algorithm (and chosen gain value) is for a variety of manoeuvers, severity of manoeuvers, and manoeuver timing. The following plot sets are intended to illustrate the PN guidance performance against a variety of target manoeuvers. In addition (and more importantly) the somewhat cumbersome nature of these plots will then be used as a stepping stone into the critical next chapters that enable us to combine thousands of these model executions into a single computer run.

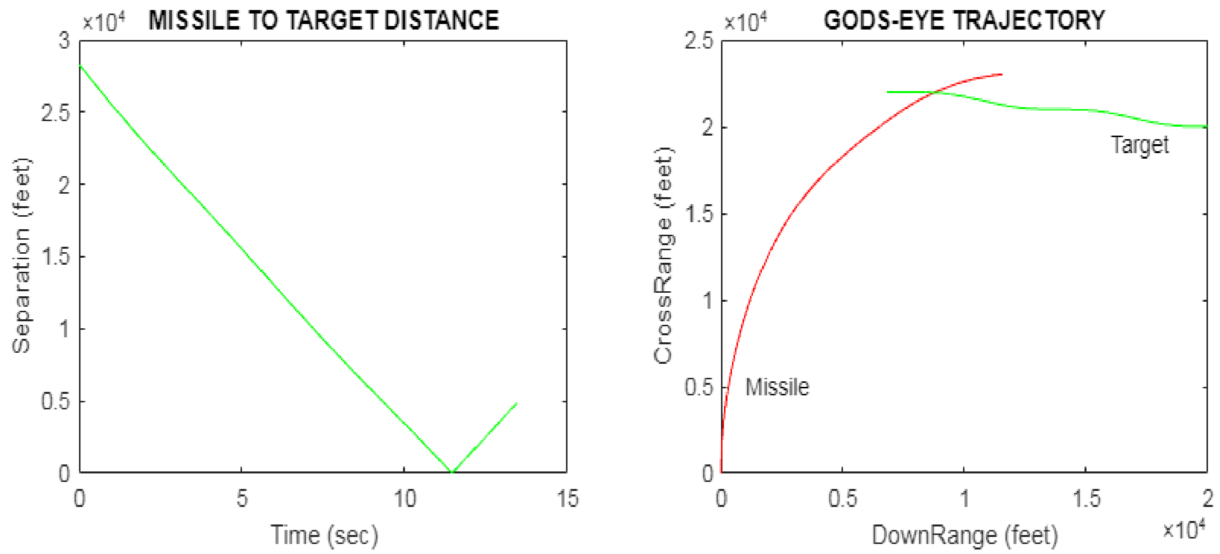
The listed manoeuvers were chosen as a representative sample that would be studied for a typical missile design. For comparability, all runs in this section have been executed using a midcourse gain of "2" and a terminal gain of "2":

- No Manoeuver
- 3G Turn In / 5G Turn In

- 3G Turn Out / 5G Turn Out
- 5G Side-to-Side Manoeuver

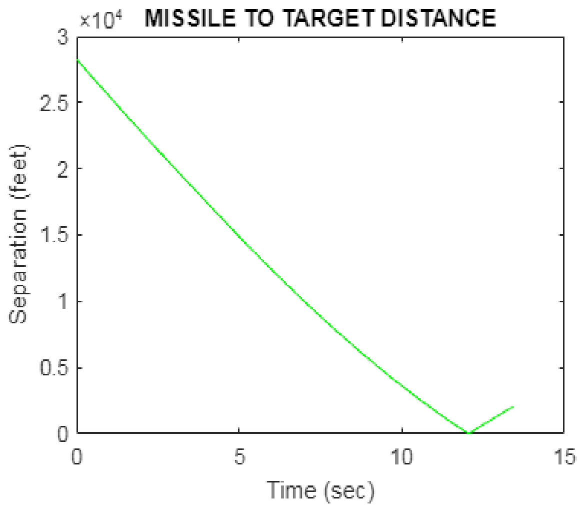
Results are provided with each plot set and detail including the miss distance.

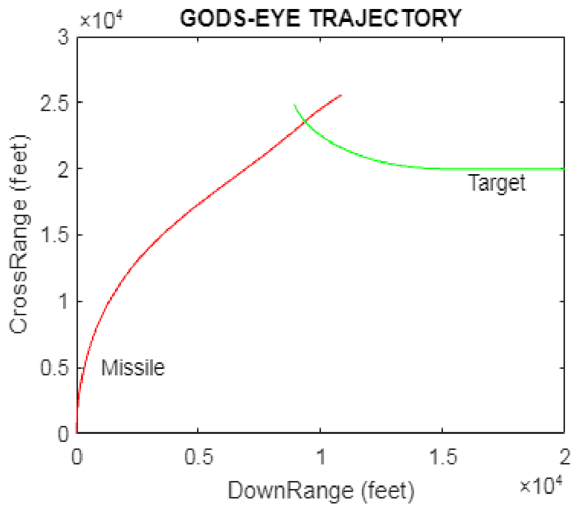
## I. Side To Side Manoeuver 5g




The Miss Distance for Midcourse Gain = **2**, Terminal Gain **2** Is **36.051 feet**. The side-to-side maneuver, although I thought this might be extremely effective, was fairly easy for the model to handle. A primary reason for this might be that the model does not consider radar cross section of the target or noise effects. As a result, the achieved miss distance was low (comparable to the no target maneuver scenario).

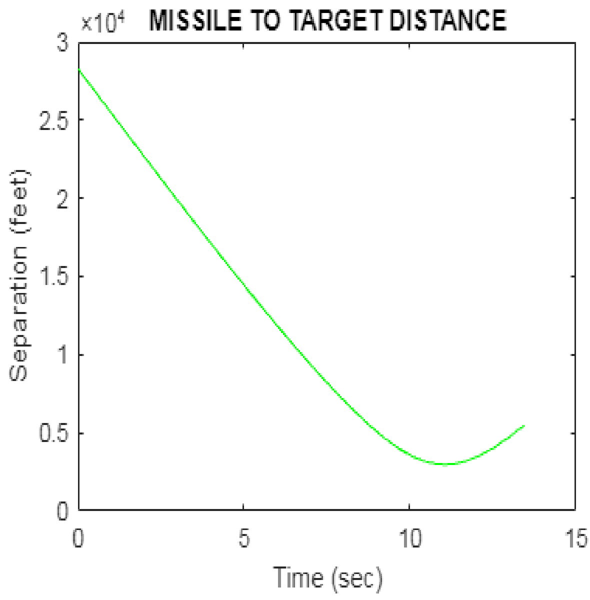
## II. Late 5g Turn-Out

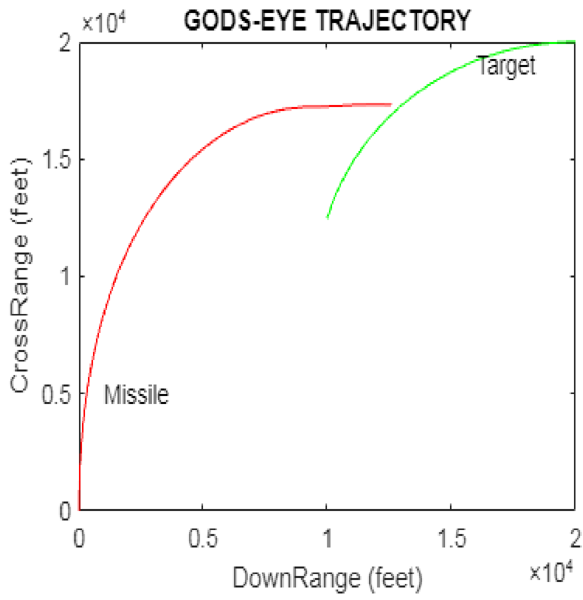





The Miss Distance for Midcourse Gain = **2**, Terminal Gain **2** Is **48.446 feet**. The turn-out manoeuver is easier for a missile to handle than the turn-in. A primary reason for this is that it has additional distance=time to make corrections and the relative closing velocity decreases to the difference between $V_T$ and $V_M$. Not surprisingly, the achieved miss distance is low (comparable to the no target manoeuver scenario).

## III. Late 3g Turn-In






The Miss Distance for Midcourse Gain = **2**, Terminal Gain **2** Is **2971.33 feet**

A turn-in manoeuver at the correct time is an extremely effective avoidance manoeuver, as demonstrated by the large increase in miss distance. A primary reason for this is that it has less distance=time to make corrections and the relative closing velocity increases as the scenario becomes more of a head-on engagement. Again, the relatively low gain constant used results in the missile never achieving a "collision triangle".

# RESULTS SUMMARY

- The model confirms that at a simplistic level, starting with a midcourse gain of 2 and a terminal gain between 2 and 4 is correct.
- Higher gains, particularly in the terminal phase can improve miss distance.
- Higher gains will be costly in the boost and midcourse phases when noise is factored in.
- Target manoeuvring (in general) can be used to bleed missile energy (speed).
- PN (and general intercept problems) are much more susceptible to turn-in target manoeuvers than to turn-out target manoeuvers due to the increase in closing velocity.

# SUMMARY AND CONCLUSIONS

Missiles and rockets being currently developed are a complex marvel of accelerometers, resolvers (control mixers), rollerons (gyroscopic/aerodynamic roll stabilizers), range radar, and doppler radar. Although simplifications and omissions have been made to focus on the principles and avoid certain complexities, the general concepts stand sound and form the foundation of much of the guidance and control engineers tasking. Each of the major design problems have been examined including:

- Minimization of Miss Distance… as impacted by many factors.
- Stability / Robust behaviour… over a variety of manoeuvers, gain constants, etc.
- Saturation of Missile Parameters … requiring model limits on commanded acceleration.
- Model Processing Limitations… requiring innovative tools to reduce the workload.

This project has studied many of the practical considerations and manipulations required to realize good (application specific) proportional navigation systems. A useful two dimensional non-linear model was then designed using a set of ordinary differential equations and Runge-Kutta numerical integration techniques. The model was used to draw several conclusions about the utility, robustness, and design trade-offs involved in PN control theory and application design.

This concludes our study into the basic derivation, principles of proportional navigation as well as introducing modelling used by the guidance and control system designer. In truth, this project has just scratched at the surface of the work involved to develop a functional guidance system.

# REFERENCES

**Rocket Guidance** (www.webcom.com/sknkwrks/guidance.htm) Sourced July 2005.

**Paul B. Jackson**, Overview of Missile Flight Control Systems.

**Neil F. Palumbo, Ross A. Blauwkamp, and Justin M. Lloyd**, Basic Principles of Homing Guidance.

**Michael J Voss,** Proportional Navigation (Principles / Applications / Modelling).

**Zarchan P**, Tactical and Strategic Missile Guidance (4th Edition), 2002.