Numerical Solution to a Nonlinear External Ballistics Model for a Direct Fire Control System

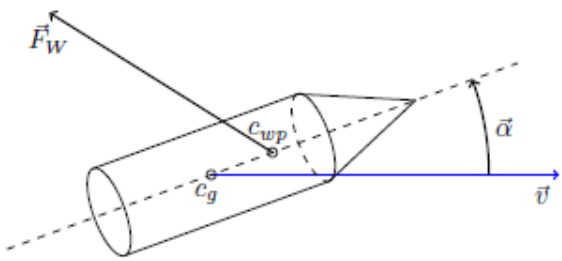
MARTIN SKANDE

KTH Industrial Engineering
and Management

Master of Science Thesis
Stockholm, Sweden 2014

Numerical Solution to a Nonlinear External Ballistics Model for a Direct Fire Control System

Martin Skande



Master of Science Thesis MMK 2014:58 MDA 487
KTH Industrial Engineering and Management
Machine Design
SE-100 44 STOCKHOLM

KTH Industriell teknik och management

Examensarbete MMK 2014:58 MDA 487

Numerisk Lösning för en Olinjär Ytterballistik till ett Direkt Eldledningssystem

Martin Skande

Godkänt	Examinator	Handledare
2014-06-26	Jan Wikander	Bengt O. Eriksson
	Uppdragsgivare	Kontaktperson
	Fredrik Lundh	Daniel Hellberg

Sammanfattning

Detta examensarbete är en utvecklings- och evalueringsprocess av en numeriskt löst ytterballistik modell för ett eldledningssystem utvecklat av SAAB Technologies. Den befintliga ballistik modell som används av eldledningssystemet idag är linjäriserad vilket simplifierar problemet att nå upp till de hårda realtidskrav som finns, dock på grund av de approximationer och simplifieringar knutna till linjäriseringen finns det utrymme för förbättringar i termer av beräkningsnoggrannhet. Ökande krav på systemnivå prestanda tillsammans med initiala utredningar motiverade en undersökning av en numeriskt löst ytterballistik i termer av prestanda relativt den befintliga linjäriserade ytterballistik modellen.

Examensarbetet har fokuserat på hela processen från första koncept av en numeriskt löst ytterballistik modell ända till implementering på eldledningssystemet. Arbetet är grundat i ett teoretiskt ramverk i huvudsak inriktat på ballistikteori.

Konceptet som har tagits fram använder en olinjär ytterballistikmodell baserad på en befintlig modell beskriven i teori och löses med hjälp av en fjärde ordningens Runge-Kutta integrator. En iterativ metod liknande en diskret kontrollstruktur används för att finna initial- och ändvärden till ballistikmodellen och en adaptiv steglängdsalgoritm används som klarar av att upprätthålla realtidskraven. Parametrisering av modellen gjordes mot eldledningstabeller från ammunitionstillverkare och den resulterande beräkningsnoggrannheten visade sig vara förbättrad relativt den befintliga linjära ytterballistik modellen.

Slutsatserna av projektet är att en numeriskt löst olinjär ytterballistik är möjlig trots de hårda realtidskrav som finns på systemet samt kan förbättra systemnivå prestanda, speciellt för långa avstånd. Dock på grund av att konceptet ökar komplexiteten av ballistik mjukvaran samt komplicerar parametriseringen så avråds SAAB från att använda denna lösning såvida systemnivåprestandan inte anses ett problem.

KTH Industrial Engineering and Management

Master of Science Thesis MMK 2014:58 MDA 487

Numerical Solution to a Nonlinear External Ballistics Model for a Direct Fire Control System

Martin Skande

Approved	Examiner	Supervisor
2014-06-26	Jan Wikander	Bengt O. Eriksson
	Commissioner	Contact person
	Fredrik Lundh	Daniel Hellberg

Abstract

This Masters Thesis is a development and evaluation process of a numerically solved nonlinear external ballistics model for a direct Fire Control System developed by SAAB Technologies. The currently used external ballistics model is linearized which significantly simplifies the task of meeting hard real-time constraints on the system but due to the approximations and simplifications tied to the linearization of the external ballistics there exists room for improvements in terms of calculation accuracy. Increasing demands on system level performance together with initial investigations on the Fire Control System concluded that there was interest for an evaluation of a numerical solution in terms of calculation accuracy and processing load.

The project focuses on the process from concept development of a numerical external ballistics solution all the way to actual implementation on the targeted Fire Control System. The work is founded on a theoretic frame of reference on primarily external ballistics theory.

The concept that has been developed uses a nonlinear external ballistics model based on an existing Modified Point Mass Trajectory Model (MPMTM) described in the theoretic frame of reference and is solved using a fourth order Runge-Kutta integrator. An iterative method similar to that of a discrete time control loop is used to find boundary values of the ballistics model satisfying hard real-time constraints on the system. Parameterization of the external ballistics model was carried out using firing-tables provided from the projectile manufacturer and the resulting calculation accuracy was improved relative the existing linearized solution.

The conclusion of the project is that a numerical solution is possible despite the hard real-time constraints on the system and can increase the system level performance, particularly so for long range trajectories. However because the solution increases the complexity of the ballistics calculation software and complicates the parameterization method it is not advised unless the system level performance is deemed a problem.

Nomenclature

MIL	Model In the Loop
SIL	Software In the Loop
HIL	Hardware In the Loop
PIL	Processor In the Loop
RTOS	Real-Time Operating System
FCS	Fire Control System
MBD	Model Based Development
t_{fl}	Flight time of the projectile from muzzle to target
SE	Superimposed Elevation
SA	Superimposed Azimuth
QE	Quadrant Elevation
$\vec{x}_t$	Target position
x_t, R_t	Distance to target
$\dot{\vec{x}}_t$	Target velocity
$\dot{x}_t$	Target speed
$\vec{x}_{poi}$	Impact point position
x_{poi}, R_{poi}	Distance to impact point position
α	Angle of attack or yaw angle
α_e	Yaw of Repose
t_{del}	Time delay

C_D	Aerodynamic drag coefficient
C_L	Aerodynamic lift coefficient
C_{Mag}	Aerodynamic Magnus coefficient
C_{Spin}	Aerodynamic spin damping coefficient
S	Cross section area

Contents

Chapter 1

INTRODUCTION

1.1 History and background

In etymology dictionaries the word *ballistics*; "art of throwing; science of projectiles", dates back as far as 1753 based on the Latin word *ballista*, a military machine for hurling stones as well as the Greek *ballistes* from *ballein*, "to throw; to throw as to hit". The art of ballistics is ancient, shaping projectiles such as arrows for a better, more predictable flight through air. Later on, ballistics was broken down into subfields such as *internal* ballistics referring to the study of propulsion of a projectile and *terminal* ballistics referring to the behavior and effects of a projectile when hitting its target. Today however, the dictionaries description would perhaps fit best to the subfield *external* ballistics, referring to the behavior of a non-powered or free-flight projectile in the air.

Complex external ballistic models of projectile trajectories taking into account air temperature, humidity, pressure, wind and even the rotation of Earth through the Coriolis effect are used in gyro stabilized weapon platforms today. Models like these are implemented and solved in embedded systems, referred to as Fire Control Systems (FCS). FCS usually has the capability to calculate the trajectory for several different projectiles, each projectile type having its own fitted aerodynamic coefficients and parameters in a external ballistics model. A common problem in live firing scenarios is that the calculations have to be made quickly so there exists time constraints, due to processing performance limitations model simplification is therefore often a necessity.

Initial investigations with SAAB on live, direct FCS concluded that there is interest for an evaluation of a numerical ballistics calculations software developed for such an embedded system, showing opportunities to increase system performance, which gave raise to the Master Thesis proposal.

This is a Master Thesis project in Mechatronics at the Royal Institute of Technology in Stockholm. The project work is made at SAAB Technologies, business area Security and Defence Solutions in Järfälla.

1.2 Purpose

The master thesis aims to evaluate performance when a nonlinear external ballistics trajectory model is solved numerically compared to that of a linearized model. A existing linearized model known to and developed by SAAB that can be solved algebraically will serve as the basis for comparison in terms of model accuracy and calculation intensity. To do this the numerically solved nonlinear model needs to be parameterized against ballistic data in the form of firing tables, just as the linearized model. The main purpose is to evaluate the ballistics calculation on the embedded system and depending on the validity and performance of the numerical method, it may serve as a basis for future work. The external ballistics needs to be feasible for use in a targeted live and direct FCS and if possible even implemented. The different problems and tasks involved in developing a nonlinear external ballistic trajectory model and implement such a model on the target FCS is described in more detail in the following section.

1.3 Detailed problem description

The objective of the FCS is to aim with the sight on target and at the same time point the muzzle of the gun with a specific deviation from the line of sight so as to give the highest probability of hitting the target. The deviation angles, in this thesis called the Superimposed-Elevation (SE) and Superimposed-Azimuth (SA), are the ballistic offsets as well as the estimated lead angles caused by the relative motion between gun and target. With sensors or user inputs, a number of estimated variables are input to an external ballistics model to predict specifically the time of flight of the projectile from muzzle to target (t_{fl}) as well as needed SE and SA.

Range to predicted impact point where target and projectile meets, atmospheric density, elevation angle to predicted impact point, charge temperature, wind direction, wind speed, relative target motion and geographical latitude and longitude name a few variables that may need to be known depending on the external ballistics model complexity and degree of accuracy. The SE and SA angles are mainly used in the reference for the servo control of the turret actuators. The t_{fl} is used for programmable munitions but will also be needed in the prediction of the impact location for a moving target. Apart from the nonlinear ballistics model and a numerical method to solve this, a method that solves the boundary conditions involved in finding a trajectory that hits the target is also needed.

The external ballistics calculation is carried out in software modules implemented on an embedded system with limited processing performance. It is only a few of several tasks running on the embedded system, a very basic overview of the relevant calculation tasks running on the embedded system is shown in figure 1.1. The embedded system uses a Real-Time Operating System (RTOS) with preemptive task scheduling and runs the tasks periodically in different cycle frequencies. The tasks that this master thesis mainly concerns is the ballistics calculation soft-

ware, the impact point prediction calculation software and the interfacing there between. The processing power of the system is limited and the computational intensity rises with model complexity, so a balance between model accuracy, adaptability, complexity and calculation intensity is needed if it is to be feasible for usage in the target FCS considering the hard real-time constraints involved.

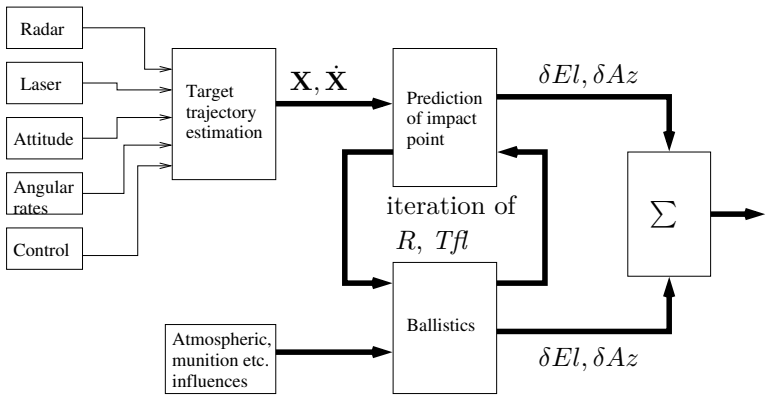


Figure 1.1: Simplified overview of relevant tasks in FCS

The FCS should be capable of solving the projectile trajectory for several munitions, each projectile has different properties and needs to be parameterized to the developed model. Data is available in the form of firing tables for each projectile individually. These firing tables consists of the necessary ballistic offsets SE and SA, time of flight to target t_{fl} and the end-point velocity ($\dot{x_p}$) for a number of different trajectories corresponding to target locations at different ranges. The influences such as atmospheric conditions and the range to target and elevation (EL) to the target are stated for every trajectory. Using this data, a method for parameterization of the developed trajectory model from the firing tables is needed.

1.4 Method

The thesis involves the development of a external ballistics trajectory model, an numerical integration method for this model, methods for solving the boundary conditions to hit the target, a prediction algorithm to find the target location at impact time as well as test cases and implementation of the model on the target platform. To achieve this the basis behind the method is inspired by a Model Based Development (MBD), MIL-SIL-PIL-HIL process with several test along the process of development in order to evaluate, validate and verify. The list below summarizes the process:

1. A literature study of the subject matter is made which is summarized in a theoretic frame of reference.

2. A first concept with the design of the external ballistics model, the integrative method and boundary value solution is developed.

3. The concept is implemented in MATLAB®.

4. Testing of the first concept. This test serves to evaluate the model, integrative methods and algorithms developed. Aerodynamic coefficients and other model parameters from a known projectile can be used at this stage.

5. Parameterization algorithms. The external ballistics trajectory model needs to be parameterizable from the limited amount data available in firing tables. Methods for doing this is researched and developed at this stage. There is a possibility that the external ballistics trajectory model may also need to be revised at this stage in order for the parameterization to be feasible.

6. Parameterization testing and evaluation. A suitable projectile is chosen and the firing tables for this projectile is used to find any aerodynamic coefficients and parameters needed in the model for this projectile. The process is evaluated in MATLAB®.

7. MIL testing and results. Complete testing of the model is done at this stage with the parameters from the before mentioned process in MATLAB®. The model accuracy relative the firing tables is compared to the results of the known linear model. The purpose is to evaluate the performance at this stage to the linear model and if possible, verify that the new model meets set system level requirements in terms of ballistic calculation accuracy relative firing tables.

8. Implementation of the complete model in the target platform and hardware. Software is written in C++ and implemented on the targeted FCS.

9. Implementation of specific test cases in the target platform. Some test cases are written into the software to make the final verification and validation process simpler.

10. Calculation intensity evaluation. A simpler form of evaluation of the calculation intensity on the embedded system is of interest to verify the feasibility of the concept in the hard real-time environment.

11. HIL testing and final results. The final testing will be carried out on the target platform, just as in the MIL test comparisons are made between the numerical and the known linearized approach in terms of calculation accuracy. This final test will also speak for the validity of using this type of external ballistics in the targeted platform. Discussion and conclusions are to follow.

1.5 Delimitations

There are several delimitations of the thesis, the list below is a summarization of both important limitation considering what methods and algorithms can be chosen as well as limitations in the project work:

- The firing tables consists of a very limited amount of data points (usually around 30-100), which severely limits the available methods for parameterization and system identification.
- The integration method is to run periodically on the target platform with timing-constraints, as such there will be hard limitations in how many cycles and consequentially integration steps the numerical ODE solver can take.
- The external ballistics shall be capable of taking target movement into account.
- The targeted platform is a live, direct FCS, meaning it needs a visual line of sight to the target. This imposes some practical limitations in the range to target, specifically there exists a maximum range outside of which there are no system level requirements in calculation accuracy because of the improbability of hitting the target and difficulties in visually obtaining the target, its movement and the range. This maximum range may differ between projectiles can be shorter then the available trajectories in the firing tables.
- Parallax compensation due to the locational difference between sight and gun position on the turret does not need to be considered, it is done separately in the target FCS.
- While the target platform implementation should be carried out, there is a possibility that such implementation is unfeasible due to hard real-time constraints in the FCS.

Chapter 2

FRAME OF REFERENCE

2.1 A linearized approach

A simpler modeling of the trajectory might be beneficial in terms of low calculation intensity and yet achieve sufficient model accuracy. This section gives a simplified view of how the current linearized external ballistics model is derived. The basic forces acting on the projectile are shown in figure 2.1 and include the drag force $\vec{F}_D$, gravitational force $m\vec{g}$ as well as a force $\vec{F}_N$ which can be considered a compensation for approximations and linearizations.

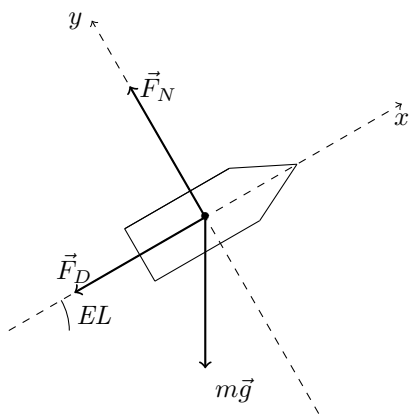


Figure 2.1: 2DOF modeling of a projectile trajectory

The task of calculating SE, SA and t_{fl} is broken down separately. Starting with the t_{fl} calculation certain assumptions and linearizations are made to simplify the problem and make it possible to solve analytically. For this to be possible the calculation t_{fl} does not consider the arc of the trajectory, the projectile is assumed

to fly in a straight line towards the target. For such a projectile the retardation of the projectile due to drag a_D can simply be put up as the following:

$$m\ddot{x} = -S\frac{\rho}{2}v^2 C_D \tag{2.1}$$

The speed of sound for an ideal gas:

$$v_s^2 = \kappa p v \tag{2.2}$$

The Mach number:

$$M = \frac{v}{v_s} \tag{2.3}$$

By combination of 2.3 and 2.2 into 2.1 the following equation is reached:

$$\ddot{x} = -K_0 p \Theta \tag{2.4}$$

where

$$K_0 = \frac{\kappa S}{2m} \tag{2.5}$$

and

$$\Theta = M^2 C_D \tag{2.6}$$

p is the pressure and assumed to be constant throughout the trajectory. An analytical linearized approximation of the aerodynamic drag coefficient C_D is made that works well for the case $M > 1$, this is parameterized from a few points in the tabular data. The result is a differential equation that is simply solved by for example Laplace transformation.

Determination of Superimposed-Elevation *SE* and Superimposed-Azimuth *SA*

When t_{fl} has been determined SE is approximated by:

$$SE = \sin\left(\frac{y}{x}\right) \approx \frac{y}{x} \tag{2.7}$$

The vertically traveled distance y is solved from:

$$\ddot{y} = -\cos(EL)\, g + \frac{F_N}{m} \tag{2.8}$$

where F_N is a linearized compensation force assumed to depend on $\ddot{y}$ and $\dot{y}$. As before the result is a linear ODE that can be solved easily.

The Superimposed-Azimuth SA is a simple polynomial approximation dependent on only SE.

The full model is analytically solvable, given a certain t_{fl} the projectile position and deviation angles can be calculated directly. To parameterize this model a Least Mean Square method can be used to fit the model to tabular data. This results in a rather straight forward solution that is suitable for a hard real-time environment.

2.2 Aerodynamic coefficients

Aerodynamic drag is the resistance from a object moving through a fluid like air or water. In external ballistics the different forces and moments caused by air drag resistance play a major role of determining where the projectile will end up. To this end big efforts are made for each developed projectile to determine a number of different aerodynamic coefficients, properties and dependencies by munition manufacturers and ballisticians alike. Typical ways of determining these are through live-firing tests with Doppler radar measurements, Computer Fluid Dynamics (CFD) analysis tools and extensive wind tunnel testing. Modeling the drag can be a great challenge as it is very hard or often near impossible to analytically model these aerodynamic coefficient due to the complex behavior of the flow of air around the projectile due to its shape. From years of knowledge we know that it is possible to represent the drag force felt by a body by the equation below: (Gregorek, 1970)

$$F_D = S\frac{\rho}{2}v^2 C_D \tag{2.9}$$

Where S is the cross-section area of the body, for a projectile flying straight this is generally the area of an circle with diameter d as the caliber of the projectile. ρ is the density of air, considerations should be made that the density is not constant but varies with atmospheric conditions and altitude. v is the projectile speed with respect to air and C_D is the aerodynamic drag coefficient, that varies with the orientation of the flow around the object. A widely adopted way of modeling the drag coefficient in external ballistics is: (Jizhang Sang, 2013) (Baranowski, 2013c) (Dickinson, 1965) (Celmins, 1990)

$$C_D = C_{D_0}(M) + C_{D_{\alpha^2}}(M)\alpha^2 \tag{2.10}$$

Where C_{D_0} is the zero-yaw drag coefficient and is a function of the Mach number M and $C_{D_{\alpha^2}}$ is the quadratic yaw drag coefficient as a function of the Mach number

M and proportional to the quadratic yaw angle α^2. α is the angle of attack, or angle of yaw towards the direction of the flow, if the projectile is flying straight the angle of yaw α is equal to zero.

Apart from the drag force, there exists other forces due to the air flow, such as lift and Magnus forces. These forces will be covered later in section 2.3, however their aerodynamic coefficients are generally modeled in the same way.

An example of a linearly interpolated C_{D_0} dependent on M, is available in figure 2.2. The data is from a 8-inch HE M106 projectile available for public by Dickinson (1965) and shows the typical shape of the aerodynamic drag coefficient for a projectile.

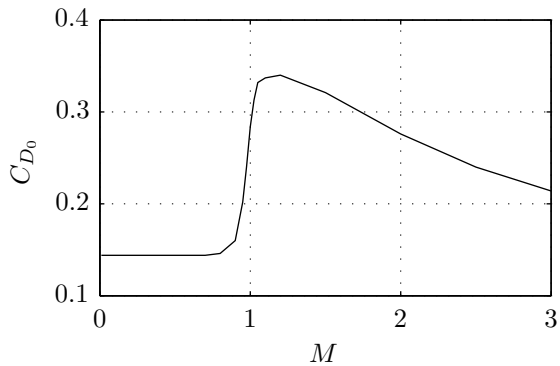


Figure 2.2: The zero-yaw drag coefficient of an 8-inch HE M106 projectile

2.3 Equations of motion for a rigid body projectile with 6DOF

For a full mathematical description of the dynamic projectile motion, the projectile has to be considered as a rigid body with 6 degrees of freedom (6DOF). This type of modeling is computationally intensive and typically not suitable for live-firing scenarios due to processing power limitations of the FCS and time constraints (Baranowski, 2013b). The main use of these type of models is for flight stability testing of spin-stabilized projectiles or calculation of firing tables. In this section a brief overview is presented of the the different forces and moments affecting a projectile when considered as a rigid body in free flight derived from a theoretic frame of reference including Baranowski (2013a), Baranowski (2013c), Baranowski (2013b), Celmins (1990) and Nennstiel (2013).

Equations of motion

The force of gravity acts on the center of mass of the projectile directed towards the center of earth. It proportional to the mass and the local acceleration of gravity which commonly is modeled as function of the latitude and distance from the center of Earth.

$$\vec{F_g} = m \cdot \vec{g} \tag{2.11}$$

The fictitious Coriolis force due to the rotational velocity of Earth is a function of the angular velocity vector of earth $\vec{\Omega}$ and the velocity of the projectile relative earth $\dot{\vec{x}}$.

$$\vec{F_c} = 2m \left(\dot{\vec{x}} \times \vec{\Omega} \right) \tag{2.12}$$

For the wind forces acting on the projectile the velocity with respect to air $\vec{v}$ is needed, which is dependent on the velocity of the projectile $\dot{\vec{x}}$ and the velocity of the wind $\vec{w}$.

$$\vec{v} = \dot{\vec{x}} - \vec{w} \tag{2.13}$$

Furthermore to consider the projectile as a rigid body the orientation angles of the projectile relative the ground-fixed coordinate system are needed. A second projectile body-fixed Cartesian coordinate system with unit vectors $\vec{e_{x_p}}$, $\vec{e_{y_p}}$ and $\vec{e_{z_p}}$ is therefore introduced with its origin in the center mass of the projectile. The angle between the velocity vector $\vec{v}$ and the symmetry line along $\vec{e_{x_p}}$ is defined as the angle of yaw α. Angular transformation methods between the ground fixed and projectile fixed coordinate system is typically needed to calculate α, Baranowski (2013a) utilizes Tait-Bryan angles which avoid singularities in the kinematic equations during angular transformation.

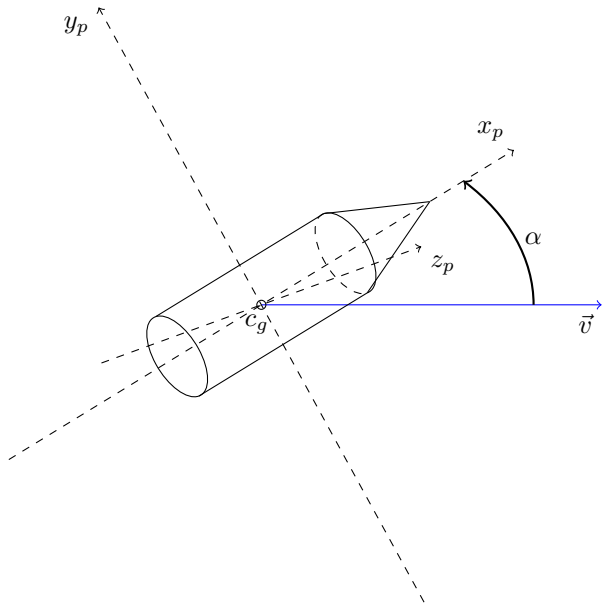


Figure 2.3: The angle of yaw α defined as the angle between the symmetry line of the projectile along x_p and the velocity vector $\vec{v}$.

The force of the wind F_W for a projectile with an angle of yaw α acts at a center of wind pressure of the projectile and in the same plane as the angle of yaw α. Where this center of pressure is located is dependent on the velocity, the angle of yaw, the geometry of the projectile and other factors. Typically it is located in front of the center of mass of the projectile which results in an overturning moment that acts to destabilize the projectile as can be seen in figure 2.4 and 2.5. Because of this some kind of stabilization is needed in order for the projectile to not tumble through the air. One way of solving this problem is to shape the projectile as to move the center of pressure behind the center of mass, such a projectile is self-stabilizing. A common way of doing this is by adding fins at the rear of the projectile, for this reason these types of projectiles are called fin-stabilized projectiles. The other way of stabilizing the projectile is to use gyroscopic forces. By adding an angular spin rate to the projectile along its symmetry line the overturning moment causes gyroscopic precession and nutation. With large enough spin rate this stabilizes the projectile similarly to a spinning toy top. It is worth noting that a spin-stabilized projectile can be over-stabilized, that is the angular spin rate of the projectile is so high that the projectile will not change its angular orientation along the arc of the trajectory and might come in at a high angle of yaw when reaching the target. (Baranowski, 2013a) (Nennstiel, 2013)

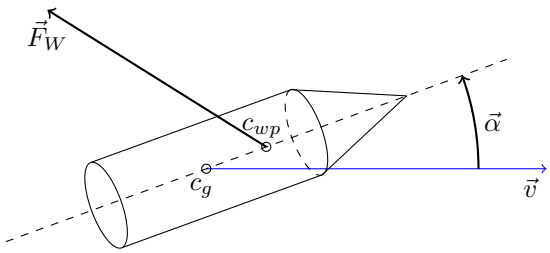


Figure 2.4: The wind force $\vec{F}_W$ acting on the center of wind pressure c_{wp} for a spin-stabilized projectile when traveling with velocity $\vec{v}$ and yaw angle $\vec{\alpha}$.

The wind force $\vec{F}_W$ is divided into two force components and an overturning moment acting on the center of gravity of the projectile. The drag force component $\vec{F}_D$ acting in the opposite direction of the velocity vector $\vec{v}$ and the lift force component $\vec{F}_L$ acting orthogonal to $\vec{F}_D$ and in the direction of $\vec{\alpha}$. The overturning moment $\vec{M}_W$ is the turning moment at the center of mass of the projectile due to the wind force $\vec{F}_W$.

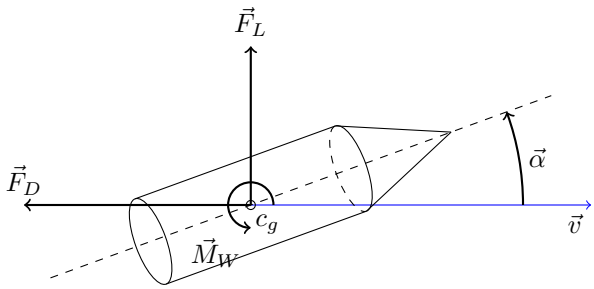


Figure 2.5: The resulting forces and moment acting on the projectiles center of gravity due to the wind force $\vec{F}_W$.

In figure 2.5 the overturning moment $\vec{M}_W$ acts to increase the angle of yaw α, for this reason the projectile is unstable and needs to be stabilized be means of gyroscopic moments due to spin. In accordance to Baranowski (2013a) and Nennstiel (2013) the different forces of the projectile can be presented as such:

$$\vec{F}_D = -S\frac{\rho}{2}C_D v \cdot \vec{v} \tag{2.14}$$

$$\vec{F}_L = S\frac{\rho}{2}C_L v^2 \cdot \vec{e}_L \tag{2.15}$$

$$\vec{M}_W = Sd\frac{\rho}{2}C_M v^2 \cdot \vec{e}_W \tag{2.16}$$

S is the cross section area of the projectile, d is the diameter of the projectile, $\vec{e}_L$ and $\vec{e}_W$ is force directions and C_D, C_L as well as C_M are aerodynamic coefficients. ρ is the air density and needs to be determined throughout the trajectory by means of atmospheric modeling. All aerodynamic coefficients are approximated as described in section 2.2.

The spin of a spin-stabilized projectile introduces forces and moments on the projectile. Because the projectile does not fly perfectly straight towards the wind, there exists a component of the flow-field perpendicular to the projectile axis of symmetry. This flow-field component coupled with the angular spin of the projectile creates areas of low and high pressure which results in a Magnus effect acting on a center of pressure usually located behind the center of mass depending on the projectile geometry. (Baranowski, 2013a) (Nennstiel, 2013)

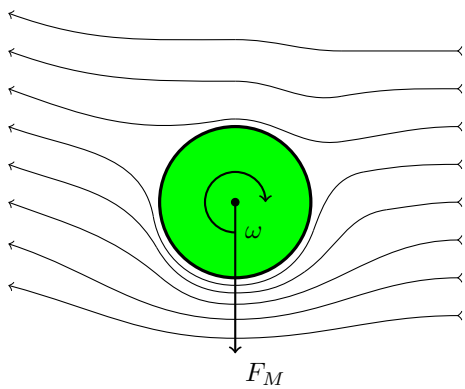


Figure 2.6: The flow-field component orthogonal to the symmetry line of the projectile viewed from the backplane of the projectile. F_M is the resulting force from the Magnus effect.

As before this force is split up into a moment M_M and a force F_M acting on the center mass of the projectile.

$$\vec{F}_M = S\frac{\rho}{2}C_{Mag}\omega d v \cdot \vec{e}_M \tag{2.17}$$

$$\vec{M}_M = S\frac{\rho}{2}C_{Mp}\omega d^2 v \cdot \vec{e}_{MM} \tag{2.18}$$

The rotational speed ω around the projectile symmetry line is damped out by a spin damping moment $\vec{M}_S$.

$$\vec{M}_S = -S\frac{\rho}{2}C_{Spin}\omega d^2 v \cdot \vec{e}_c \tag{2.19}$$

2.4 Modified Point Mass Trajectory Model

The Modified Point Mass Trajectory Model (MPMTM) proposed by Robert F. Lieske (1966) and reviewed by Baranowski (2013b) is a trajectory model based on the equation of motion of a projectile with an approximated term for the yaw angle α referred to as Lieskes yaw of repose $\vec{\alpha}_e$. This has the desirable feature of representing the effects of axial spin and angle of yaw along the trajectory. The MPMTM was developed for use in ground artillery FCS but can be used for producing ground firing-tables as well. Application of the MPMTM requires parameterization of a number of projectile data such as aerodynamic coefficients and several fitting factors introduced in the model. Baranowski (2013b) mentions the existence of documentation elaborating extensive methods for determining these fitting factors using live firing tests.

The force equation at the center of mass point of the projectile is given by several equations very similar to the ones described in section 2.3. The coordinate system is ground-fixed Cartesian with unit vectors $\vec{e}_1$, $\vec{e}_2$ and $\vec{e}_3$.

The equation of motion for the center mass of the projectile is:

$$m\ddot{\vec{x}} = \vec{F}_c + \vec{F}_g + \vec{F}_D + \vec{F}_L + \vec{F}_M \tag{2.20}$$

$\vec{F}_c$ is the fictitious force due to Coriolis effect:

$$\vec{F}_c = 2m\left(\vec{\omega} \times \dot{\vec{x}}\right) \tag{2.21}$$

where ω is the angular velocity of Earth:

$$\vec{\omega} = \begin{pmatrix} \Omega\cos(lat)\cos(az) \\ \Omega\sin(lat) \\ -\Omega\cos(lat)\sin(az) \end{pmatrix} \tag{2.22}$$

The latitude (lat) is negative in the southern hemisphere, the Azimuth (az) is the bearing of the 1-axis relative true North.

The gravitational force works in the radial direction towards the center of Earth with a magnitude of the gravitational acceleration that typically can be modeled as a function of the latitude and longitude:

$$\vec{F}_g = -mg_0(lat, az)\begin{pmatrix} \frac{X_1}{R_z} \\ 1 - \frac{2X_2}{R_z} \\ \frac{X_3}{R_z} \end{pmatrix} \tag{2.23}$$

The drag force is described similar to section 2.3 but with added fitting factors:

$$\vec{F}_D = -Si\frac{\rho}{2}\left[C_{D_O}C_{D_{\alpha^2}}\left(Q_D\alpha_e\right)^2\right]v\cdot\vec{v} \tag{2.24}$$

where Q_D is the yaw drag fitting factor and has been chosen as a constant and i is a cubic form factor dependent on the Quadrant Elevation (QE):

$$i = a_0 + a_1(QE) + a_2(QE)^2 + a_3(QE) \tag{2.25}$$

The lift force is described similar to section 2.3 but with an added fitting factor:

$$\vec{F}_L = Sf_L\frac{\rho}{2}\left(C_{L_\alpha} + C_{L_{\alpha^3}}\alpha_e^2\right)v^2\vec{\alpha}_e \tag{2.26}$$

where

$$f_L = b_0 + b_1(QE) + b_2(QE)^2 + b_3(QE) \tag{2.27}$$

The Magnus force is described similar to section 2.3 but with an added fitting factor:

$$\vec{F}_M = SdQ_M\frac{\rho}{2}C_{Mag}\left(\vec{\alpha}_e\times\vec{v}\right) \tag{2.28}$$

where Q_M is the Magnus force factor and has been chosen as a constant.

The angular spin rate along the projectile axis of symmetry is described as:

$$\dot{p} = \frac{Sd^2vC_{Spin}}{2I_x} \tag{2.29}$$

The yaw of repose equation is described as follows:

$$\vec{\alpha}_e = -\frac{2I_xp}{Sd\rho(C_{M_\alpha} + C_{M_{\alpha^3}}\alpha_e^2)v^4}(\vec{v}\times\dot{\vec{u}}) \tag{2.30}$$

Chapter 3

FIRST EXTERNAL BALLISTICS CONCEPT

3.1 Reference coordinate system

A reference coordinate system is needed to be able to model the equations of motion for a projectile. The coordinate system described in this section is originally based on the conventions for aircraft local reference frames as described in Int (1985) and conforms to what is described in Baranowski (2013b) and Robert F. Lieske (1966). It is a ground fixed right hand side Cartesian coordinate system with unit vectors $\vec{e}_1$, $\vec{e}_2$ and $\vec{e}_3$. The 1-3 plane creates the tangent surface of Earth with the 1-axis pointing in the direction of the line of sight and the 2-axis pointing straight up. The Azimuth az is the angle between the 1-axis of the coordinate system and the north direction. The latitude is positive in the northern hemisphere and negative in the southern hemisphere. Earth is approximated by a sphere with radius R_z and rotational speed Ω.

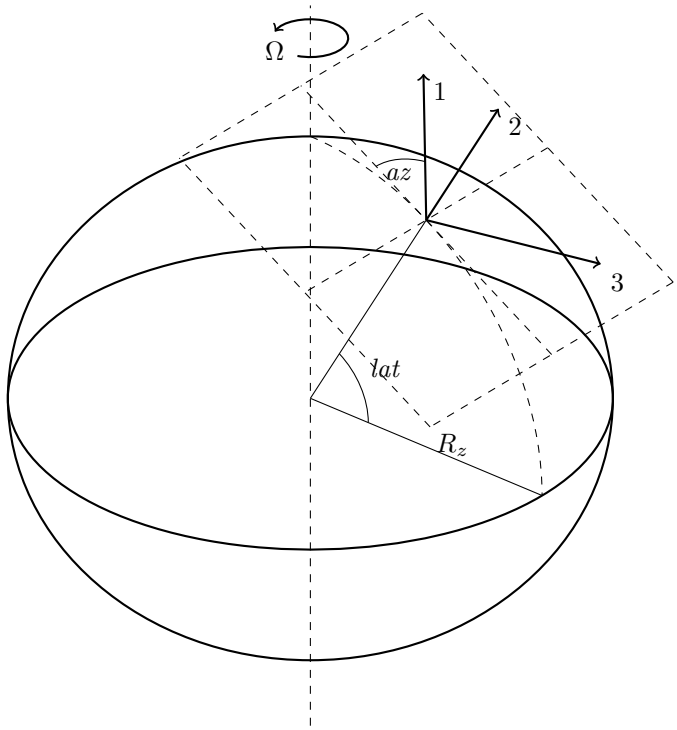


Figure 3.1: The Cartesian coordinate system.

Ballistic deviation angles definition

The center bore-line and the optical line of sight is separated by the Superimposed-Elevation angle *SE* and the Superimposed-Azimuth angle *SA*. The Elevation angle *EL* is the angle between the optical line of sight and the 1-axis, the Quadrant Elevation *QE* is the algebraic sum of *SE* and *EL*.

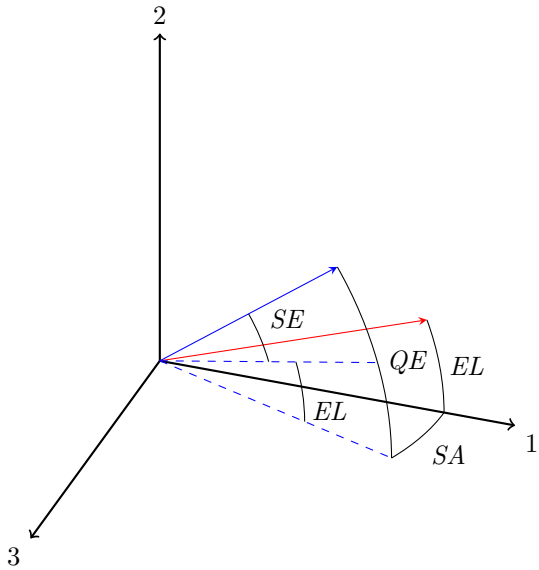


Figure 3.2: The optical line of sight is shown in red and the center bore-line is shown in blue. Note that since by definition the target is in the 1-2 plane, the line of sight can only have an elevation angle EL

The transformation between the polar coordinates and Cartesian coordinates is carried out as follows:

$$Elevation = arctan2\left(x_2, \sqrt{x_1^2 + x_3^2}\right) \tag{3.1}$$

$$Azimuth = arctan2\left(x_3, x_1\right) \tag{3.2}$$

$$x = \sqrt{x_1^2 + x_2^2 + x_3^2} \tag{3.3}$$

Where $arctan2$ denotes four quadrant inverse tangent, this is used to retain quadrant information in the polar transformation. The inverse transform is as follows:

$$x_1 = x\cos\left(Elevation\right)\cos\left(Azimuth\right) \tag{3.4}$$

$$x_2 = x\sin\left(Elevation\right) \tag{3.5}$$

$$x_3 = x\cos\left(Elevation\right)\sin\left(Azimuth\right) \tag{3.6}$$

$$\tag{3.7}$$

$$\tag{3.8}$$

3.2 External ballistics

The external ballistics model developed for use in FCS is based upon the theoretic frame of reference with consideration for delimitations in the available data for parameterization. The model must be simple in the context of computation but very precise from the point of view of firing in real atmospheric conditions. The MPMTM described in section 2.4 describes an widely adopted standardized model for just this purpose, however with the limitations in available data for parameterization the full MPMTM cannot be used because of the sheer amount of unknown variables and coefficient functions it introduces. It is therefore desirable to use this model as a basis for development and strategically reduce the model complexity to a level that permits parameterization of the unknown aerodynamic coefficient functions, initial values and other parameters directly from the ground-firing tables. The task of designing the trajectory model can be separated in three parts; defining a model of the gravitational force and atmospheric model, defining the reference coordinate system and its motion as well as defining the aerodynamic forces and its dependencies. The reference coordinate system defined in section 3.1 is used as is, since it conforms to the theoretic frame of reference the implemented model will be based upon. Earth is therefore approximated by a sphere with a radius of R_z and constant angular velocity $\vec{\Omega}$ as previously mentioned.

Gravitational force and atmospheric modeling

The equation for the gravitational force is the same as described in section 2.4.

$$\vec{F}_g = mg_0 \begin{pmatrix} \frac{x_1}{R_z} \\ 1 - \frac{2x_2}{R_z} \\ \frac{x_3}{R_z} \end{pmatrix} \tag{3.9}$$

The proposed model for gravitational acceleration conforms with the international gravity formula mentioned in Int (1997). This form of approximation is also what Baranowski (2013b) presents and seems to be a widely used approximation.

$$g_0 = g_n(1 - \alpha_g \cos(2lat) + \textit{smaller terms ...}) \quad \mathrm{m\,s^{-2}} \tag{3.10}$$

Where $g_n = 9.806\,65\,\mathrm{m\,s^{-2}}$ which corresponds to the acceleration constant at latitude 45°, $\alpha_g = 0.0026$ and $R_z = 6\,356\,766\,\mathrm{m}$. The smaller terms are omitted.

The Coriolis force for the reference system is as before mentioned:

$$\vec{F}_c = 2m\left(\vec{\omega} \times \dot{\vec{x}}\right) \tag{3.11}$$

A model for describing the atmospheric pressure, temperature and density variations with altitude is needed. A widely adopted and standardized model in use in the aviation industry is described in Int (1997).

$$T_{atm} = T_b + \beta\,(H - H_b) \quad \mathrm{K} \tag{3.12}$$

$$p_{atm} = p_b \left[1 + \frac{\beta}{T_b}\,(H - H_b)\right]^{-\frac{g_n}{\beta R}} \quad \mathrm{Pa} \tag{3.13}$$

$$v_s = \alpha_s \sqrt{T_{atm}} \quad \mathrm{m\,s^{-1}} \tag{3.14}$$

$$\rho_{atm} = \frac{p_{atm}}{R_{air} T_{atm}} \quad \mathrm{kg\,m^{-1}} \tag{3.15}$$

For the purpose of ground firing we are only concerned with atmospheric variations within the stratospheric layer where $T_b = 288.15\,\mathrm{K}$, $p_b = 101\,325\,\mathrm{Pa}$, $H_b = 0\,\mathrm{m}$, $g_n = 9.806\,65\,\mathrm{m\,s^{-2}}$, $R_{air} = 287.052\,87\,\mathrm{J\,kg^{-1}\,K}$ and the temperature gradient $\beta = 0.0065\,\mathrm{K\,m^{-1}}$. By combining 3.12 and 3.13 into 3.15 and approximating the altitude H with x_2 we get the following:

$$\rho_{atm} = \frac{p_b \left[1 + \frac{\beta}{T_b}\,(x_2 - H_b)\right]^{-\frac{g_n}{\beta R}}}{R_{air} T_b + \beta\,(x_2 - H_b)} \tag{3.16}$$

Aerodynamic force modeling

The ground-firing tables that the integrative ballistics model is to be fitted to is most likely based on the common way of modeling aerodynamic forces presented in the theoretic frame of reference, which is also why this way of aerodynamic force modeling is preferable for use in the integrative ballistics model. Most of the unknown ballistic parameters that will need to be fitted against tabular data is contained in the aerodynamic force models, mainly the different coefficient functions will need some means of model construction. It is not likely that the full aerodynamic models presented in the MPMTM can be used simply because this would need model reconstruction of several independent functions from one set of output data. As a first step of reduction the quadratic yaw of repose terms are omitted, doing so reduces the number of aerodynamic coefficient functions by half. The reasoning is that for a dynamically stable projectile the angle of yaw is somewhat small, otherwise the projectile would be over-stabilized or instable as mentioned in section 2.3.

$$\vec{F}_D = -\frac{\pi d^2 \rho}{8} C_{D_O} v \cdot \vec{v} \tag{3.17}$$

Although small, an approximation of the angle of yaw, or at least it's effects is preferable when modeling spin-stabilized projectiles due to the lift and Magnus forces imposing significant effects in the trajectory. The yaw of repose mentioned in section 2.4 is therefore used for approximation.

$$\vec{F}_L = \frac{\pi d^2 \rho}{8} C_{L_\alpha} v^2 \vec{\alpha}_e \tag{3.18}$$

$$\vec{F}_M = -\frac{\pi d^3 \rho}{8} C_{Mag} \left(\vec{\alpha}_e \times \vec{v}\right) \tag{3.19}$$

$$\vec{\alpha}_e = -\frac{8 I_x p}{\pi d^3 \rho C_{M_\alpha} \alpha_e^2 v^4} (\vec{v} \times \ddot{\vec{x}}) \tag{3.20}$$

$$\dot{p} = \frac{\pi \rho d^4 v C_{Spin}}{8 I_x} p \tag{3.21}$$

This external ballistics model will work as a start that may be improved upon in later sections when the model accuracy and parameterization has been evaluated properly.

3.3 Differential equation solution

The main equation of motion can be arranged as a system of second order differential equations:

$$\begin{aligned}\frac{d^2 x_1}{dt^2} = &-\frac{\pi d^2 \rho}{8} C_{D_O} v v_1 + \frac{\pi d^2 \rho}{8} C_{L_\alpha} v^2 \alpha_{e1} + \frac{\pi d^3 \rho}{8} C_{Mag} \left(\alpha_{e2} v_3 - \alpha_{e3} v_2\right) \\ &- 2\Omega \left[\sin(lat)\dot{x}_3 + \cos(lat)\sin(az)\dot{x}_2\right] - g_0 \frac{x_1}{R_z}\end{aligned} \tag{3.22}$$

$$\begin{aligned}\frac{d^2 x_2}{dt^2} = &-\frac{\pi d^2 \rho}{8} C_{D_O} v v_2 + \frac{\pi d^2 \rho}{8} C_{L_\alpha} v^2 \alpha_{e2} + \frac{\pi d^3 \rho}{8} C_{Mag} \left(\alpha_{e3} v_1 - \alpha_{e1} v_3\right) \\ &- 2\Omega \left[\cos(lat)\sin(az)\dot{x}_1 + \cos(lat)\cos(az)\dot{x}_3\right] - g_0 \left(1 - \frac{2x_2}{R_z}\right)\end{aligned} \tag{3.23}$$

$$\begin{aligned}\frac{d^2 x_3}{dt^2} = &-\frac{\pi d^2 \rho}{8} C_{D_O} v v_3 + \frac{\pi d^2 \rho}{8} C_{L_\alpha} v^2 \alpha_{e3} + \frac{\pi d^3 \rho}{8} C_{Mag} \left(\alpha_{e1} v_2 - \alpha_{e2} v_1\right) \\ &- 2\Omega \left[\cos(lat)\cos(az)\dot{x}_2 - \sin(az)\dot{x}_1\right] - g_0 \frac{x_3}{R_z}\end{aligned} \tag{3.24}$$

Where:

$$v_i = \frac{dx_i}{dt} - w_i \qquad i = 1, 2, 3 \qquad v = \sqrt{v_1^2 + v_2^2 + v_3^2} \tag{3.25}$$

$$\rho = \frac{p_b \left[1 + \frac{\beta}{T_b}\left(x_2 - H_b\right)\right]^{-\frac{g_n}{\beta R}}}{R_{air} T_b + \beta\left(x_2 - H_b\right)} \tag{3.26}$$

$$g_0 = g_n(1 - \alpha_g \cos(2lat)) \tag{3.27}$$

And the equation for yaw of repose as:

$$\alpha_{e1} = -\frac{8 I_x p \left(v_2 \frac{d^2 x_3}{dt^2} - v_3 \frac{d^2 x_2}{dt^2}\right)}{\rho \pi d^3 C_{M_\alpha} v^4} \tag{3.28}$$

$$\alpha_{e2} = -\frac{8 I_x p \left(v_3 \frac{d^2 x_1}{dt^2} - v_1 \frac{d^2 x_3}{dt^2}\right)}{\rho \pi d^3 C_{M_\alpha} v^4} \tag{3.29}$$

$$\alpha_{e3} = -\frac{8 I_x p \left(v_1 \frac{d^2 x_2}{dt^2} - v_2 \frac{d^2 x_1}{dt^2}\right)}{\rho \pi d^3 C_{M_\alpha} v^4} \tag{3.30}$$

The equation for the projectile spin rate around the projectile axis of symmetry:

$$\frac{dp}{dt} = \frac{\pi \rho d^4 v C_{Spin}}{8 I_x} p \tag{3.31}$$

Integration method

Depending on the stability of the non-linear differential equation an appropriate numerical method needs to be chosen. In a 6DOF model the yaw angle α of the projectile is a very oscillatory component as shown by Baranowski (2013c) due to the fast nutation and precession motion caused by gyroscopic forces for a spin-stabilized projectile. Typically oscillatory terms like this can cause numerical instability and can require a high computational intensity. On the contrary the Yaw of Repose introduced by Robert F. Lieske (1966) show no such behavior which suggests less computation requirements and numerical stability. The accumulated

or global truncation error over the integrated trajectory is preferable to be kept low, so a fourth order Runge-Kutta (RK4) is chosen as a basis for the step update.

The yaw of repose terms 3.28, 3.29 and 3.30 in equations 3.22, 3.23 and 3.24 are problematic to substitute into the differential equation because of their second order dependencies. A solution to this problem is to separate the two, solving the differential equation system for 3.28, 3.29, 3.30 and 3.31 with a fixed yaw of repose, then use this solution to calculate $\vec{\alpha}_e$ at the current time step, which is then used for the next time step and so on. To solve the differential equation $\mathbf{y}$ is defined:

$$\mathbf{y} = \begin{cases} x_1 \\ x_2 \\ x_3 \\ \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ p \end{cases} \tag{3.32}$$

The yaw of repose can then be written as a function $\mathbf{q}$ of $\mathbf{y}$ and $\mathbf{y}'$:

$$\vec{\alpha}_e = \mathbf{q}(\mathbf{y}', \mathbf{y}) = \begin{cases} \text{equation 3.28} \\ \text{equation 3.29} \\ \text{equation 3.30} \end{cases} \tag{3.33}$$

so that

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}, \mathbf{q}(\mathbf{y}', \mathbf{y})) = \begin{cases} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \text{equation 3.22} \\ \text{equation 3.23} \\ \text{equation 3.24} \\ \text{equation 3.31} \end{cases} \tag{3.34}$$

The step update with RK4 has the form:

$$\mathbf{y}_{i+1} = \mathbf{y}_i + \frac{h}{6}\left(\mathbf{f}_1 + 2\mathbf{f}_2 + 2\mathbf{f}_3 + \mathbf{f}_4\right), \quad \begin{cases} \mathbf{f}_1 = \mathbf{f}(t_i, \mathbf{y}_i, \mathbf{q}_i) \\ \mathbf{f}_2 = \mathbf{f}(t_i + \frac{h}{2}, \mathbf{y}_i + \mathbf{f}_1 \frac{h}{2}, \mathbf{q}_i) \\ \mathbf{f}_3 = \mathbf{f}(t_i + \frac{h}{2}, \mathbf{y}_i + \mathbf{f}_2 \frac{h}{2}, \mathbf{q}_i) \\ \mathbf{f}_4 = \mathbf{f}(t_i + h, \mathbf{y}_i + \mathbf{f}_3 h, \mathbf{q}_i) \end{cases} \tag{3.35}$$

$$t_{i+1} = t_i + h \tag{3.36}$$

$$\mathbf{y}'_{i+1} = \mathbf{f}(t_{i+1}, \mathbf{y}_{i+1}, \mathbf{q}_i) \tag{3.37}$$

$$\mathbf{q}_{i+1} = \mathbf{q}(\mathbf{y}'_{i+1}, \mathbf{y}_{i+1}) \tag{3.38}$$

3.4 Iterative boundary value solver and adaptive step length algorithm

The model parameters d, I_x, p_b, g_n, β, T_b, R_{air}, H_b, Ω and R_z are constant at all times and are assumed to be known. The conditions az, lat, w_1, w_2 and w_3 are fixed throughout the trajectory and assumed to be known at time of firing. The aerodynamic coefficients C_{Spin}, C_{Mag}, C_{M_α} and C_{D_0} are all assumed functions of the Mach number. The initial conditions of the projectile states are as follows:

$$q_0 = \begin{cases} \alpha_{e1_0} = 0 \\ \alpha_{e2_0} = 0 \\ \alpha_{e3_0} = 0 \end{cases} \tag{3.39}$$

$$y_0 = \begin{cases} x_{1_0} = 0 \\ x_{2_0} = 0 \\ x_{3_0} = 0 \\ \dot{x}_{1_0} = \dot{x}_0 \cos(QE)\cos(SA) \\ \dot{x}_{2_0} = \dot{x}_0 \sin(QE) \\ \dot{x}_{3_0} = \dot{x}_0 \cos(QE)\sin(SA) \\ P_0 = P_{ms} \end{cases} \tag{3.40}$$

where $\dot{x}_0$ is the muzzle velocity and P_{ms} is the initial spin and is assumed known at time of fire. Previously mentioned in section 3.1, the task for the FCS is to determine the ballistic deviation angles SE and SA so that it has the highest probability of hitting the target. In the FCS the current position of the target to hit is given as an estimate to the ballistics module, these coordinates of the target position is transformed into polar coordinates and the external ballistics reference coordinate system orientation is defined so the target lays in the 1-2 plane. The target location is as follows:

$$\begin{aligned} x_{1t} &= R_t \cos(EL) \\ x_{2t} &= R_t \sin(EL) \\ x_{3t} &= 0 \end{aligned} \tag{3.41}$$

Where R_t is the range to target. To begin with the target is assumed stationary so the task is to find the initial values of SE and SA so that the projectile ends up

as close to the position of the target as possible. This introduces another problem, the step length must be adapted so that the integrator stops iterating at the right time, there needs to exist a boundary value for the integrator. Figure 3.3 shows an example of the projectile position vector at integration step i. Since the integrator is to run in a hard real-time environment there are also time constraints, that means there needs to be some kind of control over the amount of integration steps needed to finish integrating the trajectory.

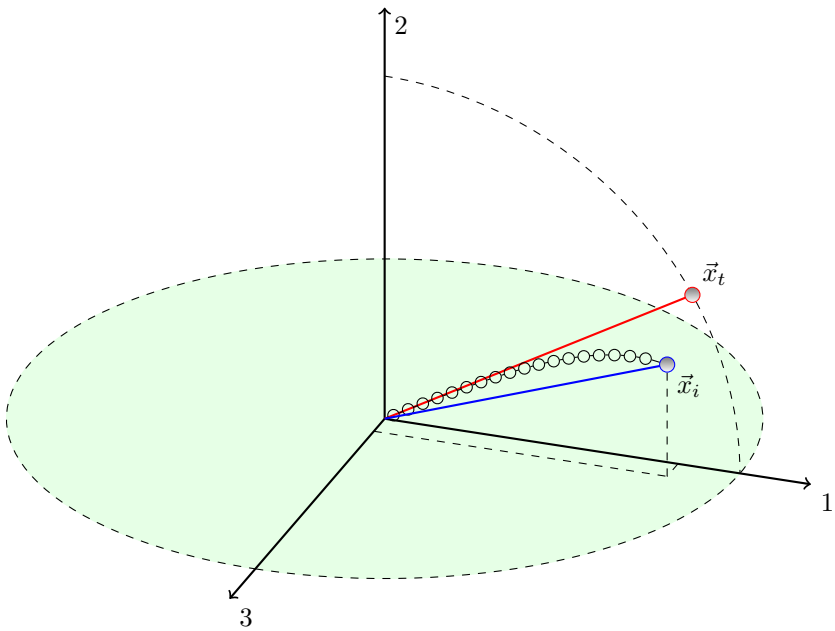


Figure 3.3: Target position vector $\vec{x}_t$ and projectile position vector $\vec{x}_i$ at integration step i fired with the initial conditions $SE = SA = 0$

A solution to the boundary value is to immediately stop the integration when the absolute radial distance to the projectile x_i is equal to or above the radial distance to the target x_t or as called previously R_t. The integration boundary is then:

$$\sqrt{x_1^2 + x_2^2 + x_3^2} \leq R_t \tag{3.42}$$

The integrator should not integrate further than this radial boundary, but it is desirable to end up close to the radial boundary. An adaptive step length algorithm so that the number of integration steps are sufficiently predictable and the last integration step is taken so that the projectile end up very close to the boundary

value is needed. The radial velocity of the projectile at integration step i is the scalar product between the projectile velocity $\dot{\vec{x}}$ and the radial direction:

$$\dot{\vec{R}}_i = \dot{\vec{x}}_i \bullet \frac{\vec{x_i}}{\sqrt{x_{1i}^2 + x_{2i}^2 + x_{3i}^2}} \tag{3.43}$$

And the radial speed:

$$\dot{R}_i = \sqrt{\dot{R}_{1i}^2 + \dot{R}_{2i}^2 + \dot{R}_{3i}^2} \tag{3.44}$$

The radial distance at the next integration step can then be estimated as:

$$\hat{x}_{i+1} = x_i + \dot{R}_i h \tag{3.45}$$

Normally for very long trajectories fired at high quadrant elevation angles the projectile speed can reach a minimum at the top of the trajectory arc and increase as the projectile falls to the ground, however for the targeted platform these trajectories are not considered since there are range limitations. An assumption that within the range limitations the projectile will always drop in speed is made. Further an assumption is made that the radial speed then also always drops, this assumption should work well as long as the difference in direction of the radial velocity vector and the projectile velocity vector is not to big, which is the case unless firing at very long ranges. If this assumption holds true, then implicitly the estimation should end up a bit short:

$$\hat{x}_{i+1} < x_{i+1} \tag{3.46}$$

The targeted number of integration steps is defined as n_s, a step length algorithm is written as:

$$h = \frac{R_t}{n_s \dot{R}_i} \tag{3.47}$$

Taking the previous assumption into account this algorithm takes approximately radial equidistant steps and after n_s steps the radial distance should be close to but less then R_t. The step length algorithm is used for this purpose, then the last few steps are taken so that the projectile should reach very close to the radial boundary value:

$$h = \frac{(R_t - R_i)}{\dot{R}_i} \tag{3.48}$$

The proposed step length algorithm with the boundary values is implemented so that:

$$h = \begin{cases} \frac{R_t}{n_s R_i} & \text{if } (R_t - R_i) \geq \frac{R_i}{n_s} \\ \frac{(R_t - R_i)}{R_i} & \text{while } R_i \leq R_t - \epsilon_r R_t \end{cases} \tag{3.49}$$

The step length algorithm should approximately take n_s steps in the first condition and end up close to the target, then the last few steps are taken so that the projectile ends up approximately equal radial distance to that of the target when stopping the integration. The term $\epsilon_r R_t$ is an error margin that is proportional to the distance to the target, if previous assumptions are true this step length algorithm will always keep the trajectory within the radial boundary but never reach it, so a small error margin must be allowed.

Assuming the step length algorithm works, the projectile end-point position x_n where it is approximately equal to that of R_t is found and the deviation angles from the target to the projectile end location can be calculated through inverse polar coordinates transformation.

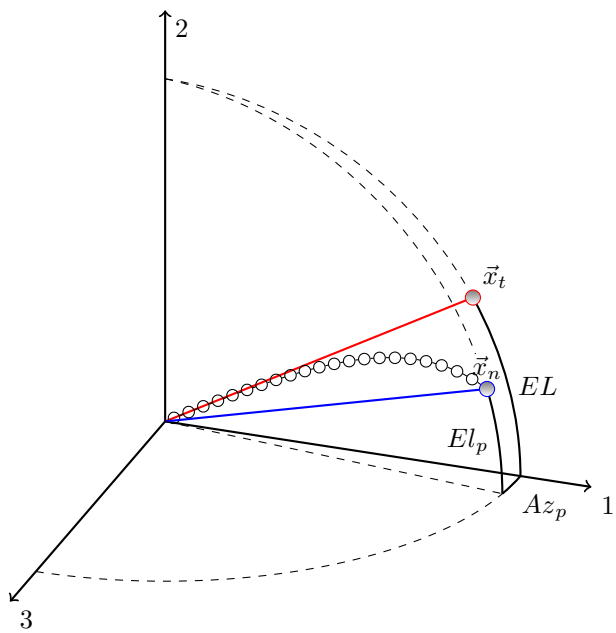


Figure 3.4: Projectile position vector $\vec{x}_n$ after last integration step and the polar angles.

The difference in polar angles between the projectile end point and the target position:

$$\delta El = EL - El_p \tag{3.50}$$
$$\delta Az = 0 - Az_p \tag{3.51}$$

A method for minimizing this error will solve the problem. If the elevation angle to the target (EL) is seen as the reference signal, the deviation angles to the projectile position at the radial boundary (El_p,Az_p) are seen as a system output and the error signal are the differences (δEl,δAz), a prediction method similar to that of a discrete time controller using the integrated sum can be used as shown in figure 3.5.

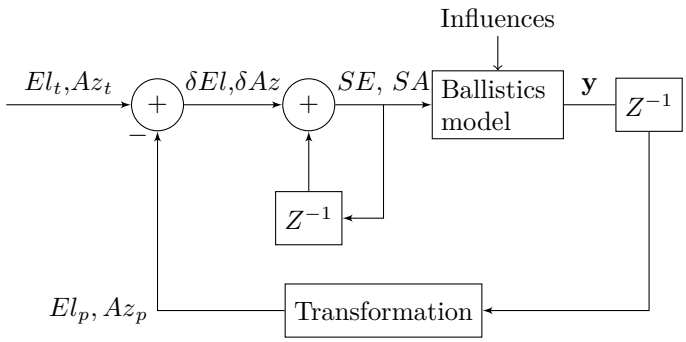


Figure 3.5: The initial value predictor solution similar to that of a discrete time controller

The controller is simply implemented as:

$$SE = SE_{k-1} + \delta El \tag{3.52}$$
$$SA = SA_{k-1} + \delta Az \tag{3.53}$$

Where SE_{k-1} and SA_{k-1} are the conditions used to find El_p and Az_p, initial conditions are set to $SE_0 = SA_0 = 0$ which corresponds to firing directly at the target. This prediction controller is cyclically executed and runs in iterations indefinitely in the target platform or for evaluation purposes until the error is small enough:

$$\delta Az < \epsilon_a \tag{3.54}$$
$$\delta El < \epsilon_a \tag{3.55}$$

Where ϵ_a is a small error threshold for the deviation angles.

3.5 Moving target

So far the iterative boundary value solver only concerns finding the initial condition for the ballistic deviation angles, however if the target is moving the projectile would miss completely because of the time it takes for the projectile to reach the target. The actual FCS utilizes several sensors to not only detect the range to the target but also estimate the target movement. The problem is defined as with a target at position $\vec{x}_t$ and traveling in a straight line with velocity $\dot{\vec{x}}_t$ predict a impact point $\vec{x}_{poi}$ where the target and projectile meets.

$$\vec{x}_{poi} = \vec{x}_t + t\dot{\vec{x}}_t \tag{3.56}$$

The estimated target movement and position are for the purpose of this thesis considered known variables, which leaves us with determining the time t, which is the time of flight of the projectile to the target t_{fl} as well as a delay compensation t_{del}. For the targeted platform t_{del} is constant because the system runs in known cycle frequencies, it should consider how old the data from the target movement estimator is, how long time it takes for the actual ballistics calculation and the actuation time of the gun servos as well as any signal propagation and filtering delays.

A solution is to use the controller developed for the static target but loop back the time of flight through the prediction and try to hit a predicted impact point instead.

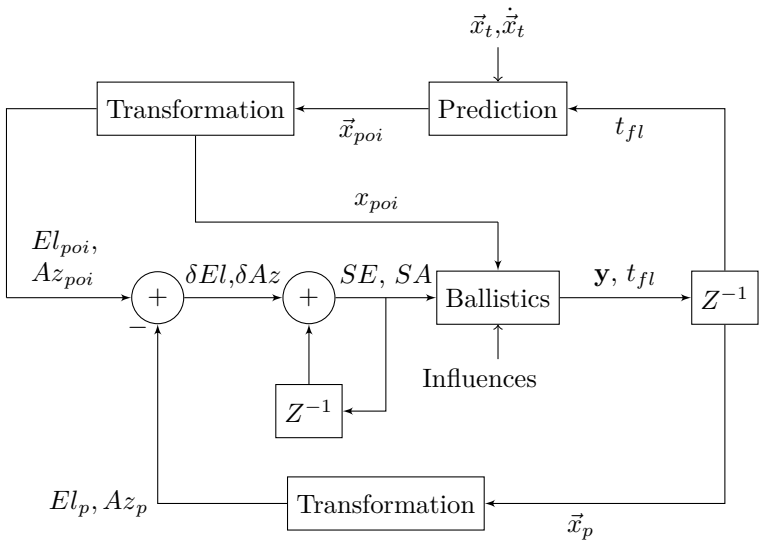


Figure 3.6: The extended prediction controller to solve the dual boundary value problem of moving target.

$$\vec{x}_{poi_k} = \vec{x}_{t_k} + \left(t_{fl_{k-1}} + t_{calc} \right) \dot{\vec{x}}_{t_k} \tag{3.57}$$

Chapter 4

CONCEPT VALIDATION

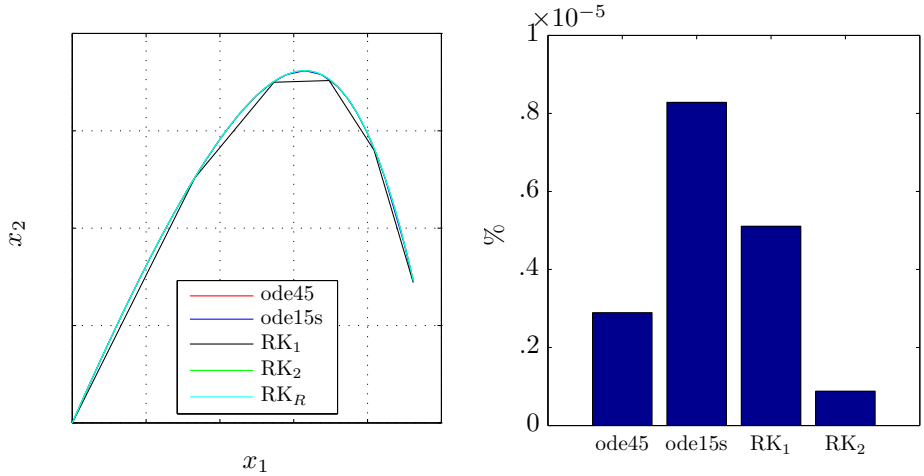
The external ballistics model and the methods presented in chapter 3 needs to be assessed, this is done by implementing the external ballistics model as well as the boundary value solver in MATLAB® to be evaluated. The ballistic parameters are taken from a projectile with known parameters that can be used for validating the external ballistics model. The projectile used is a spin-stabilized tracer round.

4.1 Integrator verification

The ballistics model is solved in MATLAB® to evaluate how the step length and the numerical integration method affects the results. Different initial conditions and boundary values were tested, presented in figure 4.1 is a worst case where the projectile was fired a high angle and integrated to a set time point. Fourth order Runge-Kutta with a fixed and very small time step is used as a reference trajectory (RK_R). This is compared to MATLAB® ode45, ode15s as well as the same method with with varying amount of steps. Figure 4.1b shows the relative distance error at the projectile end-point against the reference model in percent, calculated as:

$$\frac{\|\vec{x}_{n_R} - \vec{x_n}\|}{100\|\vec{x}_R\|} \tag{4.1}$$

Where $\vec{x}_R$ is the reference trajectory projectile end-point and $\vec{x}$ is the compared trajectory projectile end-point. The term is a relative accumulated truncation error measure in percent considering RK_R as the true trajectory.



(a) Resulting trajectories, extreme range

(b) Variation relative RK_R

Figure 4.1: The difference in trajectory with varying step sizes and methods. RK_1, RK_2 and RK_R are fourth order Runge-Kutta integrators with a total of 5, 50 and 1×10^6 integration steps respectively. In (a) the trajectory shapes are shown while in (b) the relative difference of the end point towards the reference RK_R is shown in percent according to equation 4.1.

4.2 First model concept evaluation

All the different projectile states and forces can be evaluated at each integration step, conclusions can be made about their significance, direction and effects on the trajectory. The model can be evaluated in accuracy by some comparisons to the ground firing table. The longest trajectory available from the firing-tables is a 7000 m trajectory and was chosen as evaluation. The ballistic deviation angles was set accordingly, the trajectory was then integrated to range and the resulting end-point was compared to that of the ground-firing table, shown in figure 4.2.

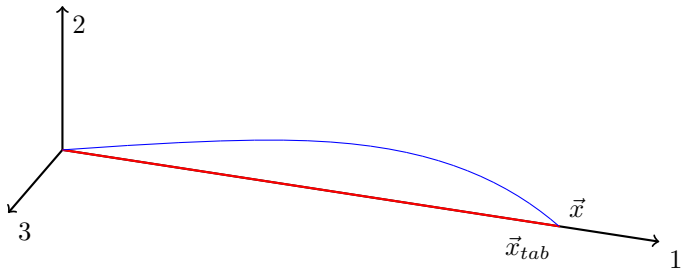
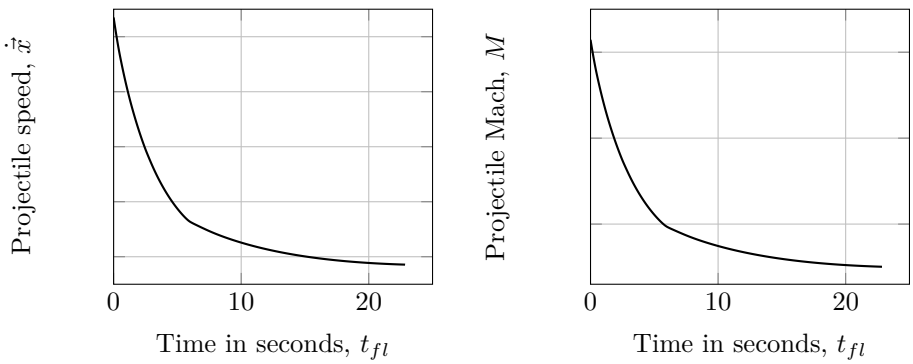


Figure 4.2: Actual trajectory calculated from MATLAB® using the proposed RK4 solver. The difference in tabular end-point $\vec{x}_{tab}$ and trajectory end-point $\vec{x}$ is smaller then the picture can show, however the relative error is calculated to 2.4×10^{-5} % using equation 4.1.

For this trajectory the projectile states over time can be evaluated. In figure 4.3 the velocity and Mach number are shown over the trajectory the flight time.

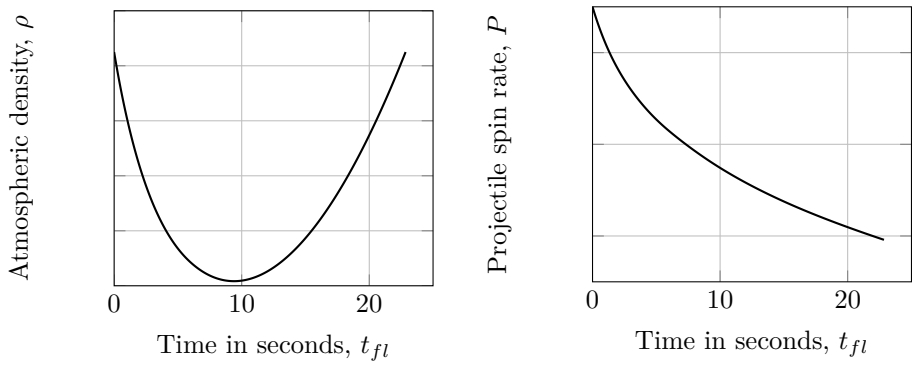


(a) Projectile speed over time.

(b) Projectile Mach over time.

Figure 4.3: The variation in Mach and speed over flight time. The curves are slightly different because of density variations over the trajectory. At around 6 s the drag resistance drops slightly due to transition from supersonic to subsonic speeds.

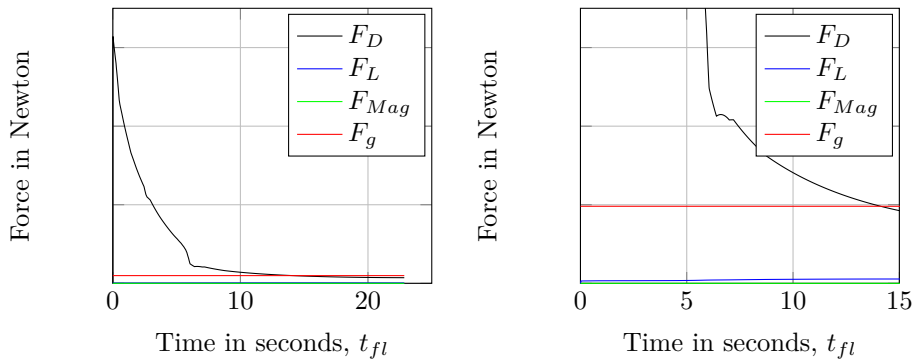
The atmospheric density varies with the projectile trajectory in accordance to the altitude of the projectile, in figure 4.4a the density variation throughout the projectile trajectory is shown as a function of time and in figure 4.4b the spin damping over the trajectory can be seen.



(a) Air density over time.

(b) Projectile spin rate over time.

Figure 4.4: The variation in atmospheric density at the projectile location and projectile spin rate.

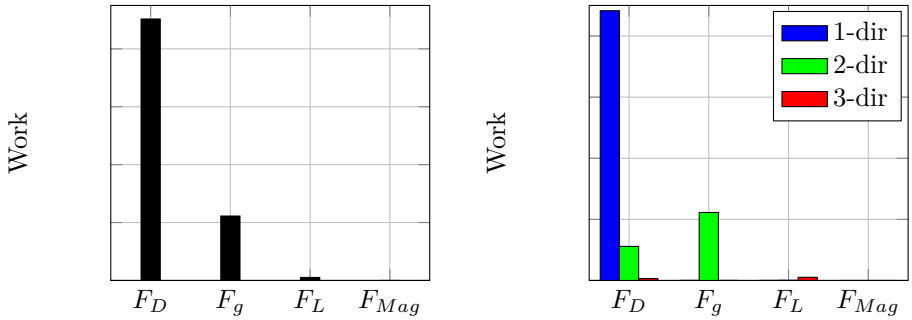


(a) Force magnitudes over time.

(b) Force magnitudes over time, zoomed in.

Figure 4.5: The Force magnitudes over time, clearly, the the drag force F_D is very important to model accurately. The effects of transition from supersonic to subsonic speeds can be seen in the drag force. The Magnus force F_{Mag} is however very small and can potentially be omitted from the model.

From the forces it is possible to assess the total work split by each force in magnitude and also in direction which can be used to tell how important each part of the external ballistics trajectory model is, so the model can be reduced further and assumptions needed for depicting suitable methods in the parameterization can be made with this in mind.

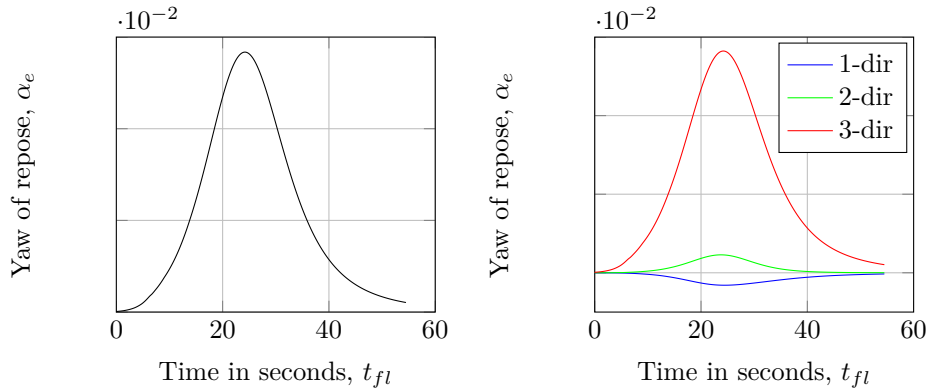


(a) Total work over trajectory. (b) Total work over trajectory, components.

Figure 4.6: The work for each force and the work for each force split by components. Looking at the direction components, the lift force has a small component to the side in the 3-direction, meaning the projectile is veering to the side by a small amount.

From the force and work magnitudes it is clearly shown that the major imparting force is the drag force. The Magnus force however seems to be more or less negligible for this type of trajectory modeling. This can be explained from the theoretic frame of reference, Baranowski (2013c) shows that the Magnus effect imparts moments that account for projectile stabilization for spin-stabilized projectiles, however Lieskes yaw of repose does not consider the stabilization effects, so the remaining Magnus force seem to have little effect of the overall trajectory. This means that the aerodynamic coefficient relating to the Magnus force, C_{Mag}, is likely to be very hard to parameterize from tabular data where only a few trajectory end-points are available. Omitting the Magnus force will therefore likely reduce the model complexity without effecting the trajectory model noticeably. The Lift force is small, but is kept because it is mainly responsible for the projectile drifting to the side, as can be seen in figure 4.6b.

The yaw of repose α_e is also interesting, it is possible to determine how this yaw angle approximation affects the trajectory by looking at the magnitude and directions. By solving a trajectory with a high quadrant elevation QE large yaw of repose is achieved. The yaw of repose can also be compared to the frame of reference where Baranowski (2013b) shows how the yaw of repose approximates the actual yaw angle in a 6DOF model.



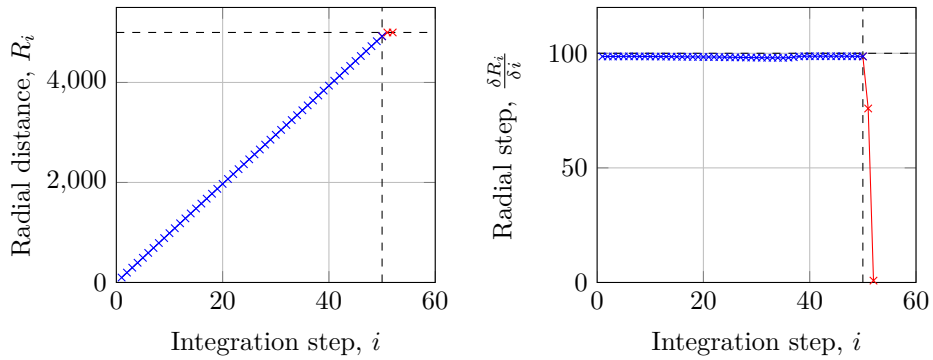
(a) Yaw of repose magnitude over time. (b) Yaw of repose over time, components

Figure 4.7: The Yaw of repose α_e is shown for a trajectory with a steep quadrant elevation $QE = 45°$. The Yaw of repose shows that the projectile is yawing in the 3-direction, in reality this happens because of gyroscopic moments due to the arcing trajectory and spin of the projectile. The effect is that a spin-stabilized projectile curves slightly to the side, something known as drift.

From these results the model can simply be reduced to not include the Magnus force F_{Mag}. It is also clear that the drag coefficient C_D plays a major part in determining the trajectory, meaning if a projectile is to be parameterized from firing-tables, a good start is to focus on finding a model for the drag coefficient with good fitting. The lift coefficient C_L is important for a proper inclusion of the responsible forces causing projectile drift of spin-stabilized projectiles. Atmospheric modeling also plays a significant role, especially for firing at aerial targets because of how the speed of sound varies in the atmosphere, effecting all the different aerodynamic coefficients as they are dependent on the Mach number.

4.3 Initial value solution and numerical step length algorithm verification

The numerical integrator step length algorithm was developed with certain assumptions, these can be tested for varying conditions the see if they hold true, it is possible to see how well the step length algorithm finds the radial boundary. With a specific software design for the final system in mind at the time, the numerical integrator would have to find a solution in at most 64 calculation cycles, so the integrator is tested with a desired n_s of 50 steps as to allow for some slack, then the difference between projectile and target radial distance is evaluated at n_s steps and the number of additional steps needed to reach the radial boundary value is evaluated.



(a) Radial distance to projectile over integration step.

(b) Resulting radial step length over integration step, derivative of figure (a)

Figure 4.8: The step length algorithm for a very long range trajectory of 7000 m is evaluated. Integration steps taken with a step length calculated using equation 3.47 is shown in blue and 3.48 is shown in red. Targeted number of steps n_s is shown as a vertical line and the sought radial boundary is shown as a horizontal line in figure (a) and the horizontal line in figure (b) is $\frac{R_t}{n_s}$.

Figure 4.8 shows the algorithm working well for a worst case, finding the boundary within n_s+2 steps with an error margin $\epsilon_r < 1 \times 10^{-6}$ (equivalent to < 1 mm at a radial distance of 1000 m). Figure 4.8b shows that the assumption holds true over all steps taken, the actual radial step distances are all slightly smaller then $\frac{R_t}{n_s}$, so that after n_s steps, $\epsilon_r = 0.015344$. Another 2 steps with step length algorithm 3.48 is enough to find the radial boundary within the sough error margin $\epsilon_r < 1 \times 10^{-6}$. The method works very well to find the projectile position at the radial boundary for all ranges up to and including the longest ranges available in firing tables.

The iterative method for determining the initial values SE and SA so that the projectile ends up at the impact point location is also evaluated. A trajectory is solved to the radial boundary with an initial guess of $SE_0 = 0$ and $SA_0 = 0$ and then corrected according to the proposed method in chapter 3. The worst case is a trajectory with as much arc as possible, so the method is tested for the 7000 m maximum range trajectory.

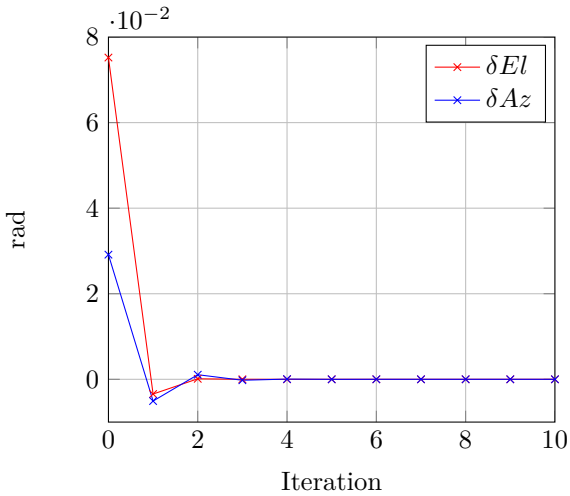


Figure 4.9: The solution converges quickly, after only 3 iterations the condition is $\delta Az, \delta El < \epsilon_a = 1 \times 10^{-4}$ rad is fulfilled

The solution converges very quickly, after 3 trajectory iterations the condition is fulfilled. The sought initial values for the ballistic deviation angles SE and SA that converges the projectile end-point and the impact point location is found within a small number of iterations using a integration method with a predictable number of calculations, which makes it feasible to use in FCS with hard real-time constraints.

4.4 Moving target evaluation

The prediction controller works well for finding a static reference, that is if the target position $\vec{x}_t$ and velocity $\dot{\vec{x}}_t$ given by the estimator is fixed. Figure 4.10, 4.11 and 4.12 shows the initial, second and third cycle of the controller for an aerial target at a position and velocity:

$$\vec{x}_t = \begin{pmatrix} 4330.12 \\ 2500 \\ 0 \end{pmatrix} \tag{4.2}$$

$$\dot{\vec{x}}_t = \begin{pmatrix} -45 \\ -25 \\ -5 \end{pmatrix} \tag{4.3}$$

Which corresponds to a target at distance $R_t = 5000\,\text{m}$ and elevation angle $EL = 30°$ moving towards the FCS and sideways to the left. The initial conditions time of flight and ballistic deviation angles are set to:

$$t_{fl} = \frac{v_m}{R_t} \tag{4.4}$$

$$SE_0 = SA_0 = 0 \tag{4.5}$$

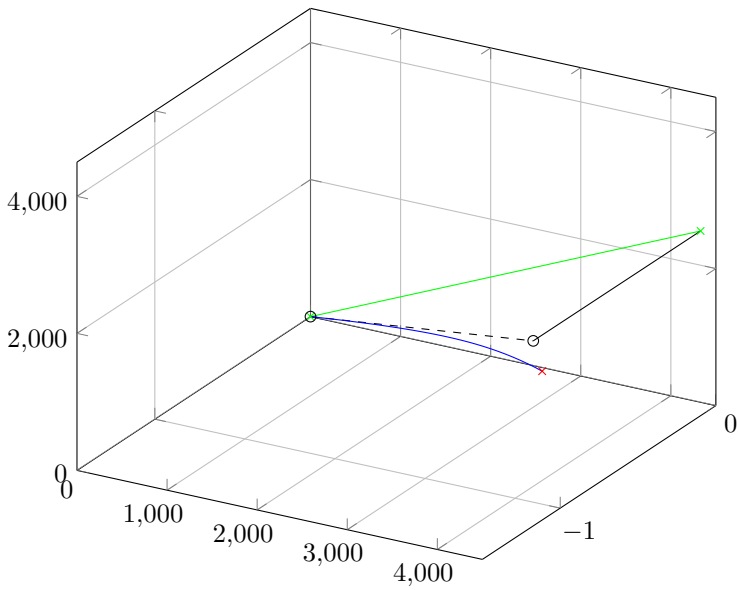


Figure 4.10: The initial cycle iteration, the black line shows the target movement and the black circle is the predicted target location after the flight time of the projectile t_{fl} (impact point) calculated using equation 3.56. The blue line and red cross shows the integrated trajectory and projectile end point, the green line and cross is the sight-line and target location.

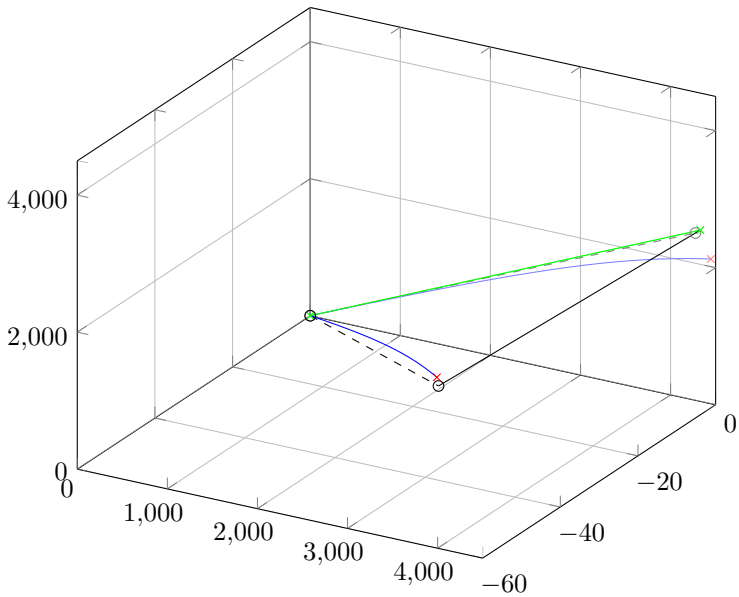


Figure 4.11: The second cycle iteration

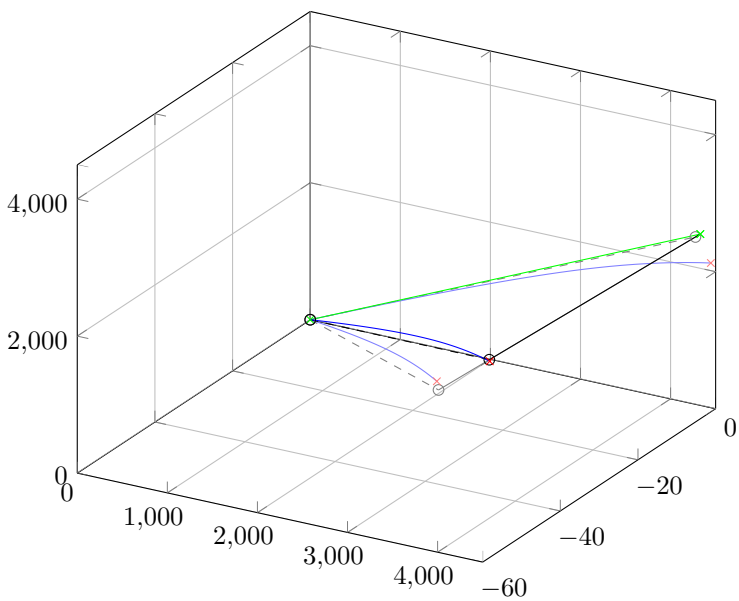
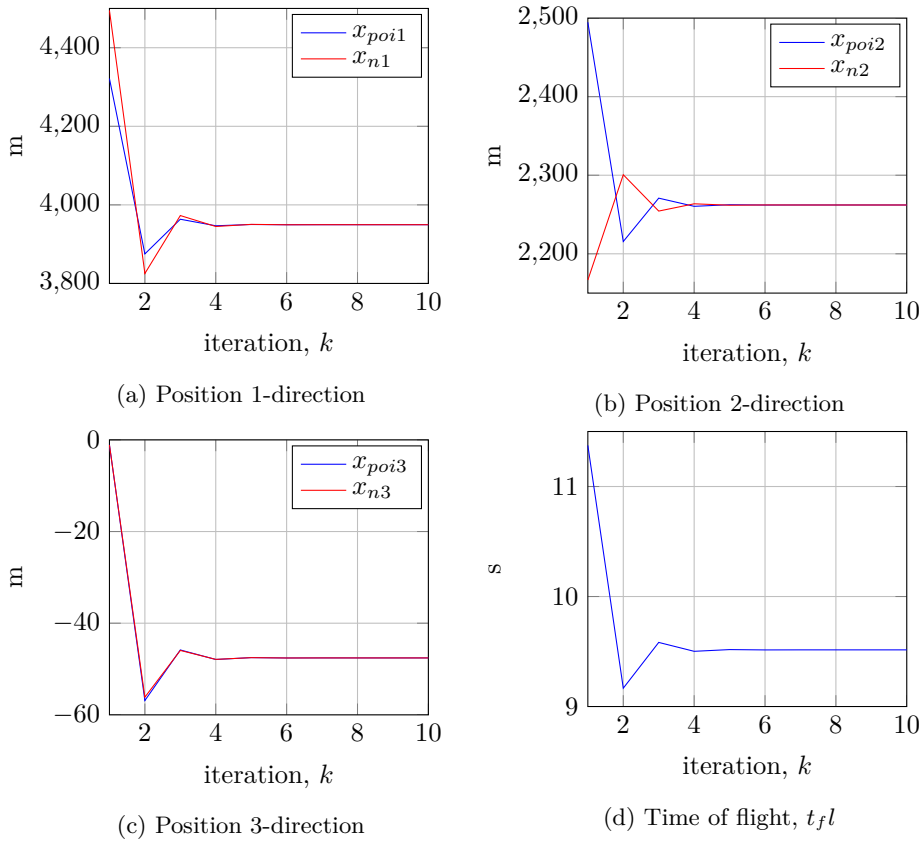


Figure 4.12: The third cycle iteration

The prediction controller is run until the distance between the projectile endpoint and predicted target location is less then 1 mm and the difference in t_{fl} between two iterations is less then 1 ms. The results are shown in figure 4.13.



(a) Position 1-direction

(b) Position 2-direction

(c) Position 3-direction

(d) Time of flight, t_fl

Figure 4.13: Moving target, static reference

The prediction controller finds a solution to the dual boundary value problem after a few iterations, however a static reference has been used, the reference target velocity or position from the estimator did not move between iterations. In the embedded system each calculation cycle takes time, which means that the reference target position moves during the calculation cycle. Each time a controller cycle has finished, a new reference position of the current target position will be fed to the predictor from the estimator. Since the target is assumed to move on a straight line, the new location in a ground fixed coordinate after one cycle is at a position:

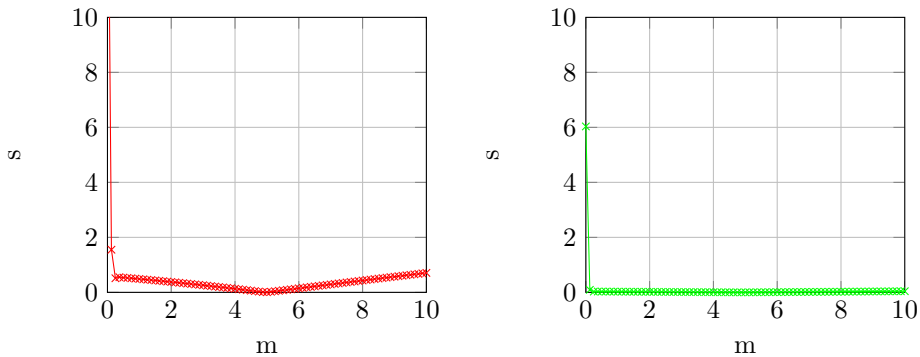
$$\vec{x}_{t_k} = \vec{x}_{t_{k-1}} + t_s \dot{\vec{x}}_{t_{k-1}} \tag{4.6}$$

Where t_s is the cycle time of the predictor calculation module. A simulated real-time test case was developed with the reference target position moving according to equation 4.6 for each iteration cycle during 10 s with a t_s of 0.1 s. This builds a vector of reference target positions and velocities along a straight line in the ground fixed coordinate system. To find the actual solution the controller was first allowed enough iterations to solve the problem of finding $\vec{x}_{poi}$ at each position along this movement and velocity reference vector, then the prediction controller was run in simulated real-time where after each new iteration the target moves corresponding to one cycle time. In this case the worst results were given when the difference in target location and velocity between two cycle iterations was large, a fast moving target at close range will produce this. The initial reference target position and velocity was set to:

$$\vec{x}_t = \begin{pmatrix} 985 \\ 174 \\ 500 \end{pmatrix} \tag{4.7}$$

$$\dot{\vec{x}}_t = \begin{pmatrix} -5 \\ -10 \\ -95 \end{pmatrix} \tag{4.8}$$

The target was set to move in a straight line according to 4.6 during 10 s with $t_s = 0.1$ s. Figure 4.14a shows the normalized distance between the estimated target location and resulting projectile location, as well as the normalized distance between the actual reference target location and the resulting projectile location when running in simulated real-time.



(a) Miss to actual impact point

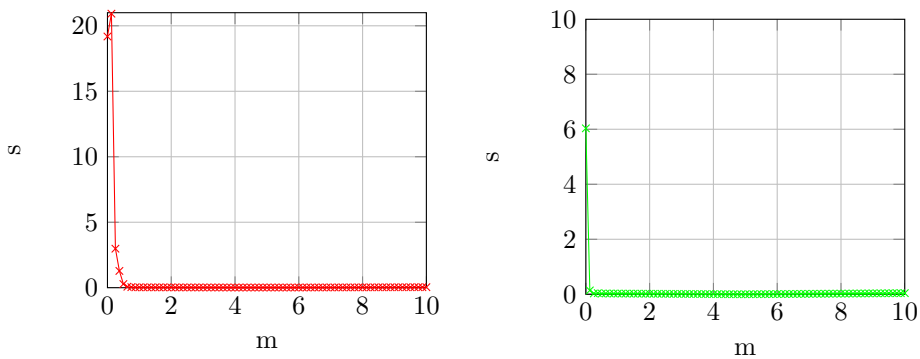
(b) Miss to predicted impact point

Figure 4.14: Normalized distance between the impact point location and resulting projectile location (Miss). The predicted impact point is hit (b), but actual reference impact point differs from the predicted (a).

While the projectile seems to hit its mark, it turns out that the predicted impact point location where the target and projectile meets is wrong, the reason being the t_{fl} used in the impact point prediction is delayed by one cycle time. The derivative of t_{fl} is approximated by backwards difference, this term is added for a better prediction of what the actual current t_{fl} is so a better prediction of the impact location is achieved, the resulting prediction becomes:

$$\vec{x}_{poi_k} = \vec{x}_{t_k} + \left(2t_{fl_{k-1}} - t_{fl_{k-2}} + t_{calc}\right)\dot{\vec{x}}_{t_k} \tag{4.9}$$

Figure 4.15 shows how this works with a initial value of $t_{fl_{k-2}} = t_{fl_{k-1}} = \frac{\dot{x}_0}{x_t}$.



(a) Miss to actual impact

(b) Miss to predicted impact

Figure 4.15: Normalized distance between the impact point location and resulting projectile location (Miss). The predicted impact point is hit (b) as well as the actual impact location (a)

Chapter 5

PARAMETERIZATION AGAINST TABULAR DATA

So far the external ballistics model has been evaluated using known aerodynamic coefficients and projectile properties, however the main purpose of this external ballistics model is not to accurately model a live trajectory, but to achieve high fitness against tabular firing data from ammunition manufacturers based on the manufacturers ballistics model. The model will need to be parameterized against firing-tables for projectiles with unknown aerodynamic coefficients and parameters. Because of reasons stated in section 4.2 the Magnus force is simply omitted from the external ballistics model. Table 5.1 recaps all the parameters in the external ballistic model and shows which parameters that are known and which parameters that needs parameterization from tabular data. asdasd

Name	Description	Known	Type and dependencies
g_n	Acceleration constant in gravitational model	Yes	Constant
α_g	Gravitational model parameter	Yes	Constant
p_b	Nominal air pressure	Yes	Constant
T_b	Nominal air temperature	Yes	Constant
β	Atmospheric temperature gradient	Yes	Constant
R_{air}	Individual gas constant for air	Yes	Constant
H_b	Atmospheric parameter	Yes	Constant
Ω	Angular speed Earth	Yes	Constant
R_z	Radius of Earth	Yes	Constant
d	Projectile diameter	Supplied by tabular data	Constant
$\dot{x}_0$	Projectile muzzle speed	Supplied by tabular data	Initial value
P_0	Projectile initial spin	Unknown	Initial value
C_D	Aerodynamic drag coefficient	Unknown	Function of Mach
C_L	Aerodynamic lift coefficient	Unknown	Function of Mach
C_{Spin}	Aerodynamic spin coefficient	Unknown	Function of Mach

Table 5.1: List of parameters in the external ballistics model

The model is reduced to finding aerodynamic coefficients for the projectile as well as the initial spin P_0. The available data for fitting in form of firing-tables is the ballistic offsets SE and SA, the time of flight t_{fl} and the terminal projectile speed at point of impact $\dot{x}_p$ for a set of trajectories with corresponding range to target R_i ,$i = \{1, 2...N\}$. This is a rather non-trivial problem as there is no solution that is obviously the correct one. Because this is not a typical signal processing problem, for there is no physical plant or otherwise method for actual measurements and the only data available are a few trajectory end-points for a set of ranges, typical estimation methods might be hard to implement. Instead, assumptions and model restrictions themselves will depict what form of method that can be chosen.

A first assumption is that the underlying external ballistics model used to produce the ground-firing tables by the munitions manufacturer is a 6DOF model, where the aerodynamic coefficient is dependent on the Mach number only. This is known to be true by the author in several cases.

Finding C_D as a function of M

For every range R_i in the ground fire table there is an underlying trajectory model with known boundary values. Since the initial projectile speed $\dot{x}_0$, the projectile speed at point of impact $\dot{x}_{poi}$ and the atmospheric conditions at these points are known from tabular data, each trajectory corresponds to a certain region of Mach M. For instance the ground-firing table may state that the terminal speed at the point of impact is $\dot{x}_{p1}$ for the corresponding trajectory to a target at range R_1, that trajectory then holds some information on how the projectile is retarded within the region $\dot{x}_0$ to $\dot{x}_1$ which can be translated to M_0 to M_1. Consequentially the tabular data for the trajectory range R_2 contains information on how the projectile is retarded within the region M_{mv} to M_2 and so on.

From previous results it was shown that the drag force is the main acting force on the projectile. If an assumption is made that the *aerodynamic* retardation of the projectile is only dependent on the drag force and the traveled distance for the projectile can be correctly modeled by solving the trajectory using methods in chapter 3, a method for finding a mean C_D value can be defined for each trajectory data in the ground firing table. The problem can be described as to for each trajectory in the ground-firing table, find C_D that minimize the error in terminal speed $\epsilon_{vt} = \dot{x}_{poi} - \dot{x}_{tab}$. Starting with the shortest trajectory in the tabular data available, a constant C_{D_1} is found using a steepest descent method. This constant is now used as the function of Mach in this region; $C_D = C_{D1}, \quad \forall M \in [M_0, M_1]$. The second trajectory contains information of the retardation in the Mach region M_0 to M_2, which includes the previous Mach region. The model for the aerodynamic drag coefficient derived from the previous trajectory is used within its corresponding region, for the remaining region M_1 to M_2 a linear interpolation is used as model to find a point C_{D2} using a steepest descent method. The final model for the aerodynamic drag coefficient C_D as a function of M is then:

$$C_D = \begin{cases} C_{D_1} & ,\forall M \in [M_{mv}, M_1] \\ C_{D_1} + (C_{D_2} - C_{D_1}) \frac{M - M_1}{M_2 - M_1} & ,\forall M \in [M_1, M_2] \\ ... & \\ C_{D_{N-1}} + \left(C_{D_N} - C_{D_{N-1}}\right) \frac{M - M_{N-1}}{M_i - M_{N-1}} & ,\forall M \in [M_{N-1}, M_N] \end{cases} \tag{5.1}$$

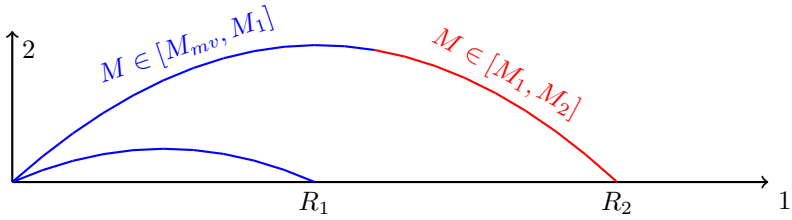


Figure 5.1: Trajectories for a target at range R_1 and R_2, the terminal Mach number M_1 for the first trajectory shown in only blue is given by the tabular data. This trajectory then contains information on how the projectile was retarded within that Mach region (blue). The next trajectory contains information on how the projectile was retarded within both the blue and red Mach region.

This process is repeated for all available data in the ground-firing table and if the method works well the result is a list of points from which the aerodynamic drag coefficient C_D is linearly interpolated from.

The parameterization problem of C_L, P_0, C_{Spin} and the final model reduction

So far the method used to model the drag coefficient minimizes the error in terminal speed v_t against tabular data. However the ballistic deviation angles SE and SA also needs to match the tabular data. The aerodynamic lift force was temporarily omitted to find C_D, however for nominal atmospheric conditions the Lift force is the only force in the current nonlinear external ballistics model with a component in the 3-direction, therefore it is the only force to cause the projectile drift. This means that in the same way that the aerodynamic drag coefficient was found, it is possible to try to find an function for the aerodynamic lift coefficient C_L from the tabular data for the SA for each trajectory. Note that for projectiles that are not spin-stabilized, the lift force has no magnitude because the projectile has no spin as can bee seen in equations 3.28 to 3.31 with $p = 0$. This means that fin-stabilized projectiles do not have drift under nominal atmospheric conditions and for such projectiles C_D is the only aerodynamic drag coefficient in the model that needs to be parameterized.

For spin-stabilized projectiles however, finding C_L as a function of the Mach number turns out to be problematic with the current external ballistics model. This is because the Lift force depends not only on the aerodynamic lift coefficient C_L, but also the yaw of repose $\vec{\alpha_e}$ which in terms depends on the projectile spin rate P which is a function of the aerodynamic spin-damping coefficient C_{Spin} and has an unknown initial spin rate P_0. Some attempts were made to parameterizing these, however they all resulted in poor fitting. Another problem is that since the yaw of repose is $\vec{\alpha}_e$ is a cross product between the velocity vector $\vec{v}$ and the acceleration $\ddot{\vec{x}}$, adjusting the aerodynamic lift force by varying C_L causes variations

of the trajectory in all directions through $\vec{\alpha}_e$. This is a quite severe model bias, so there is a possibility to end up in a situation where adjusting the the lift force to minimize the error in the 2-direction (SE) causes a larger error in the 3-direction (SA).

Unfortunately a choice had to be made to simply omit the yaw of repose as well, because of this a simpler form of model compensation had to be made to adjust for the loss of Magnus force and Lift force which were entirely dependent on the yaw. The choice fell on a simple cubic fitting of the modeling error against tabular data for the ballistic deviation angles as such:

$$SE_{corr} = SE + \frac{a_s t_{fl} + b_s t_{fl}^2 + c_s t_{fl}^3}{\cos(EL)} \tag{5.2}$$

$$SA_{corr} = SA + a_a SE_{corr} + b_a SE_{corr}^2 + c_a SE_{corr}^3 \tag{5.3}$$

MATLAB® curve fitting toolbox can be used to find the parameters a_s, b_s, c_s, a_a, b_a and c_a that minimizes the remaining modeling error of the ballistic deviation angles SE and SA against all the trajectories in the tabular firing-tables. Note that these corrections do not occur in the actual equation of motion but rather after the trajectory have been solved integrative.

Chapter 6

PARAMETERIZATION EVALUATION AND MIL RESULTS

6.1 Parameterization evaluation

The method for finding the aerodynamic drag coefficient C_D was evaluated using the firing-tables for the spin-stabilized tracer projectile with known aerodynamic coefficients supplied by the munitions manufacturer in terms of how well the method is able to reproduce the aerodynamic drag coefficient. Figure 6.1 shows the results of the proposed method, the curve looks very similar to that of the general shape of a drag coefficient however a large amount of oscillations has occurred.

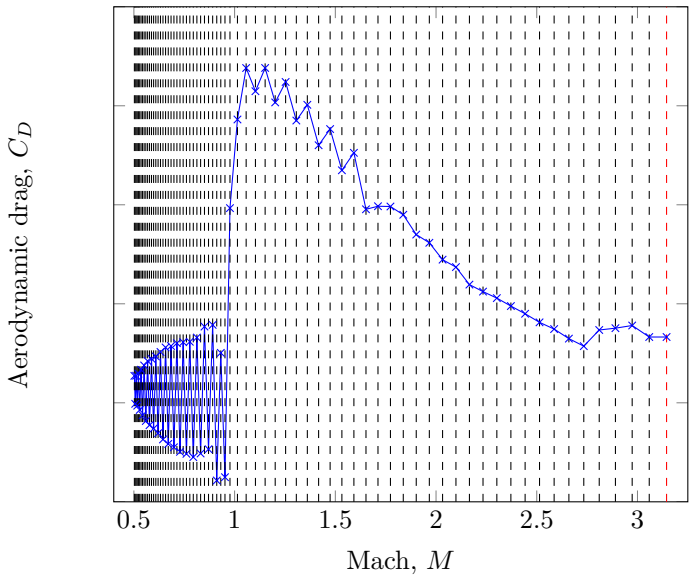


Figure 6.1: The C_D curve as a linear interpolation of M, the dashed lines represents the boundaries for the different Mach regions. While the C_D curve looks to have the right shape it oscillates largely.

The oscillations can be the cause of many things, a first observation is that the number of integration steps affects the extrapolation of the drag coefficient. Increasing the number of integration steps is essential to assure that the integrator actually takes sufficient steps within each Mach region. Secondly, the way the proposed method works introduces problems; if the initial C_{D_1} guess is wrong it will affect the next guess of C_{D_2}. The method tries to minimize the terminal velocity error ϵ_{vt} so the effect is that it compensates for the initial error by introducing an error in the other direction, a way to reduce this effect is by allowing some slack in ϵ_{vt}. Figure 6.2 shows the results of the same method with a larger slack allowed, increased number of integration steps and a smaller step size in the steepest descent method.

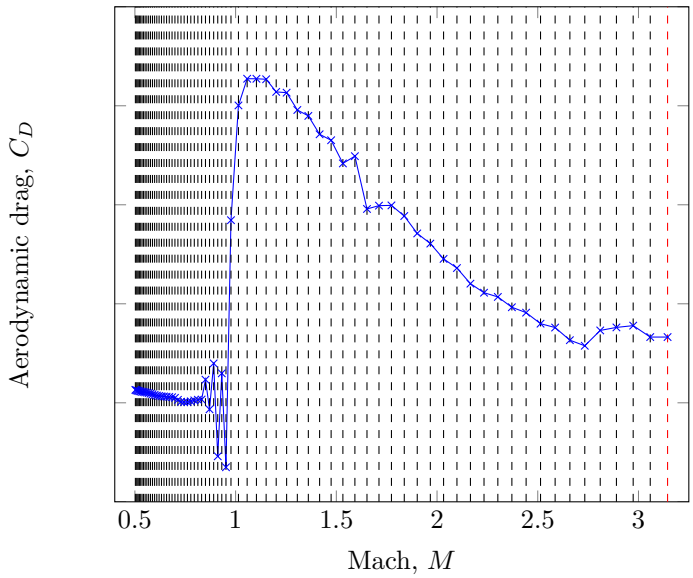


Figure 6.2: With increased number of integration steps, smaller steepest descent steps and a larger allowed slack the oscillations are reduced significantly.

Although the oscillations are reduced significantly it is still not good enough considering the actual FCS have to take a significantly smaller amount of integration steps then what was used to produce the curve in figure 6.2. To reduce the noise further a averaging filter is used:

$$[M_{n+1}, C_{D_n+1}] = \left[\frac{M_n + M_{n+1}}{2}, \frac{C_{D_n} + C_{D_n+1}}{2}\right], \qquad n = 1, 2...N-1 \tag{6.1}$$

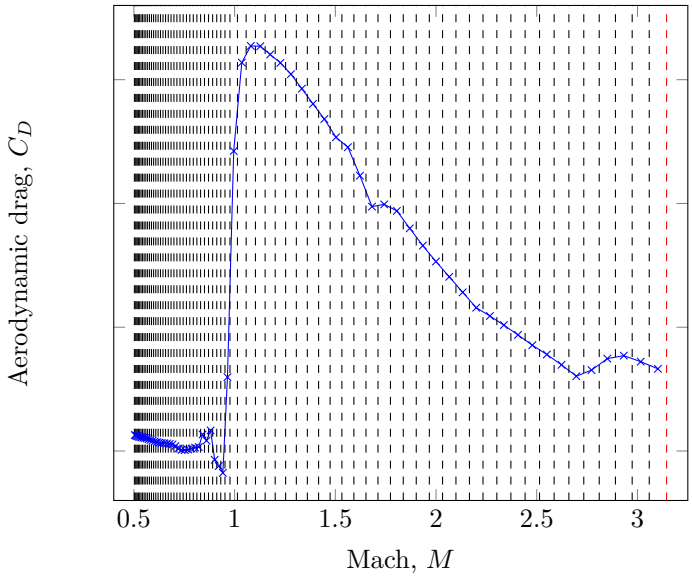


Figure 6.3: The filtered C_D curve looks to be usable, the oscillations are reduced but the error ϵ_{vp} has to be reevaluated.

The filtered result of the extrapolated aerodynamic drag coefficient looks to be usable, however the actual fitting has to be reevaluated, the goal is to minimize ϵ_{vt}. The points are also now in the middle of each Mach region, using linear interpolation points are added at each boundary line between the Mach region. The external ballistics model is then evaluated again for each trajectory in terms of ϵ_{vt}. Where the error in ϵ_{vt} is big enough, the added points are adjusted slightly to reduce error ϵ_{vp}, this may reintroduce some of the oscillative behavior, but is beneficial to reduce the error in terminal velocity ϵ_{vt}. Figure 6.4 shows the final aerodynamic drag coefficient C_D as a linearly interpolated list of Mach M. The actual aerodynamic drag coefficient provided for this example projectile is also shown.

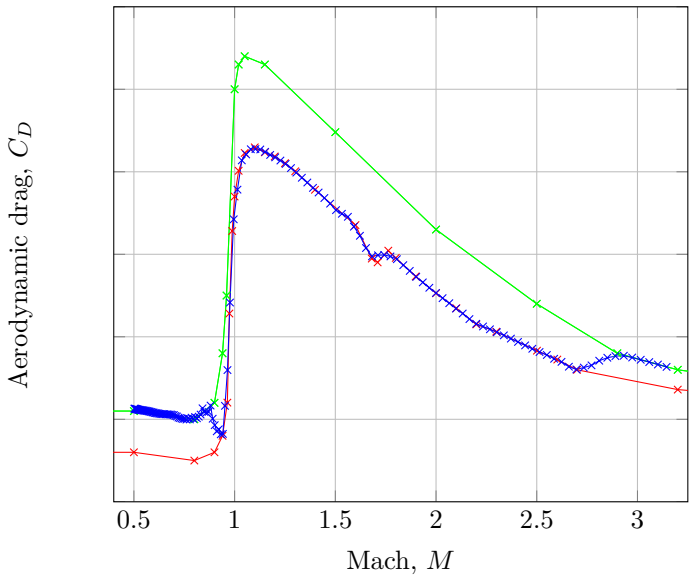


Figure 6.4: The final parameterization results of the aerodynamic drag coefficient C_D (blue). Actual data provided by the manufacturer show two drag states, inert state drag coefficient (green) and burning state drag coefficient (red). The resulting drag coefficient curve transitions between the two states.

The example projectile is a tracer projectile that turns out to have two drag states corresponding to if the projectile is lit or inert. The extrapolated drag coefficient can be seen to transition in between these two states and clearly follows the underlying drag coefficient used to produce the firing tables.

6.2 MIL results

The system level requirements are a set of maximum allowed model error against the firing-tables, these are split into three range intervals; short (100 m to 2000 m), medium (2100 m to 3000 m) and long (3100 m to 4000 m) ranges with different levels of tolerated error relative firing tables.

A numerical solution for exterior ballistics has been implemented in MATLAB®, with behavioral simulation the model can be verified against the system level requirements and model fitness against firing tables can be compared against the linearized external ballistics model presented in section 2.1. Conditions are set to nominal according to the firing tables as shown in table 6.1.

Table 6.1: Nominal conditions for the spin-stabilized projectile according to its firing tables.

Variable	Value	Description
lat	$45°$	Geographical latitude
w_1	$0\,\mathrm{m\,s^{-1}}$	Wind speed along projectile (1-direction)
w_3	$0\,\mathrm{m\,s^{-1}}$	Wind speed across projectile (3-direction)
$\dot{x}_0$	$1070\,\mathrm{m\,s^{-1}}$	Muzzle velocity
p	$101\,325\,\mathrm{Pa}$	Air pressure
T	$288.15\,\mathrm{K}$	Air temperature
EL	$0°$	Elevation angle to target

The tabular data does not consider vertical wind speeds or the Coriolis effect, so w_2 is set to zero and F_c is simply omitted in the trajectory model.

In figure 6.5 the model error in time of flight t_{fl} relative firing tables is presented. The system level requirements as maximum allowed error are visualized as red dashed lines. Both the error relative firing tables for the simplified model (shown in blue) and the error relative firing tables (shown in black) are visualized. Smaller error represents a better fitted model against the tabular data.

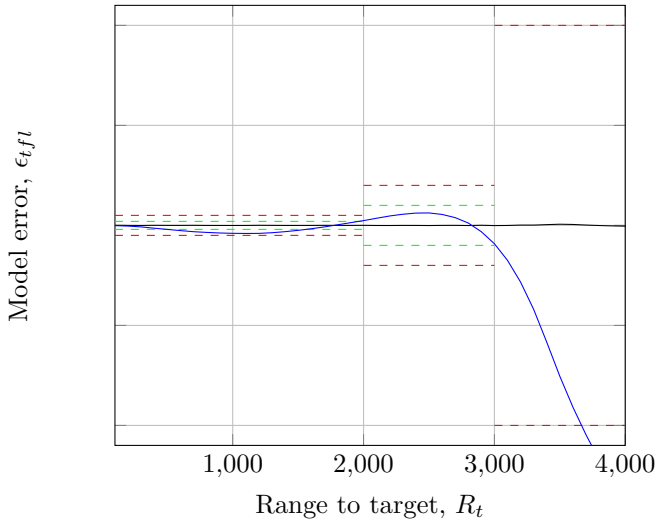


Figure 6.5: The model error in time of flight relative firing tables for nominal conditions. The red dashed lines are system level requirements, the green are desired levels. Requirements are only present for short, medium and long ranges, ranges outside of that are not considered. The numerical external ballistics is represented as a black line, and the blue is the linearized ballistics.

Table 8.4 shows the largest error for each range interval relative the requirements. The differences in error is calculated for short, medium and long ranges as such:

$$\Delta\epsilon_{tfl} = \frac{\epsilon_{tflLINEAR}}{\epsilon_{tflNUMERICAL}} \tag{6.2}$$

where $\epsilon_{tflLINEAR}$ is the error against the firing tables for the linearized model and $\epsilon_{tflNUMERICAL}$ for the integrative model respectively. $\Delta\epsilon_{tfl}$ is a relative term that shows the relative error reduction achieved by using the nonlinear external ballistics model and numerical solution as opposed to the current linearized model.

Table 6.2: Relative error size for time of flight.

	Short range	Medium range	long range
$\Delta\epsilon_{tfl}$	0.012	0.012	0.004

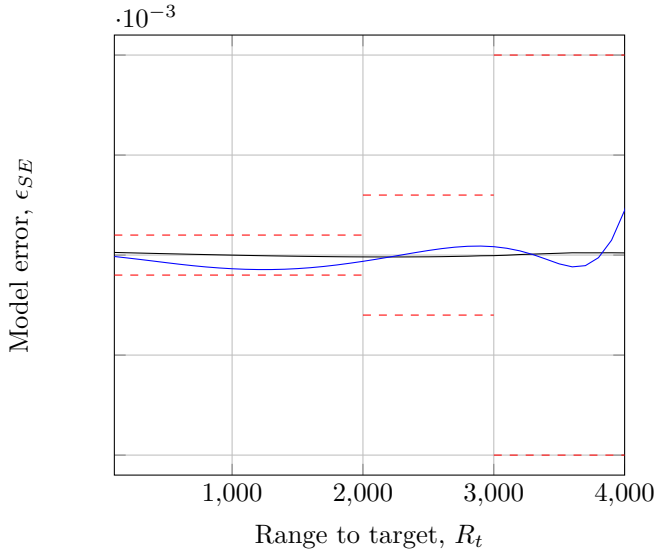


Figure 6.6: Short to long ranges

Table 6.3: Relative error size for the ballistic deviation angle SE

	Short range	Medium range	long range
$\Delta\epsilon_{SE}$	0.1809	0.1809	0.0585

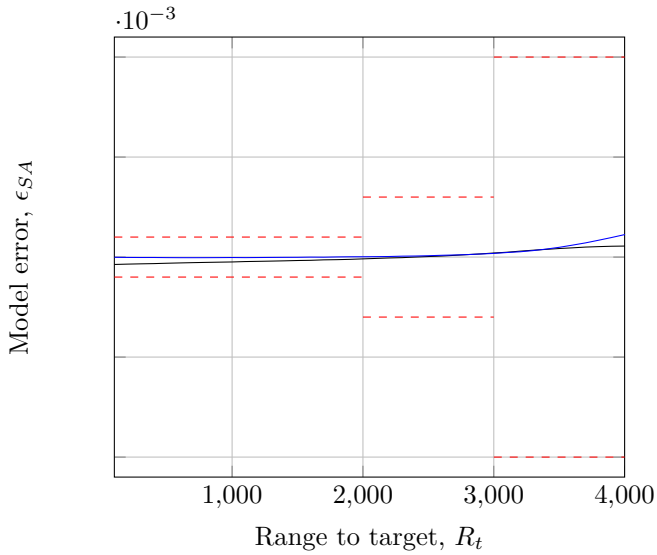


Figure 6.7: Short to long ranges

Table 6.4: Relative error size for the ballistic deviation angle SA

	Short range	Medium range	long range
$\Delta\epsilon_{SA}$	11.8211	2.0305	0.4897

Chapter 7

SOFTWARE AND HARDWARE IMPLEMENTATION

The next step in the development process involved implementing everything on the actual real-time embedded system. The external ballistics calculation is carried out on the hardware platform and evaluated through the implementation of certain test cases, then the behavior is compared to that of firing tables for verification purposes.

In the target platform software runs in modules written in C++, each preemptively scheduled by a RTOS with a set cycle frequency and priority. The targeted system is operating under hard real-time conditions so there are time constraints, namely for the system to function properly all of the cyclically executed module tasks must be completed before their next cycle deadline. Each of these modules is implemented as a C++ class and a singleton design pattern is used to limit the instantiation of the class to one object, each of these classes consists of a method which is the cyclically scheduled task.

The target platform software modules are split up into three cycle frequencies groups, a module running a higher cycle frequency always has higher priority then a module running in a lower frequency. Software modules running within the same cycle frequency group are manually prioritized. The three groups are high, medium and low frequency, where 64 high frequency or 4 medium frequency calculation cycles corresponds to one low frequency calculation cycles.

To implement the external ballistics mainly two new modules were written; one ballistics module running in high frequency and one prediction module running the 64 times slower low frequency. The ballistics module contains the actual external ballistics trajectory model and for each cycle one integration step is taken. The prediction module is what contains the algorithms for finding the ballistic deviation angles and the predicted impact location for a moving target.

7.1 Ballistics module

The ballistics module is what contains the actual external ballistics, it runs at a high cycle frequency and for each calculation cycle solves one integration step. A simple functional description of the method that is called each calculation cycle in the ballistics module is shown in figure 7.1

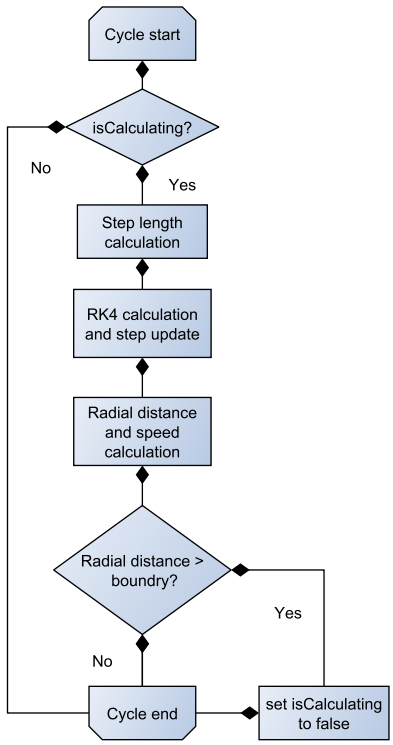


Figure 7.1: Functional description of the cyclically executed method in the ballistics module.

7.2 Prediction module

The prediction module is what constantly tries to find the initial condition of SE and SA as well as predict a impact location. It controls the ballistics module by getting the end states, setting the initial states and starting the integration appropriately.

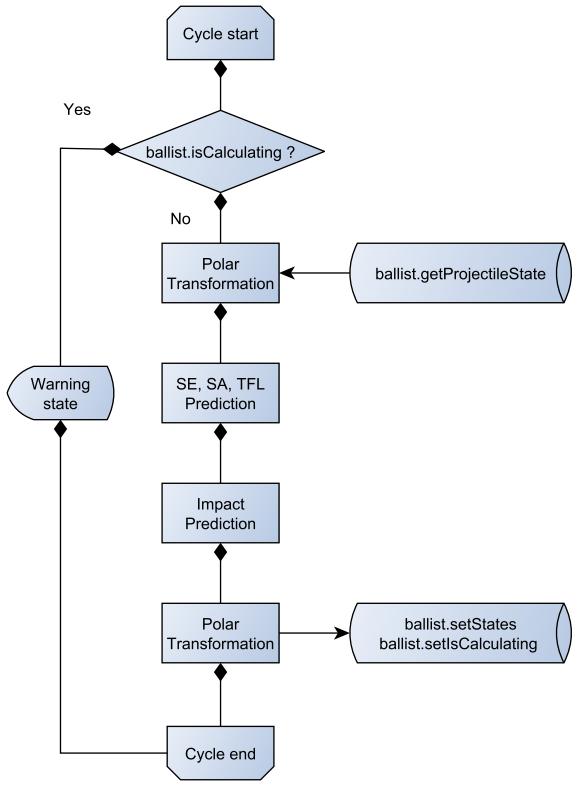


Figure 7.2: Functional description of the cyclically executed method in the prediction module.

7.3 Implementation of test case

A test case is implemented in the software to help with the verification and validation process. System level requirements are as stated in section 6.2 specifically a set of maximum allowed model differences relative the firing tables. To verify that the system complies with these requirements the test case is specifically set up to test these. It is implemented as a state within the estimator, ballistics and predictor modules that when entered, overrides any sensor with specific user input conditions. It then loops through a user input set of target locations, giving the system time to converge for each range and for the user to log relevant states. This test case is implemented to work with both the numerically solved nonlinear ballistics and prediction module as well as the linearized ballistics and prediction module and allows the user to run the test for both the linearized and numerical solution in parallel logging the results from both for each case.

Chapter 8

FINAL RESULTS

8.1 Calculation intensity comparison

A simpler method for evaluating the increase in calculation intensity on the FCS by using the numerical non-linear ballistics consists of evaluating the calculation time spent in each of the three task frequency groups. The FCS is put in a calculation intensive state, then the maximum cycle time spent in each frequency group is logged over a time, this is then used to calculate how much of the total resources is allocated to each of the three task frequency groups. Figure 8.1 shows the maximum percentage of available calculation time spent in each of these three frequency groups.

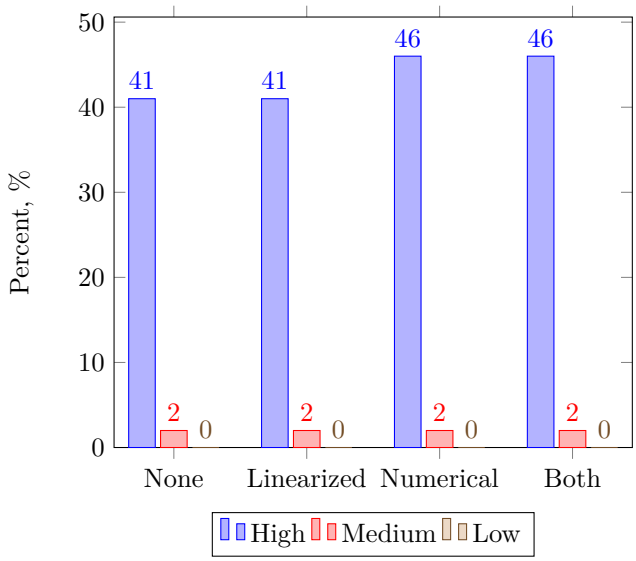


Figure 8.1: The maximum amount of total available calculation time spent in each cycle frequency group when executing the linearized ballistics, numerical ballistics, both or none of these tasks.

8.2 Model error comparison

The results are taken from the actual FCS using the test case described in section 7.3. With this method the system was set up according to atmospheric and ballistic conditions stated in the firing tables and then looped through all the target locations, logging the resulting trajectory solutions for both the linearized and the integrative method. The resulting trajectory solutions was then read into MATLAB® and compared to the firing tables giving the differences ϵ_{tfl}, ϵ_{SE} and ϵ_{SA}. System level requirements give three levels that these errors needs to be kept within for short, medium and long ranges respectively. There are no system level requirements on model fitness for trajectories exceeding these set of ranges, even if the firing tables contain such. The errors are shown in graphs together with system level requirement levels to get a overview of how well the model is fitted, the results are given in tables as the reduction factor of the modeling errors relative firing table achieved by using the integrative ballistics compared to the linearized ballistics.

The same previously mentioned spin-stabilized tracer projectile is evaluated, the aerodynamic coefficients have been parameterized using the method described in chapter 5 and implemented into software as described in chapter 7. Conditions are set according to match firing table; the Coriolis force is omitted and *lat* is set to 45° as stated by the firing tables. The atmospheric conditions are set to nominal as shown previously in table 6.1, this is the nominal conditions.

Nominal conditions

The Coriolis effect is omitted and conditions are set according to nominal as shown in table 6.1.

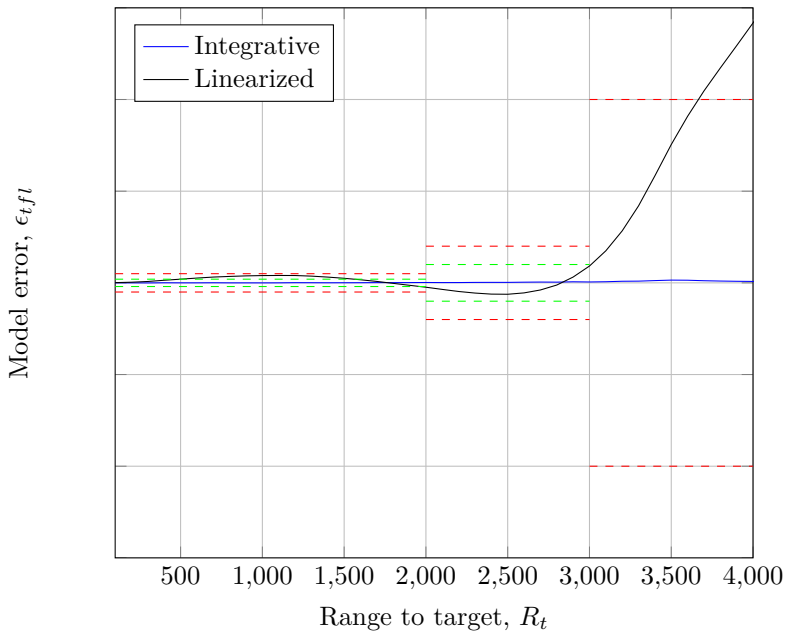


Figure 8.2: The error ϵ_{tfl} in time of flight t_{fl} relative firing tables for the fitted integrative ballistics and linearized ballistics respectively during nominal conditions. The red dashed lines are set system level requirements for the short, medium and long ranges respectively

Table 8.1: Relative error size for time of flight t_{fl}. A measure on the relative reduction in error size achieved by using the integrative method as opposed to the linearized approach.

	Short range	Medium range	long range
$\Delta\epsilon_{tfl}$	0.049	0.064	0.011

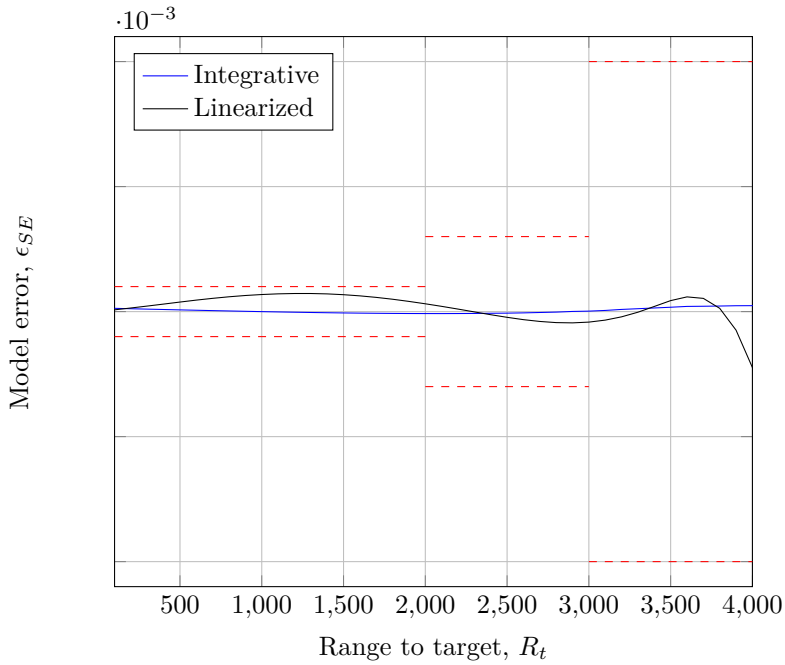


Figure 8.3: The error ϵ_{SE} in Superimposed-Elevation SE relative firing tables for the fitted integrative ballistics and linearized ballistics respectively during nominal conditions. The red dashed lines are set system level requirements for specific range intervals.

Table 8.2: Relative error size for time of flight. A measure on the relative reduction in error size achieved by using the integrative method as opposed to the linearized approach.

	Short range	Medium range	long range
$\Delta\epsilon_{SE}$	0.181	0.175	0.106

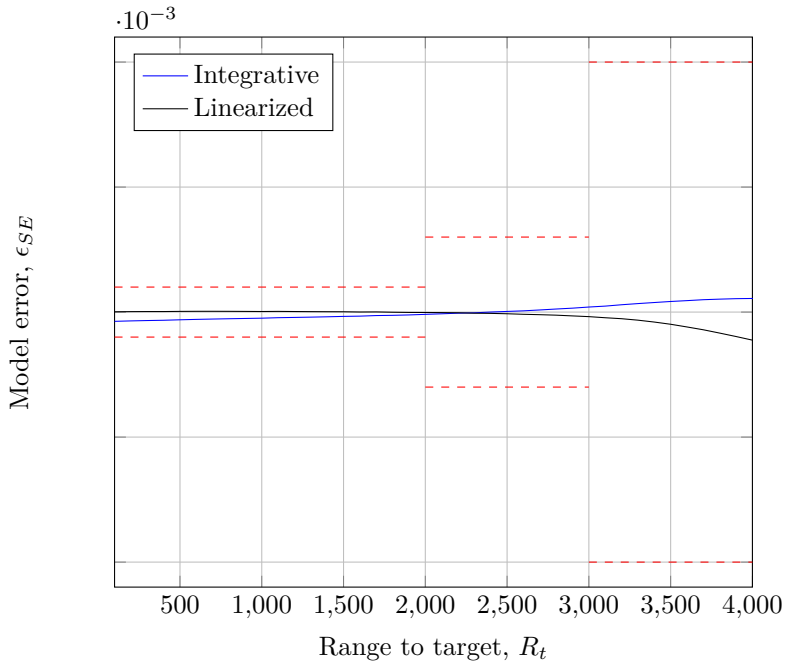


Figure 8.4: The error ϵ_{SA} in Superimposed-Azimuth SA relative firing tables for the fitted integrative ballistics and linearized ballistics respectively during nominal conditions. The red dashed lines are set system level requirements for specific range intervals.

Table 8.3: Relative error size for time of flight. A measure on the relative reduction in error size achieved by using the integrative method as opposed to the linearized approach.

	Short range	Medium range	long range
$\Delta\epsilon_{tfl}$	11.883	1.106	0.485

Muzzle velocity influences

This section shows the results for a deviation in muzzle velocity $\Delta\dot{x}_0$ of $10\,\mathrm{m\,s^{-1}}$ relative nominal conditions.

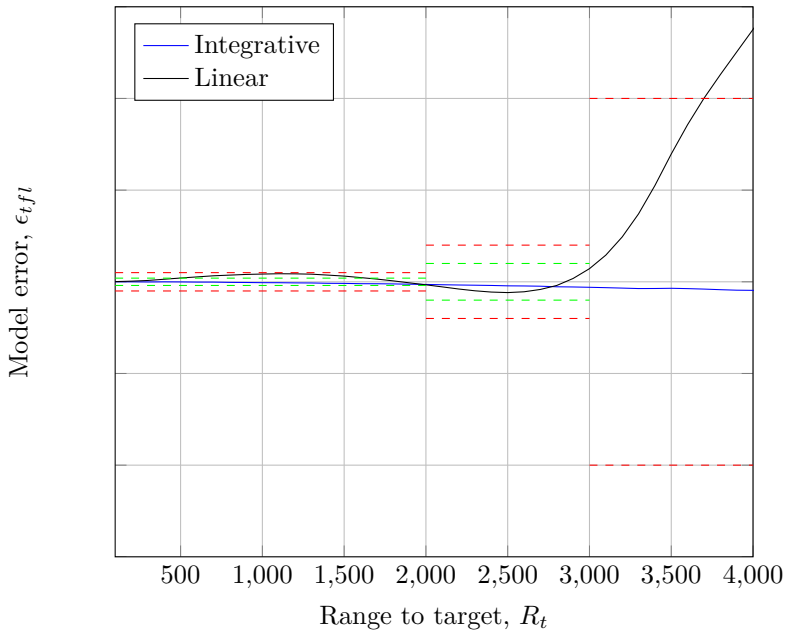


Figure 8.5: The error ϵ_{tfl} in time of flight t_{fl} relative firing tables for the fitted integrative ballistics and linearized ballistics respectively during conditions $\Delta\dot{x}_0 = 10\,\mathrm{m\,s^{-1}}$. The red dashed lines are set system level requirements for the short, medium and long ranges respectively

Table 8.4: Relative error size for time of flight t_{fl}. A measure on the relative reduction in error size achieved by using the integrative method as opposed to the linearized approach.

	Short range	Medium range	long range
$\Delta\epsilon_{tfl}$	0.327	0.418	0.034

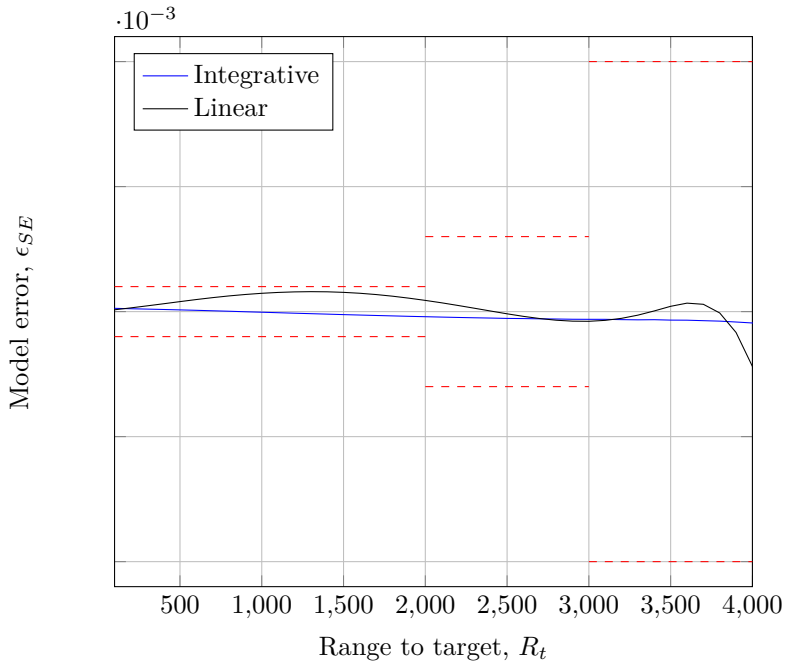


Figure 8.6: The error ϵ_{SE} in Superimposed-Elevation SE relative firing tables for the fitted integrative ballistics and linearized ballistics respectively during nominal conditions. The red dashed lines are set system level requirements for specific range intervals.

Table 8.5: Relative error size for Superimposed-Elevation SE. A measure on the relative reduction in error size achieved by using the integrative method as opposed to the linearized approach.

	Short range	Medium range	long range
$\Delta\epsilon_{SE}$	0.255	0.796	0.207

Tailwind influences

This section shows the results for a positive change in the wind component $w_1 = 10\,\mathrm{m\,s^{-1}}$ relative nominal conditions.

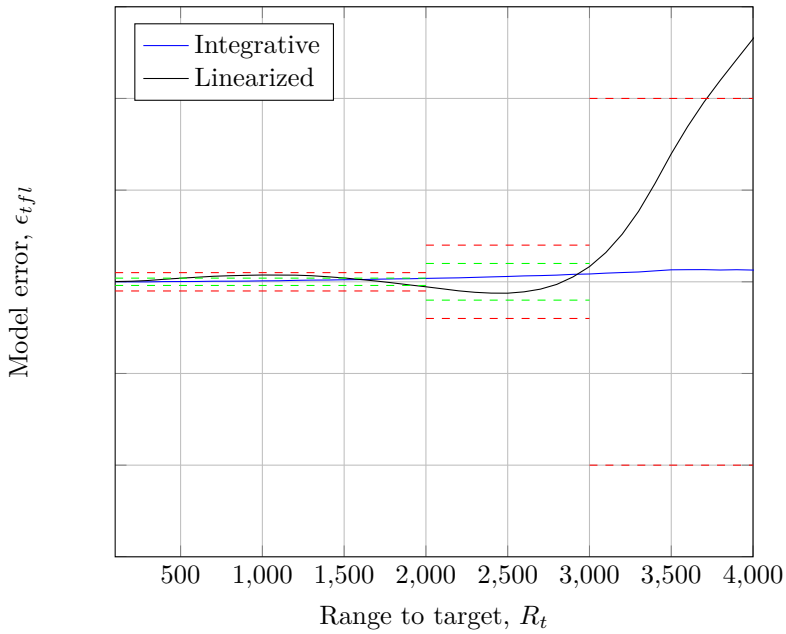


Figure 8.7: The error ϵ_{tfl} in time of flight t_{fl} relative firing tables for the fitted integrative ballistics and linearized ballistics respectively during conditions $w_1 = 10\,\mathrm{m\,s^{-1}}$. The red dashed lines are set system level requirements for the short, medium and long ranges respectively

Table 8.6: Relative error size for time of flight t_{fl}. A measure on the relative reduction in error size achieved by using the integrative method as opposed to the linearized approach.

	Short range	Medium range	long range
$\Delta\epsilon_{tfl}$	0.503	0.516	0.050

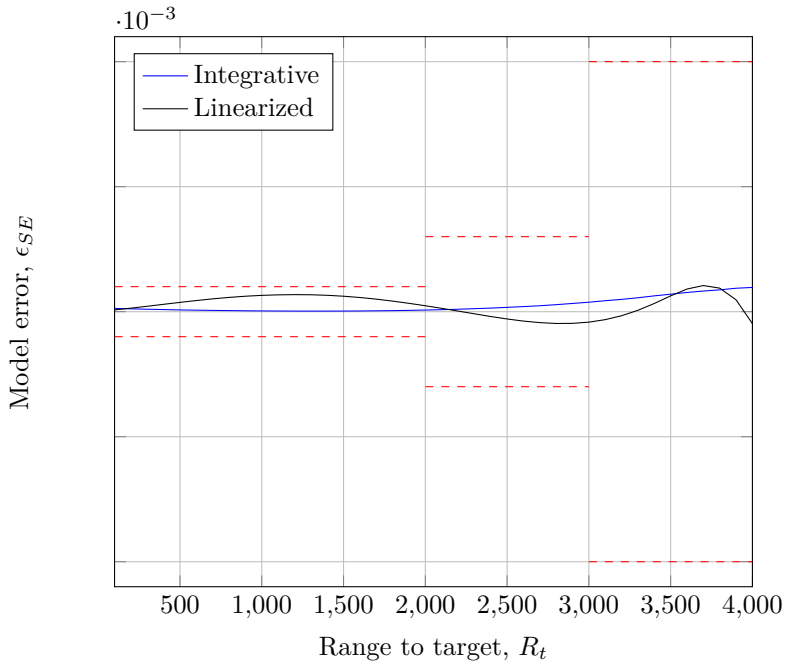


Figure 8.8: The error ϵ_{SE} in Superimposed-Elevation SE relative firing tables for the fitted integrative ballistics and linearized ballistics respectively during conditions $w_1 = 10\,\mathrm{m\,s^{-1}}$. The red dashed lines are set system level requirements for specific range intervals.

Table 8.7: Relative error size for Superimposed-Elevation SE. A measure on the relative reduction in error size achieved by using the integrative method as opposed to the linearized approach.

	Short range	Medium range	long range
$\Delta\epsilon_{SE}$	0.192	0.804	0.928

Headwind influences

This section shows the results for a negative change in the wind component $w_1 = -10\,\mathrm{m\,s^{-1}}$ relative nominal conditions.

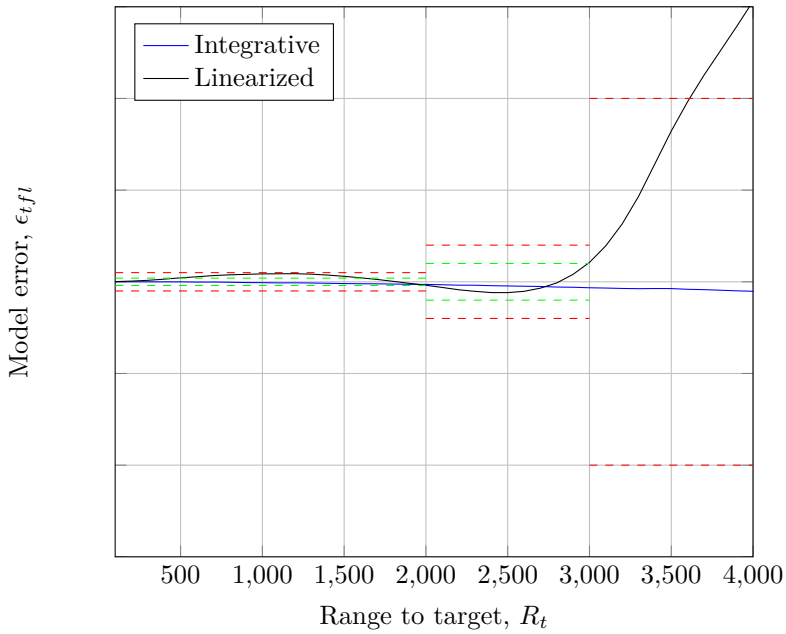


Figure 8.9: The error ϵ_{tfl} in time of flight t_{fl} relative firing tables for the fitted integrative ballistics and linearized ballistics respectively during conditions $w_1 = -10\,\mathrm{m\,s^{-1}}$. The red dashed lines are set system level requirements for the short, medium and long ranges respectively

Table 8.8: Relative error size for time of flight t_{fl}. A measure on the relative reduction in error size achieved by using the integrative method as opposed to the linearized approach.

	Short range	Medium range	long range
$\Delta\epsilon_{tfl}$	0.324	0.308	0.034

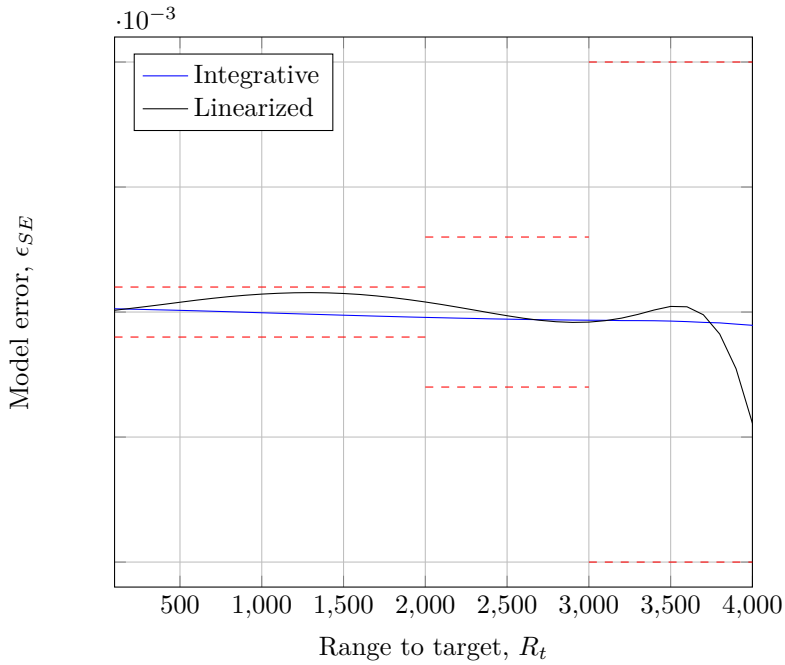


Figure 8.10: The error ϵ_{SE} in Superimposed-Elevation SE relative firing tables for the fitted integrative ballistics and linearized ballistics respectively during conditions $w_1 = -10\,\mathrm{m\,s^{-1}}$. The red dashed lines are set system level requirements for specific range intervals.

Table 8.9: Relative error size for Superimposed-Elevation SE. A measure on the relative reduction in error size achieved by using the integrative method as opposed to the linearized approach.

	Short range	Medium range	long range
$\Delta\epsilon_{SE}$	0.275	0.788	0.119

Crosswind influences

This section shows the results for a change in the wind component $w_3 = 10\,\mathrm{m\,s^{-1}}$ relative nominal conditions.

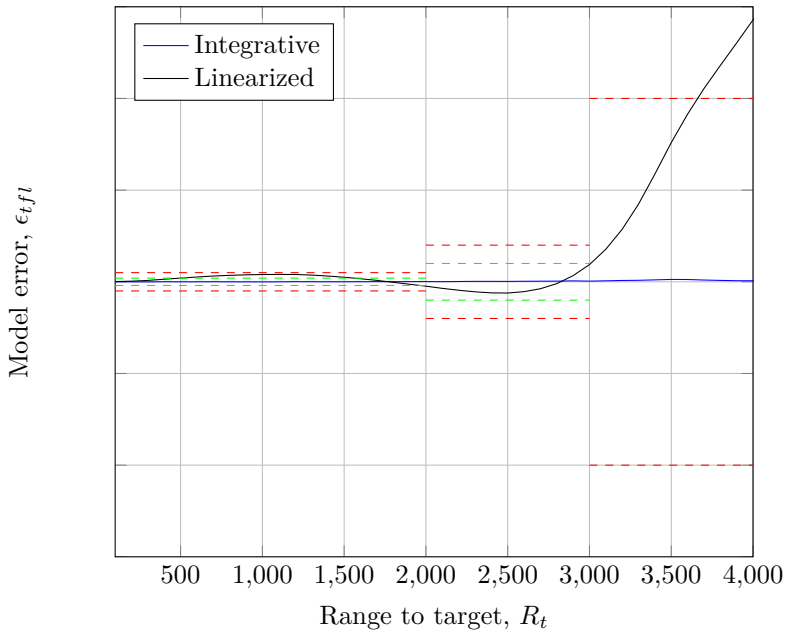


Figure 8.11: The error ϵ_{tfl} in time of flight t_{fl} relative firing tables for the fitted integrative ballistics and linearized ballistics respectively during conditions $w_3 = 10\,\mathrm{m\,s^{-1}}$. The red dashed lines are set system level requirements for the short, medium and long ranges respectively

Table 8.10: Relative error size for time of flight t_{fl}. A measure on the relative reduction in error size achieved by using the integrative method as opposed to the linearized approach.

	Short range	Medium range	long range
$\Delta\epsilon_{tfl}$	0.033	0.054	0.009

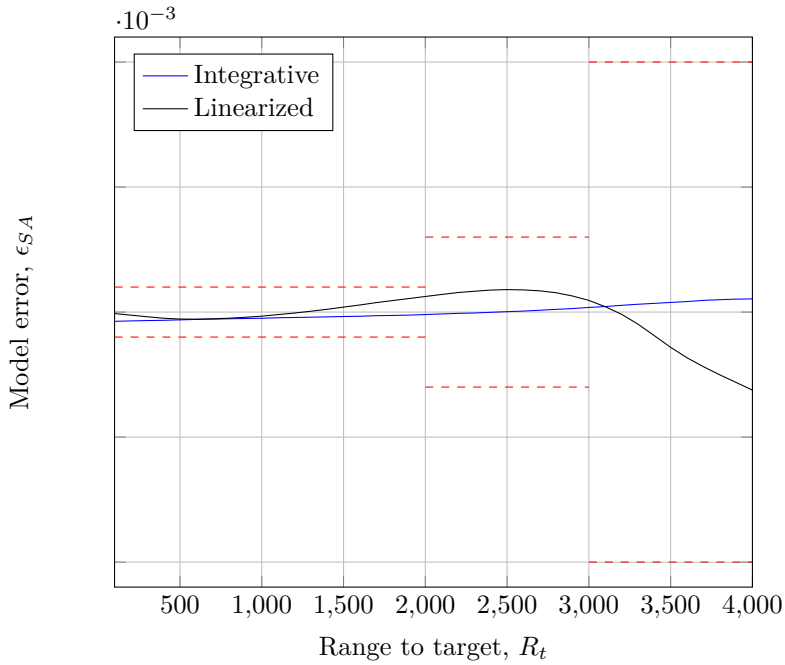


Figure 8.12: The error ϵ_{SA} in Superimposed-Azimuth SA relative firing tables for the fitted integrative ballistics and linearized ballistics respectively during conditions $w_3 = 10\,\mathrm{m\,s^{-1}}$. The red dashed lines are set system level requirements for specific range intervals.

Table 8.11: Relative error size for Superimposed-Azimuth SA. A measure on the relative reduction in error size achieved by using the integrative method as opposed to the linearized approach.

	Short range	Medium range	long range
$\Delta\epsilon_{SE}$	0.594	0.207	0.169

Chapter 9

DISCUSSION AND CONCLUSIONS

9.1 Discussion and suggestions for future work

A numerically solved external ballistics model is perhaps most benefited from in the case of indirect long range fire, where trajectory arcing, transonic speed and projectile yaw play a much bigger role in determining the impact location. In this case of direct fire the external ballistics only start to play a bigger role when firing at longer ranges. However for even for the relative short ranges involved in the direct FCS, improvements could clearly be seen. As described in section 2.1, the linearized model assumes the traveled distance can be approximated in one direction only. I would claim that this approximation coupled with a linearized aerodynamic drag coefficient model that only works well for supersonic velocities is the major cause of the somewhat poor time of flight accuracy in the linearized model, where the biggest relative improvements could be seen.

A big limitation specific to this thesis is the requirement of aerodynamic coefficient data. Preferably this should be provided from the projectile manufacturer, however as this is not always possible, parameterization is necessary. Parameterization of aerodynamic characteristics from firing tables proved rather cumbersome and some assumptions were made to depict a method to solve the aerodynamic coefficients. The major determining part of how well the non-linear model performs was how well the aerodynamic coefficients could be fitted. Note that the parameterization process is hard to automate because theres simply a lot that can go wrong in the process, also it is very time-consuming which makes it non-practical to use methods that already requires large amounts of computation. Even with the rather simple parameterization process used in this thesis the process could take up to half an hour depending on the accuracy used in the numerical method and trajectory solver.

The aerodynamic drag play a major role of determining the projectile end-point,

the drag force exert the vast majority of work on the projectile over the trajectory especially in the beginning of the trajectory. As such, it was found out that the numerical model was very sensitive to model bias in the aerodynamic drag coefficient, especially at higher Mach. Attempts were made to find a continuous model for the aerodynamic drag coefficient, however this did not produce satisfactory results for this reason, a slight bias in the higher Mach regions of the aerodynamic drag would cause growing errors at longer ranges. This means that although perhaps preferable, it is hard to fit a continuous model as each projectile can have different characteristics in the shape of the drag coefficient.

Considering the external ballistics, the developed non-linear model has quite some bias, it more or less assumes that the projectile data supplied in the form of firing tables is derived from a 6DOF ballistics model. While this is the case most of the time there might exist firing tables where the underlying model is of simpler sort, this could perhaps cause the numerical ballistics model to be very hard to fit. I would recommend to test this model against firing tables from several munition manufacturers.

Floating point precision

Although verification steps not covered in this report was carried out throughout the development of the software when moving to the hardware platform, there are slight differences in how well the actual numerical solution performed on the FCS compared to that of the MATLAB® results. The results during the MIL testing were better, especially for shorter ranges, compared to the actual HIL results. In reality however this is not too surprising, MATLAB® used double floating point precision whereas on the target platform single floating point precision was used. When the resulting error is very small such as for the numerical solution, it is quite understandable that such an reduction in calculation precision can explain the smaller differences.

Atmospheric influences

The standardized atmospheric model used in the non-linear model is incapable of considering for variation in atmospheric temperature and pressure at a static altitude, it is only a consideration for the atmospheric variations due to altitude changes. It essentially assumes that at mean sea level the air pressure is 101 325 Pa and the air temperature 15 °C. For some projectiles this is not enough and the FCS needs to consider atmospheric variations at static altitude as well. A quick and easy solution was attempted however this proved to be more of a hassle than depicted. This is something that should be investigated further.

Calculation intensity

For such a safety critical system as a FCS is, proper evaluation must be made that there is no case that can cause the system to fail or cause undefined behavior. Although it is clear from the calculation intensity evaluation that there is likely no problem with the system load, I would not consider such an evaluation enough for anything else other then evaluation and testing purposes. For a full evaluation one would have to look the time spent in each and every timing critical task by for example output pin toggling and oscilloscope measurements and use this together with a worst case scenario prediction to calculate if there is any state that could cause the system to miss a timing constraint. This is however quite a massive job considering the amount of states for each software module can have and the amount of modules. Exhaustive testing might not be possible, but for a safety critical system such as this I believe the test cases have to cover a vast majority of possible states and outcomes.

This said the actual intensity evaluation proved that there might exist room to for example double the amount of integration steps in the ballistics calculation cycle, which would improve the accuracy.

Moving targets

A full evaluation of the moving target impact point prediction is not something easily achieved without a full system hardware plant because in reality the results depends on several delays such as actuation, calculation, signal propagation, discrete and analog filtering and electrical etc. The moving target tests should perhaps therefore be carried out with the full system in place with some empirical testing of the delay compensation term t_c.

9.2 Conclusions summarized

While a numerical solution for a non-linear model has benefits it also comes with some drawbacks, it is important to point these out. The numerical solution is inherently much more complex and with greater complexity there is a lot more that can go wrong or malfunction. If the performance of a linearized external ballistics model is generally not considered a problem then it should perhaps therefore be preferred over numerical solutions.

The same applies to the method used to solve the parameterization problem, a linearized model allows for easier methods to be used while a non-linear model solved numerically requires more extensive methods that might not be as easy to automate. Ideally when the FCS needs to adopt a new projectile type the parameterization process and implementation should be rather simple and fast to carry out and not require extensive knowledge of the inner workings of the parameterization in case something unexpected happens.

On the positive side the numerical solution have the ability to better predict projectile trajectories given the right parameters the reason being a non-linear external ballistics model closer to that of the real physical process can be solved. Especially beneficial is the ballistics accuracy for long range trajectories where the transition from supersonic to subsonic speed is important and the trajectory arcs significantly.

Perhaps the most important conclusion is that a numerically solved non-linear external ballistics model is not only feasible but implementable and can improve results, even considering the hard real-time constraints of the specific direct FCS.

Bibliography

Leszek Baranowski. 2013a. equations of motion of a spin-stabilized projectile for flight stability testing. *Theoretical and applied mechanics*, 51(1):235–246. Warsaw.

Leszek Baranowski. 2013b. Feasibility analysis of the modified point mass trajectory model for the need of ground artillery fire control systems. *Theoretical and applied mechanics*, 51(3):511–522. Warsaw.

Leszek Baranowski. 2013c. Numerical testing of flight stability of spin-stabilized artillery projectiles. *Theoretical and applied mechanics*, 51(2):375–385. Warsaw.

Ilmars Celmins. 1990. *Projectile Supersonic Drag Characteristics.* U. S. Army Ballistic Research Laboratories. UNCLASSIFIED.

Elizabeth R. Dickinson. 1965. *The Zero-Yaw Drag Coefficient for Projectile, 8-Inch: HE, M106.* Ballistic Research Laboratories. UNCLASSIFIED U. S. Army Materiel Command.

Gerald M. Gregorek. 1970. Aerodynamic drag of model rockets. ESTES INDUSTRIES, INC. Penrose Colorado 81240.

1985. *Flight dynamics - Concepts and symbols - Part 2: Motions of the aircraft and the atmosphere relative to the Earth.* International Organization for Standardization.

1997. *International Standard Atmosphere ISO 2533.* International Organization for Standardization.

Craig H. Smith Jizhang Sang, James C. Bennett. 2013. Estimation of ballistic coefficients of low altitude debris objects from historical two line elements. *Advances in Space Research*, 52:117–124.

Ruprecht Nennstiel. 2013. How do bullets fly? `http://www.nennstiel-ruprecht.de/bullfly/`. Accessed: 2014-04-8.

Mary L. Reiter Robert F. Lieske. 1966. *Equations of Motion for a Modified Point Mass Trajectory.* Ballistic Research Laboratories. UNCLASSIFIED U. S. Army Materiel Command.