NABK BASED NEXT GENERATION BALLISTIC TABLE TOOLKIT

Sevsay Aytar Ortac[1], Umut Durak[2], Umit Kutluay[3], Koray Kucuk[4], Maj. Can Candan[5]

TUBITAK-SAGE, Flight Mechanics Division, 06261, Ankara, TURKEY, saytar@sage.tubitak.gov.tr
TUBITAK-SAGE, Modeling & Simulation Division, 06261, Ankara, TURKEY, udurak@sage.tubitak.gov.tr
TUBITAK-SAGE, Flight Mechanics Division, 06261, Ankara, TURKEY, ukutluay@sage.tubitak.gov.tr
TUBITAK-SAGE, Modeling & Simulation Division, 06261, Ankara, TURKEY, kkucuk@sage.tubitak.gov.tr
TLFC, Technical & Project Management Division, 06100, Ankara, TURKEY, ccandan@kkk.tsk.mil.tr

Abstract

Traditionally tabulated and graphical firing data were widely used for the solution of the fire control problems. While contemporarily computer based fire control systems enable quick and accurate results, traditional tables are still in charge as backup. It is of vital importance that solutions obtained by both methods should match within an acceptable tolerance, which can be accomplished by using same ballistic kernel.

Currently NATO Armament Ballistic Kernel (NABK-previously NATO Artillery Ballistic Kernel), which is developed by an international team of different countries including Turkey, with the scope of being joint fire control solution for different armaments including light arms, mortars, artillery and rockets, stands as this ballistic kernel.

This paper provides a brief overview of NABK and its layers which are used as the bases of the toolkit developed by TUBITAK-SAGE for the generation of ballistic tables. Until today three different products developed as a part of the toolkit to be used for three different applications, namely tabular firing tables and graphical firing tables for artillery, bombing tables for unguided aircraft bombs.

INTRODUCTION

Ballistic tables have been used in the battlefield for along time to find firing solutions. With the introduction of technical fire control software, ballistic tables stand as the back up systems. The need for matching manually computed firing solutions using ballistic tables and the results of the fire control systems changed the way to develop the software to build these tables. Contemporary approach became using the same ballistic kernel and so the same fire control inputs for both fire control software and ballistic table software. International effort in NATO for building a shareable and reusable ballistic kernel come up with a product called NATO Armament Ballistic Kernel (NABK-previously NATO Artillery Ballistic Kernel) (NABK).

With NABK, methodology for developing ballistic table software evolved to using not any other software but a NATO standard ballistic kernel that implements NATO wide agreed Firing Solution Algorithms and Trajectory Model. This approach guaranteed the exact match of computer and manual solutions of fire control problem. This paper presents NABK based ballistic table toolkit development effort in Defense Industries Research and Development Institute of Scientific and Technological Research Council of Turkey (TUBITAK-SAGE).

BALLISTIC TABLES, AN HISTORICAL PERSPECTIVE

Before going further, in this section, three different kinds of ballistic tables that can be produced by the developed software as parts of tool kit, will be presented. Graphical firing tables, tabular firing tables and bombing tables will be introduced.

Graphical Firing Table (GFT) was created due to the need of computing the firing data with ease. Because of slide rule form, reading and interpolating data is much easy. GFTs are made in two parts which are the wooden base and the cursor (See Figure 1). The rule is a rectangular base part of the GFTs generally printed both sides with one or more scales. The cursor slides on the rule. It is a transparent plastic component and to read firing data a hairline is carved in the plastic. There are three basic GFT formats which are low-angle GFTs, high-angle GFTs, and shell illuminating GFTs, applied to all weapons [8][1]. In the scope of this research two of these formats are carried out. Low-angle GFTs and high-angle GFTs.

Tabular firing tables basically contain the “tabulated” form of all necessary data which are essential to achieve a hit of the target under any condition [7][2]. For each projectile/charge/gun combination, fire commands under standard conditions (elevation and azimuth of the gun and the fuze setting) are provided as well as correction factors which are used to account for the non-standard conditions such as the wind, temperature, pressure, muzzle velocity, rotation of the earth, firing latitude and so on. Then the fire commands under any circumstance can be obtained by using linear corrections to the standard conditions using the firing tables. Although they were the primary tool for computation of fire commands until late 20th century, tabular firing tables are now used as back-up for the fire command and control computers.

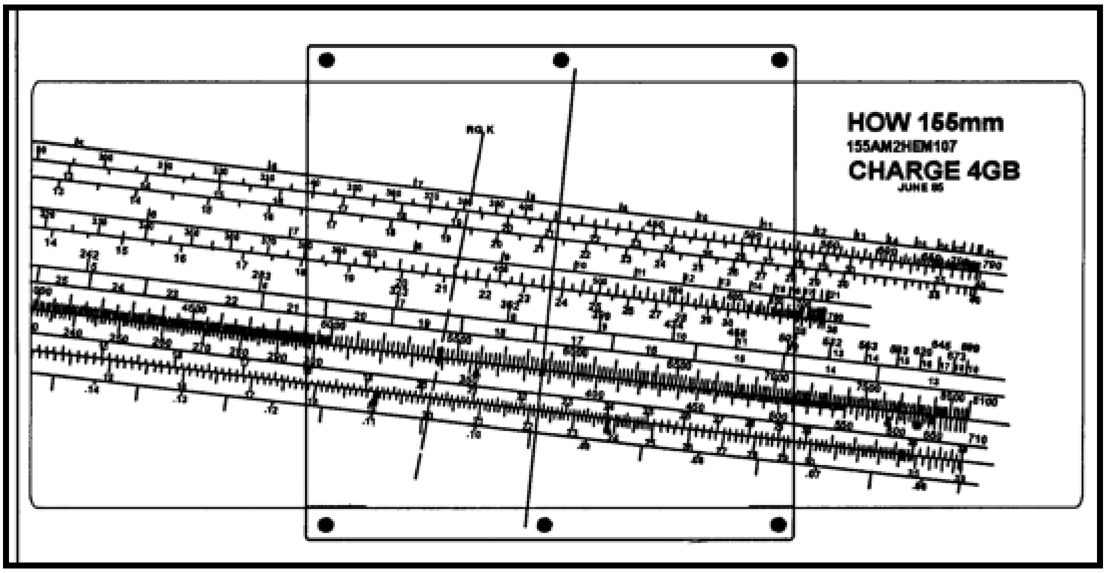

Figure 1. A Graphical Firing Table Illustration

Bombing tables on the other hand, are used mainly for mission planning before air operations. Mission planning basically results the release conditions for the bombs (i.e; approach altitude, release velocity and angle), minimum safe escape distance for the aircraft, fuze setting, and expected impact pattern. Tables can be separated -in terms of release conditions and maneuvers that has to be performed after releasing the bomb- into four parts; Level/Dive, Loft, Ripple Release and Safe Escape.

BALLISTIC TABLE SOFTWARE DEVELOPMENT, CONTEMPORARY APPROACH

Ballistic tables have been produced by using computers until the development of very first computer ENIAC [9][3]. Remembering the history, during World War II, a large number of female mathematicians were employed as "computers" to perform calculations necessary to create firing and bombing tables. The need to perform the calculations more quickly prompted the development of the ENIAC, the world's first electronic digital computer, in 1946 [5][4].

Ballistic tables depend on the concept of tabulating the firing solutions for standard conditions with the effects of non standard conditions on these solutions. Firing solution basically contains the information about how to launch the munitions. This information can contain dive angle and true air speed of a bomber aircraft or azimuth and elevation of a howitzer with the propelling charge to be used.

To answer the question “How to launch the munitions to hit the target?”, one will first need a means to compute how the munitions flies when launched in specified conditions. This means is called Trajectory Model. Then using Firing Solution Algorithms, “How to launch the munitions to hit the target?” question can be answered. Firing Solution Algorithms may consist of various iteration algorithms to converge to the target position. Ballistic Table Algorithms either uses Firing Solution Algorithms to compute how to launch to hit the specified target or uses Trajectory Models directly to compute where the munition hits when it is launched at standard conditions and lists the results. The results as an example can be listed according to range or aircraft true air speed. Then by perturbing the standard conditions, Ballistic Table Algorithms compute the effects of non standard conditions to the firing solution. For example air temperature is increased in some percent and the effect is tabulated as the effect of temperature on firing solution. This functional hierarchy of ballistic table software is depicted below in Figure 2.

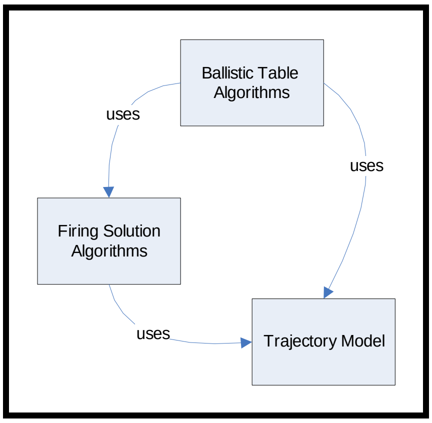

Figure 2. Ballistic Table Software Functional Hierarchy

With the introduction of computers in the battlefield, Trajectory Models and Firing Solution Algorithms were started to be used in technical and tactical fire control systems. Since the manually computed firing solutions using ballistic tables and the results of the fire control systems should match, the concept of using same piece of software that implements Firing Solution Algorithms and Trajectory Models both for fire control systems and ballistic table software emerged. This piece of software was named as ballistic kernel.

In mid 90’s, NATO Armament Ballistic Kernel (NABK-previously NATO Artillery Ballistic Kernel) project, with the scope of being joint fire control solution for different armaments including light arms, mortars, artillery and rockets, was launched. This kernel will further be introduced in the proceeding section. With this kernel, which is being used in firing control systems of many NATO Nation's fire control system and ballistic table software, contemporary methodology for developing ballistic table software evolved to using not any other software but a NATO standard ballistic kernel that implements NATO wide agreed Firing Solution Algorithms and Trajectory Model [10][5]. This design methodology which is also presented below in Figure 3, guarantees the exact match of computer and manual solutions of fire control problem.

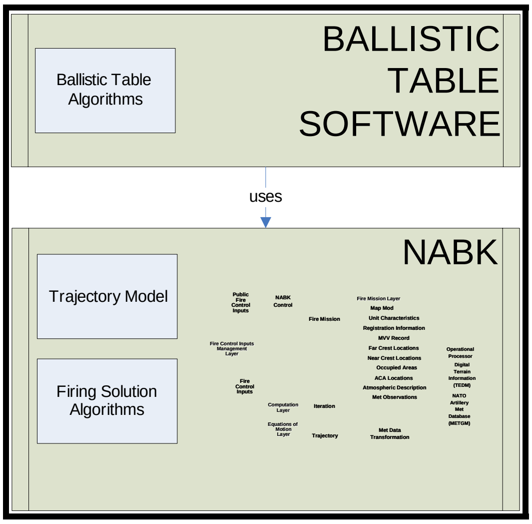

Figure 3. Contemporary Ballistic Table Software Design

NABK AS THE BALLISTIC KERNEL

The story of NABK started in 1990's when Firing Tables and Aeroballistics Branch of Aberdeen Proving Ground (USA) and Forsvarets Forsningsinstitutt (Norway) came together in order to develop a new ballistics solver for the fire control problem of the field artillery [1][6]. This joint effort was welcomed with great interest by the members of NATO Land Group 4, Sub Group 2 (LG4/SG2). As a result of this interest, first version of NABK was born, programming language being the ADA95.

Shareability and reusability were and still are the primary concerns for the development of the NABK. During years of progress, each country that uses NABK has added new modules to meet their requirements. As a result NABK, originally named as the NATO Artillery Ballistic Kernel, became an integrated platform for all surface to surface weapons and changed its name to the NATO Armaments Ballistic Kernel.

As depicted above in Figure 4, NABK consists of 4 primary layers; Fire Control Inputs Layer, Equations of Motion Layer, Computation Layer and Fire Mission Layer.

Fire Control Inputs Layer provides access to all the necessary aerodynamic and ballistic data which is used in the computation of the fire command. Equation of Motion Layer computes the trajectory of the projectile using either point mass, modified point mass, guided point mass or 5DOF models depending on the type of the projectile. Computation Layer on the other hand, is responsible for the iterative solution of the fire control problem. And lastly Fire Mission Layer is used in order to account for the meteorological conditions, muzzle velocity variations, air corridors and safety issues, near and far crest violations and so on.
This layered structure of NABK allowed it to be a joint ballistic solution platform to the wide range of problems with different and challenging requirements such as flight simulations, firing table generations, technical and tactical fire control.

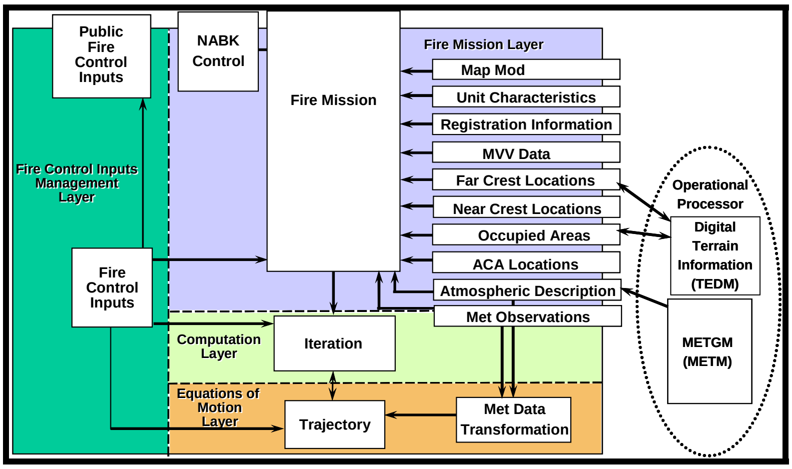

Figure 4. NABK Layers

FIRE CONTROL INPUTS, AN OVERVIEW

Fire Control Inputs (FCI) are used to relate the performance of the weapon system to the measurable initial conditions in order to hit a given target with expected accuracy. They are used by Trajectory Model, in our case NABK, to compute both trajectory and firing solution.

FCI's contain large number of data entries which are related to the different aspects of weapon system, propellant charges and terminal phase ballistics and obviously projectile ballistics itself. The required data for Modified Point Mass Model defined in STANAG 4355 can be divided into following main categories;

Physical Data: Reference mass of the fuzed projectile, square weight, reference fuze weight, center of gravity, axial inertia and spin rate

Internal Ballistic Data: Standard muzzle velocity, muzzle velocity correction for projectile mass, muzzle velocity correction for propellant temperature, and muzzle velocity loss for tube wear
External Ballistic Data:
Aerodynamic coefficients: Zero and quadratic yaw drag coeff., lift and cubic lift force coeff., overturning and cubic overturning moment coeff., magnus force coeff., and spin damping moment coeff.
*Fitting Factors and Correction Factors:*Form factor, drag factor, lift factor, yaw drag factor, and magnus force factor, time of flight correction, drift correction factors
Probable Errors: Probable errors in range to impact, deflection at impact, range to burst and height of burst [6][7].

BALLISTIC TABLE TOOLKIT

Three ballistic table software have been developed as a part of envisioned NABK based ballistic table toolkit. These products are tabular firing table software, bombing table software and graphical firing table software. All three were developed using the design methodology discussed in the previous sections. Trajectory model and firing solutions algorithms of NABK were used by the ballistic table algorithms specific to each product and those ballistic tables were produced.

Firing Table Software was built on release 3.0 of NABK. This software produces text based ballistic table output conforming STANAG 4119 Requirements for the Development and Formatting of Tabular Firing Tables for artillery projectiles. The design schema of this software is given below in Figure 5.

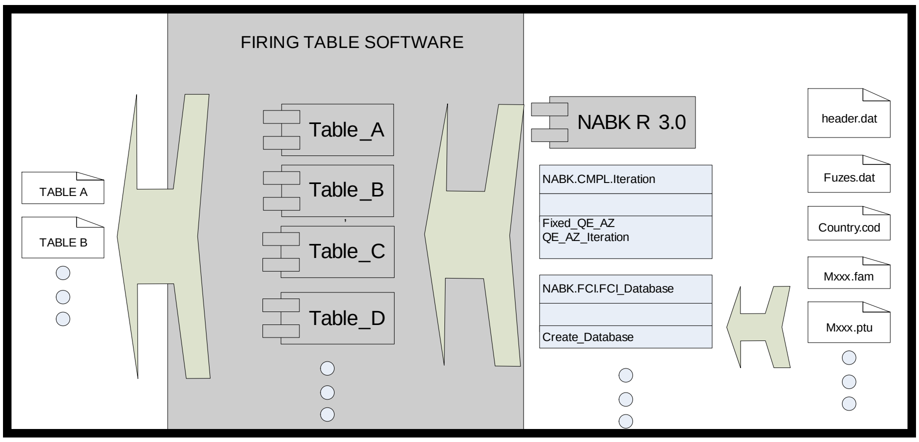

Figure 5. Firing Table Software Design Schema

Bombing Table Software was developed on the 7th release of NABK. This software was developed to compute bombing tables for level, dive, loft, ripple bombing scenarios and safe escape tables for a cluster bomb. Figure 6 depicts the design schema for Bombing Table Software.

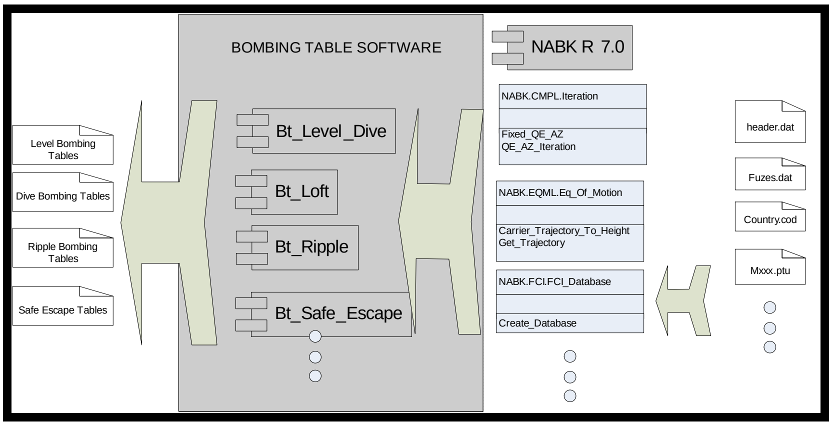

Figure 6. Bombing Table Software Design Schema

Graphical Firing Table Software as the last product of the toolkit was developed to produce low and high angle graphical firing tables. This product was also built as an operational processor of the 7th release of NABK. Different from previous two products this software produces a vector image of graphical firing tables. The packages that use NABK implements the ballistic table algorithms and produces a text output which is then be post processed by AutoLISP scripts of AutoCAD to generate a graphical output. Design schema of this product is presented below in Figure 7.

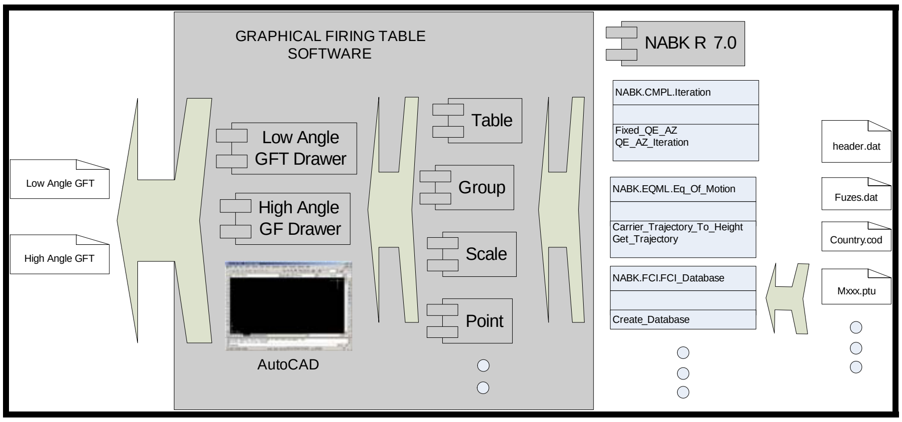

Figure 7. Graphical Firing Table Software Design Schema

CONCLUSION

TUBITAK-SAGE as presented in this paper is reusing NABK for all its ballistic table software development projects under the supervision of Turkish Land Forces Command. This design schema not only guaranties the results of ballistic tables match the operational systems but also minimizes the development efforts by enhancing the reusability. Furthermore, same FCI data is used in operational fire control software and ballistic table software which increases the operational ease.

Collaborative NABK project which is currently carried out under Land Capability Group 3 Sub Group 2 of NATO Army Armaments Group maintains the NABK software. With the active participation of this project, maintenance costs of developed ballistic table software are decreased. Problems of the current code and new capabilities are added to this shareable product is being maintained in a collaborative manner by all participants/users of the project.

REFERENCES

[1] Field Manual 6-40, “Tactics, Techniques and Procedures for Field Artillery Manual Cannon Gunnery”, Marine Corps Warfighting Publication, FM 6-40/MCWP 3-1.6.19, 1996.

[2] STANAG 4119, “Requirements for the Development and Formatting of Tabular Firing Tables”, Draft Edition 2, 2002.

[3] Harry L. Reed, “Firing Table Computations on the Eniac”, Proceedings of the 1952 ACM National Meeting, Pittsburgh, 1952.

[4] Neeraja Sankaran, “Looking Back At ENIAC: Commemorating A Half-Century Of Computers In The Reviewing System”, The Scientist 1995, 9(16):3.

[5] STANAG 4355, “The Modified Point Mass and Five Degrees of Freedom Trajectory Models”, Edition 4.

[6] Sowa J.A., Lieske, R.F., Matts J.A., Miller J.L., “*Ballistic Kernels Sharable Fire Control and Ballistic Simulation Software, Firing Tables and Aeroballistics Branch Information Report*”, FTAB-IR-32, 1997

[7] STANAG 4144, “Procedures to Determine the Fire Control Inputs for Use in Indirect Fire Control Systems”, Edition 2, 2005.