Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 05, May-2018

A Real Time Control System Simulation Model Based on LabVIEW Graphical Programming Language

M. Ahmed and H. S. Taha
Military Technical College
Cairo, Egypt

A. Saaid and M. Tarek
Military Technical College
Cairo, Egypt

***Abstract*— Embedded control system typically requires an architecture that is designed for reliability and deterministic performance. As a result, real time control system requires processes dedicated to system status monitoring, error handling and watchdog timers. This paper present a six degree-of-freedom (6-DOF) missile model implemented using structured graphical model building blocks. This model illustrates how a graphical representation of a missile model leads to a much clearer representation of the original mathematical model. The LabVIEW graphical programming software and collocate the Real Time/ (6-DOF) model software module are used to establish the missile behavior and nature which integrate the software and hardware equipment. LabVIEW includes tools for low-level system debugging and precise execution timing so that it can increase the flexibility and development of the missiles. Real time simulation is growing in its importance to the success of these systems. A validation of the developed model design is performed through software in loop simulation (SIL). This is to ensure the safety and economics, as the real flight test cannot easily be done. A missile simulation model of the flight control systems have been developed to be used as DLL modules. A co-simulation between these modules is developed in LabVIEW programming environment, where they are tested and verified using model in the loop test.**

Keywords*— *Embedded control system, simulation model, graphical programming, 6-DOF, software in the loop (SIL)

I. INTRODUCTION

This paper presents a simulation for a complete nonlinear 6-DOF missile simulation model for a missile using implemented structured graphical blocks. The profile of the missile trajectory from launch to impact point can be divided into three dependent phases of flight: boost phase/ powered-flight phase, free flight phase and re-entry flight phase [1].

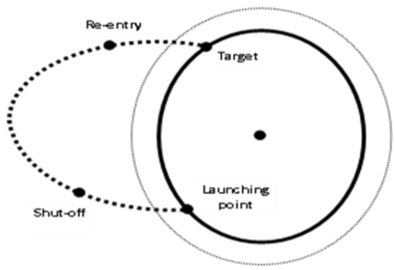


Fig.1. missile trajectory phases

The model is a simulation for the boost phase/ powered-flight phase only, which lasts from the launching time to missile motor thrust burnout, because the guidance of the missile occurs entirely during this phase. This model consists of mainly of six hierarchy graphical blocks. The model contains simple simulated blocks for a real ballistic missile that can be replaced simply and quickly by other blocks using drag and drop features for the development and enhancement purpose.

For investigating the performance of the system and keeping with the requirements for a continued development, the model is structured as a series of modules. These modules can be individually developed for instance; there are airframe structure module, thrust variation module, weight data module, variation of missile mass center module, variation of mass center of oxidizer and fuel tanks, gravity module, earth module, variation of inertial moment module, and atmospheric data module. LabVIEW™ is a modeling and implementation graphical environment, which is more useful for those who would like to take advantage of the benefits from implementations in reconfigurable hardware. Graphical design and programming can enhance development experience and productivity by making graphical blocks design with defined-user libraries and offering a powerful integrated modeling, analysis, and deployment environment [2], also it provides a low time cost for developers by achieving the usability of previously implemented design blocks and libraries. This paper is organized into 5 sections. Section 2 illustrates the real time flight control system. Section 3 introduces the LabVIEW graphical programming environment. Section 4 presents the proposed mathematical 6-DOF simulation model, the function of each graphical block for the model and the benefits of graphical programming. Simulation results and analysis are discussed in section 5. Paper terminated with the conclusion in section 6.

II. REAL TIME EMBEDDED FLIGHT CONTROL SYSTEM

In this section a surveying for the real time operating system (RTOS) will be presented such that the target machine will include a VxWorks RTOS. A brief of the flight control system types and its main components for missile system is illustrated. The real time flight control system is the most significant issue for a missile in the guided active phase and it will be the point of interest in this paper.

A. Real time operating system (RTOS)

A real time operating system is the ability of the operating system to provide a required level of service in a bounded response time [3]. RTOS are classified into two categories [4]: Hard real time: Occurring of system failure if the required task had not been achieved before the deadline time neglecting the degree of tolerance. Soft real time: Missing the deadline time from time to time will not cause a system failure and the system's quality is acceptable. The features of RTOS are multithreading, short latencies, task priority and inter-tasks communication [3, 4]. Limitations of RTOS are [3]: It can be costly, RTOS are complicated and amount of processor cycles can be consumed, and without round-robin scheduling RTOS doesn't support multitasking such that multiple tasks will be sequentially processed in a circular manner such as 1, 2, 3, 4, 1, 2, 3, 4…. [4]. Examples for RTOS: VxWorks, Windows CE, QNX Neutrino, RTAI, LynxOS, Micrium µC/OS-II, Jbed and Symbian [3, 4].

B. Flight control system simulation model types and components

In this section, the flight control system simulation model types and its components will be introduced. The flight control system is consists of four main basic elements as shown in Fig.2 [5]. The autopilot unit takes its inputs from the guidance law and the inertia measuring unit (IMU) that can sense the inertial motion of the missile and outputs the processed commands such as commanded deflection angles to the actuation system in which these commands are turned into physical motion to be processed by the missile dynamics.

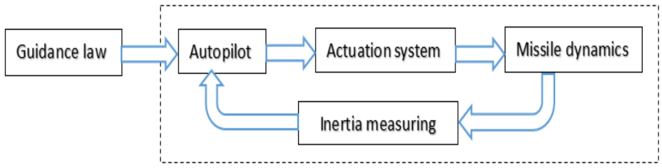


Fig. 2. Flight control system sub-modules

Several factors are affecting the flight control system implemented on a particular missile such as cost, packaging constraints, and the system requirements and its mission. Flight control systems are classified into three major types [5]:

1) Acceleration Control System: This type of control systems depends on the sensed data of the rate of the pitch angle of the missile and its acceleration by the IMU. The autopilot receives the sensed data as an input and produces the deflection angles to the actuators that in turn moves the fins to control the movement of the missile body. During the motion of the missile an angle of attack (AOA) is produced which in turn produces aerodynamic lift to accelerate the airframe.

2) Attitude Control System: Another control system type that depends on the missile attitude. Such that it controls the thrust deflection angles. It is used in this work.

3) Flight-Path Angle Control System: an autopilot that can be used to track flight-path angle commands using thrust-vector control. This type of system assumes that aerodynamic forces are small and hence applies for exoatmospheric flight or for endoatmospheric flight when the missile speed is low.

The main components of the control system can be classified by the following modules [5]:

1) Airframe Dynamics: the objective of the flight control system is to force the missile dynamics to track the input command. The dynamics of the airframe are governed by fundamental equations of motion, with their specific characteristics determined by the missile aerodynamic response, propulsion, and mass properties.

2) Actuator: the deflection commands produced from the autopilot is converted to physical motion by the actuation system to control missile motion.

3) Autopilot: the autopilot process the commands from the equation of motion and the sensed data from the IMU to produce the deflection commands for the actuation system.

4) Inertial Measurement Unit: The IMU includes accelerometers and gyroscopes [5] to senses the missile angular velocity and the acceleration to the autopilot.

III. THE LABVIEW GRAPHICAL PROGRAMMING ENVIRONMENT

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) was introduced as a scientific software system by National Instruments Corporation (NI) around 1986. Most of programming languages such as C, C++, JAVA... etc. are textually oriented but LabVIEW uses a graphical programming language called G which is a programming language based on dataflow model of computation [2]. It is program structure are composed of Virtual Instrument (VI) that supports important programming language features such as modularity and hierarchical concepts. VI is the combination of a user interactive front panel that controls the provide input and display output, and a background block diagram that represent the graphical design of the program with the necessary primitive functions as shown in Fig.3.

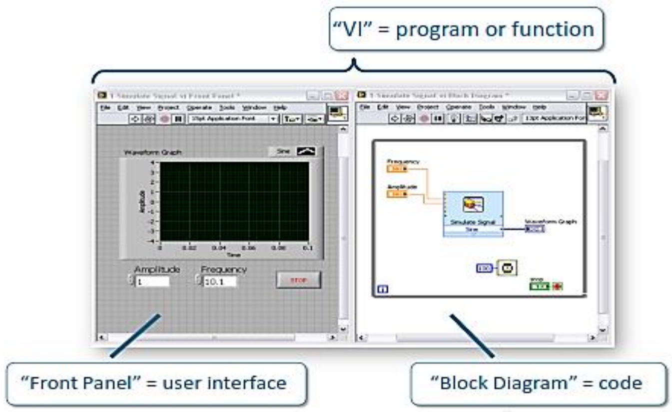


Fig.3. LabVIEW development environment

The LabVIEW Simulation Module provides a means of representing dataflow logic in control block diagram form typical in the design of control systems, and includes numerical ordinary differential equation solvers for simulation or real time implementation [6]. One of the main advantage of the LabVIEW is the usability of codes written in C, C++ and JAVA

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 05, May-2018

programming languages through the link mechanisms used for external code or software, such as DLL and DDE (shared libraries) and active-X. It also has powerful functions for Internet, and support of common network protocol. This work make use of these advantages by achieving code usability of a simulation model written in C-programming language. Deterministic is the most important concept for hard real time applications that can be achieved by the LabVIEW Real Time. The LabVIEW RT architecture consists of the following three components [7]: LabVIEW software, the Real Time Development System and the Real Time Engine as shown in Fig 4.

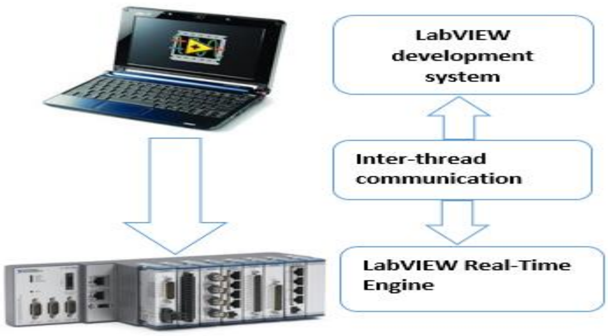


Fig. 4. Components of LabVIEW Real-Time Architecture

CompactRIO is an example of Real Time engine based on reconfigurable I/O (RIO). It includes of three components [7] as shown in Fig. 3:

1) Processor to execute LabVIEW Real Time applications for reliable real time operating system (VxWorks RTOS or RT-Linux).

2) Reconfigurable embedded chassis with programmable FPGA core that can be accessed and configured using LabVIEW graphical development software.

3) Hot-swappable industrial I/O modules with built-in signal conditioning for direct connection to a variety of sensors and actuators.

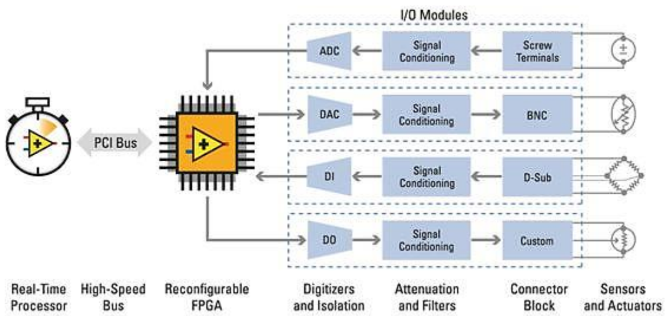
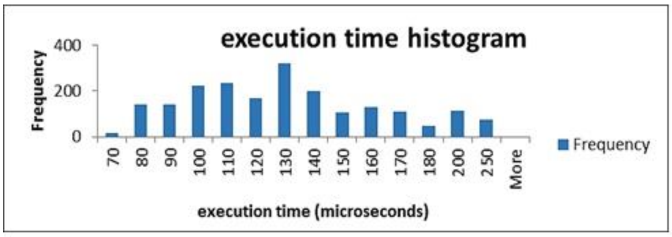
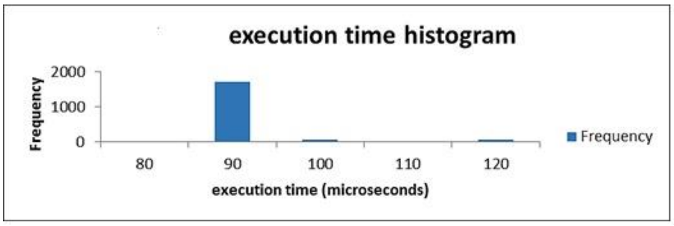


Fig.5. Components of CompactRIO

Development of the simulation model under real-time operating system ensures the determinism which can't be achieved if a non-real-time operating system is used as many factors are affecting it such as latency, scheduling algorithm, jitter …etc. for example: Jitter is very high under non real-time operating systems such as windows but it is very low in real-time operating systems such as VxWorks. Fig.6. shows the jitter of a certain task under both types of operating systems.



(a) Jitter under non real-time O.S



(b) Jitter under real-time O.S

Fig.6. Execution time histogram

For real-time O.S the task under test has execution time of 90µs but it rarely takes from 100µs to 120µs for about 2000 iterations. This low jitter means that the time analysis of the simulation model under VxWorks RTOS will be very precise as a result VxWorks is used for real time applications. On the other hand the jitter for non-real-time windows OS is measured using the NI Veristand testing tool [8] that is able to measure time in microseconds under windows platform. While the CompactRIO real-time target has the ability to measure execution time in microseconds. In this work a CompactRIO 9074 with VxWorks real time operating system have been used for the simulation model development in real-time with rate of 400 HZ to achieve determinism and to be used for developing hardware in the loop test (HIL) for the autopilot controller validation and verification to reduce both of risk and cost.

IV. PROPOSED 6-DOF MATHEMATICAL MODEL SIMULATION FOR BALLISTIC MISSILE

The 6-DOF simulation coded from the proposed missile system specifications can be used to verify the predicted performance of the system once the full aerodynamic characteristics, functional algorithms, and expected noise sources are included. A 6-DOF simulation can be used to design initial flight tests to exercise various missile subsystems at particular operating conditions. Similarly, after flight testing of the system has begun the 6-DOF simulation parameters can be validated against the measured telemetry. The simulation can be used to assess the risks associated with modifications to the missile or to assess its performance against a new threat. For an operational system, a simulation identifies and illuminates key sensitivities in the existing hardware. The effect on system responsiveness and lethality of potential hardware or software changes can be characterized. Understanding this system behavior allows the engineer to revise the missile specifications, if necessary, for future deployments. Finally, it can be used to generate operational guidelines for deployment and firing protocols. Fig. 7 illustrates the representation of the missile developed 6-DOF orientation in body coordinates (x, y, and z). Moreover, it also represents the symbols ϑ, ψ, φ are the pitch, yaw, and roll angles rates of the missile respectively. α is the angle of attack, β is the slip angle, Ψ is the trajectory turning angle and θ is the missile flight path angle, V_m is the velocity vector. The illustration of the most important parameters for the missile simulation model will be introduced in the experimental results section.

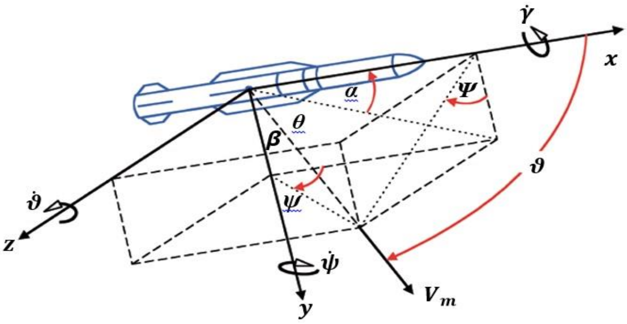


Fig.7. The missile 6-DOF coordinate system (x, y, and z)

The dynamical equation of motion of missile center of mass [9]:

$$\begin{bmatrix} dV_{gxl}/dt \\ dV_{gyl}/dt \\ dV_{gzl}/dt \end{bmatrix} = \frac{1}{m} \boldsymbol{C}_b^l \left(\begin{bmatrix} A_{aero_x} \\ A_{aero_y} \\ A_{aero_z} \end{bmatrix}_b + \begin{bmatrix} P_{thr_x} \\ P_{thr_y} \\ P_{thr_z} \end{bmatrix}_b \right) + \boldsymbol{C}_e^l \left(\begin{bmatrix} g_{xe} \\ g_{ye} \\ g_{ze} \end{bmatrix} + 2\Omega \begin{bmatrix} -V_{gye} \\ V_{gxe} \\ 0 \end{bmatrix} - \Omega^2 \begin{bmatrix} x_e \\ y_e \\ 0 \end{bmatrix} \right) \quad (1)$$

Dynamics equation of rotational motion of the missile [9]:

$$\begin{bmatrix} d\omega_x^b/dt \\ d\omega_y^b/dt \\ d\omega_z^b/dt \end{bmatrix} = \begin{bmatrix} M_x/I_x \\ \left((I_c - I_x)\omega_z^b\omega_x^b + M_y\right)/I_c \\ \left(-(I_c - I_x)\omega_x^b\omega_y^b + M_z\right)/I_c \end{bmatrix} \quad (2)$$

Aerodynamic forces and moments [9]:
Aerodynamic force: One way to resolve aerodynamic force acting on the missile in air-stream axes with the corresponding coefficients C_D, C_L, C_C:

$$D = (1/2)C_D \rho V_a^2 S \quad (3)$$
$$L = (1/2)C_L \rho V_a^2 S \quad (4)$$
$$C = (1/2)C_C \rho V_a^2 S \quad (5)$$

Aerodynamic moment: Components of aerodynamic moment about center of gravity in body frame are expressed in terms of moment coefficients such as [9]:

Rolling moment $M_x = (1/2)C_{mx}\rho V_a^2 Sl$ (6)
Yawing moment $M_y = (1/2)C_{my}\rho V_a^2 Sl$ (7)
Pitching moment $M_z = (1/2)C_{mz}\rho V_a^2 Sl$ (8)

The 6-DOF model is developed using the graphical programming language LabVIEW that provides a very important abstraction mechanism that helps to increase the readability and understandability of a program. The development and implementation of the model are performed in the host and real time target machines. The diagram of a VI can optionally have an icon that suggests its functionality. The icon associates a number of the interactive control inputs with wiring connections on the icon so that data can be passed to and from the VI diagram [7]. This allows easy re-use of LabVIEW modules and supports the hierarchical composition of structured dataflow diagrams. Complex diagrams can be made more manageable by breaking them into modular pieces. It seems that this approach seems very natural and intuitive to most users. The mathematical simulation model is simulated such that modularity, readability and structure programming concepts are achieved to facilitate system development and enhancement. The general structure block diagram is illustrated as shown in Fig. 8. Different block functions are explained with the important input and output parameters that have the most powerful effect on the model.

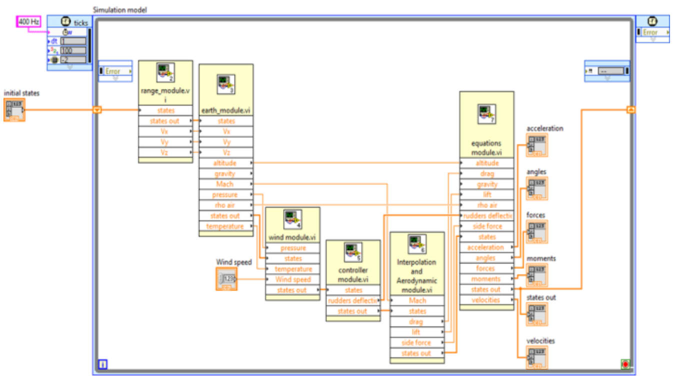


Fig.8. General Block diagram for 6-DOF of the missile

The description for the main six developed function modules of the 6-DOF mathematical model can be presented as follows:

1) Range Control module: The range control module is responsible for calculating the shutoff time of the missile engine based on the missile velocities. A certain range is settled by the user in the pre-launch phase. This module is responsible for calculating the required range according to the three velocities of the missile.

2) Earth model module: All data about the earth and the effect of the earth rotation on missile parameters are included in the earth model. It entirely contains a graphical block responsible for calculating the Mach number and the pressure that affect the missile according to the altitude reached. The gravity of earth is also included in that model where the gravity across different height is calculated; Carioles effect is taken into consideration.

3) Wind module: Disturbance is an important factor that must be studied well, where it affect the angle of attack of the missile by having a data for the wind speed, atmospheric

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 05, May-2018

pressure and temperature, then calculating the effect of them on missile velocity and angle of attack.

4) Controller module: The actuator system is presented including the rudders and servo mechanical system, autopilot mathematical model and the generated trajectory profile are included for controlling the missile body. The output of the model is the fin commands that are sent to the Fins/Actuators Model. The main task of the proposed real time control system is the calculation of the proportional integral derivative (PID) as follows [9].

$$\vec{\delta}(t) = K_p\vec{e}(t) + K_I \int_0^t \vec{e}\,(\tau)d\tau + K_d \frac{d}{dt}\vec{e}(t) \qquad (9)$$

Where:
$\vec{e}(t)$ is the error vector between the required $\boldsymbol{\vartheta_p}$, $\boldsymbol{\vartheta'_p}$ and sensed angles, rates and Vz velocity. K_p, K_I, K_d are the gains, tuning parameters of the missile. $\vec{\delta}(t)$ is the output commands for the servo system angle the rudder deflections.

5) Interpolation and Aerodynamic module: Mathematical interpolation is used for finding new points for a given range of non-continuous points, so new points are generated by this module given two successive Mach number and producing the drag, lift and side forces affecting the missile.

6) Equations module: All produced parameters from previous blocks are the input for the last block which contains all mathematical equations to produce forces needed to move rudders, moments affecting the body, angles and rates of the moving body.

V. RESULTS AND SIMULATION

The resulting model is graphically organized in a way that lends itself to intuitive navigation, even by those who did not participate in its development. Moreover, one of the most power features in the LabVIEW is model analysis where LabVIEW provide very precise tool to measure the execution time in one microsecond precision. The real time control system development on the host machine and real time engine (CompactRIO) could be analyzed to measure the execution time for the effective functions in the model, so these results are available for enhancement purpose. Once the simulation is assembled, the plotting capability and the ease in which constants could be changed made it quick and easy to debug and fine-tune the model. A validation of the developed model design is performed through software in loop simulation (SIL) within the LabVIEW platform. This is to ensure the safety and economics, as the real flight test cannot easily be done. As a result of completes simulation run, an illustration of the most important parameters for the missile simulation model will be introduced. The following simulated results represent the active guided phase of the ballistic trajectory only.

Angles of attack $\boldsymbol{\alpha}$ and slip angle $\boldsymbol{\beta}$ are represented as shown in Fig. 9. Fig.10 represents the simulated Pitch, Yaw, Roll angles and angle rates (IMU and Rate Gyro) with Pitch profile (command) angle and its angle rate ($\boldsymbol{\vartheta}$, $\boldsymbol{\psi}$, $\boldsymbol{\varphi}$, $\dot{\boldsymbol{\vartheta}}$, $\dot{\boldsymbol{\psi}}$, $\dot{\boldsymbol{\varphi}}$, $\boldsymbol{\vartheta_p}$ and $\boldsymbol{\vartheta'_p}$) where the left column depicts angles and right column introduces the angle rates. Fig. 11 demonstrates the three simulated velocities ($\boldsymbol{V_x}$, $\boldsymbol{V_y}$ and $\boldsymbol{V_z}$) of the missiles in the body axis. Fig. 12 illustrates the commanded control rudder deflection angles ($б_1$, $б_2$, $б_3$ and $б_4$) which are implemented in the PID controller module based on the simulated angles and rates.

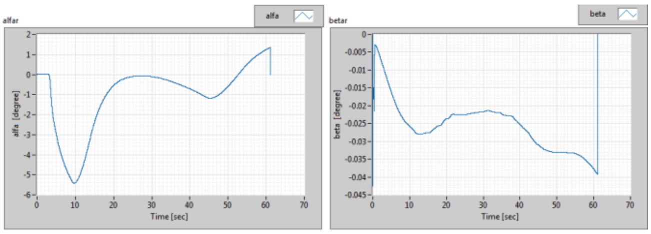


Fig. 9. Angles of attack α and slip angle β

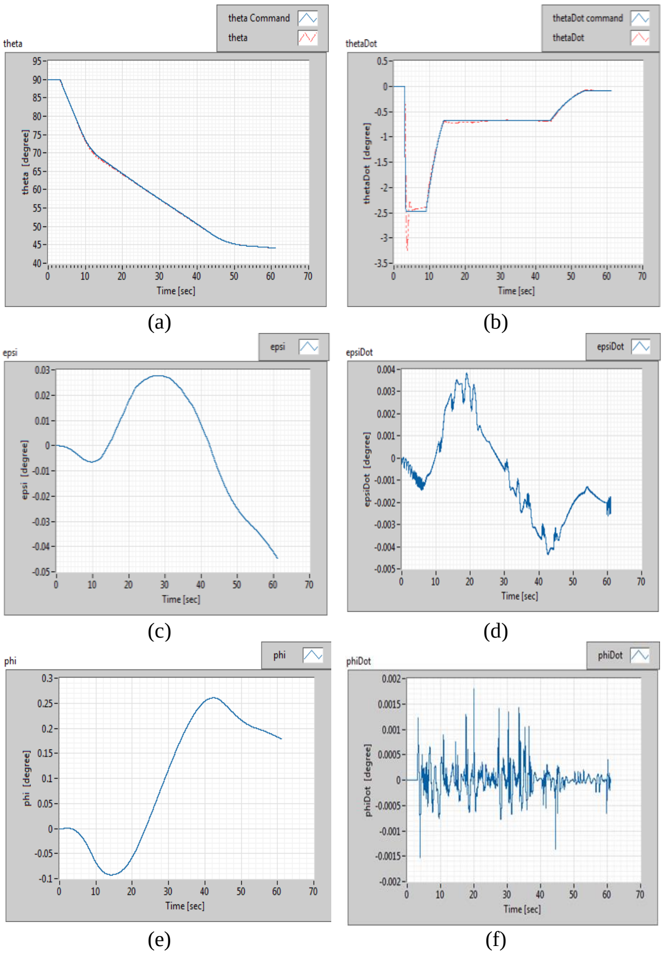


Fig.10. Pitch, Yaw, Roll angles and angle rates with Pitch profile (Command) angle and its angle rate: ϑ, ψ, φ, $\dot{\vartheta}$, $\dot{\psi}$, $\dot{\varphi}$, ϑ_p and ϑ'_p

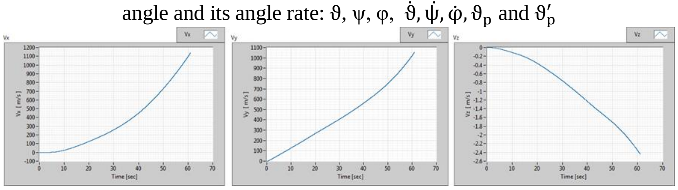


Fig. 11. The Three velocities in body axis V_x V_y V_z

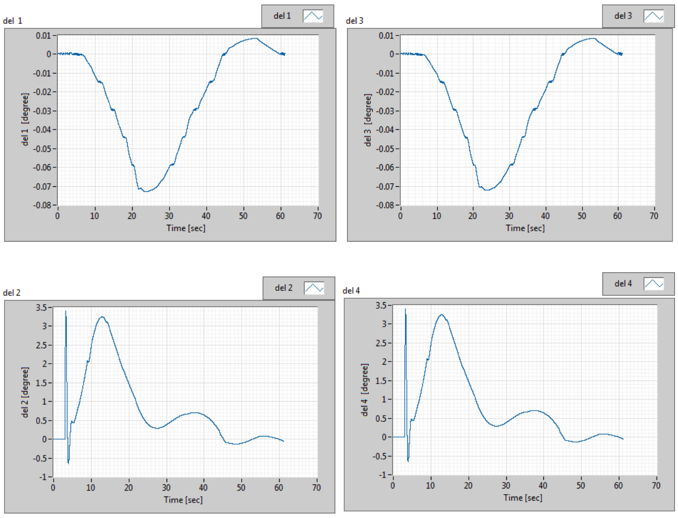


Fig.12. Rudder deflection ($б_1$, $б_2$, $б_3$ and $б_4$)

VI. CONCLUSIONS

In this paper, the LabVIEW, graphical programming language provided an environment to develop the 6-DOF Missile model. A real time flight simulation was developed as a benchmark to demonstrate the value of graphical modeling. With its many point and click features, model development became easier and faster either by developing new model blocks or utilizing existing blocks from the LabVIEW modules. The resulting model is graphically organized in a way that lends itself to intuitive navigation, even by those who did not participate in its development. Once the simulation is assembled, the plotting capability and the ease in which constants could be changed made it quick and easy to debug and fine-tune the model. Individual deterministic analysis and visualization is easily completed in LabVIEW. Finally, software in the loop simulation was performed to analysis the controller quality. Thus, with the advantage of rapid prototyping using LabVIEW, it’s efficient and convenient to design a real time simulation system with low cost of time and well expandability.

VII. REFERENCES

[1] H. L. Shu, "Ballistic Missile and Launch Vehicle Design", School of space technology, China, BUAA, 2002.

[2] G. Wang, T. N. Tran, and H. A. Andrade, "A graphical programming and design environment for FPGA-based hardware", in Field-Programmable Technology (FPT), 2010 International Conference on, 2010, pp. 337-340.

[3] P. Hambarde, R. Varma, and S. Jha, "The Survey of Real Time Operating System: RTOS," in Electronic Systems, Signal Processing and Computing Technologies (ICESC)”, 2014 International Conference on, 2014, pp. 34-39.

[4] K. V. Prashanth, P. S. Akram, and T. A. Reddy, "Real time issues in embedded system design", in Signal Processing And Communication Engineering Systems (SPACES), 2015 International Conference on, 2015, pp. 167-171.

[5] P. B. Jackson, "Overview of missile flight control systems", Johns hopkins APL technical digest, vol. 29, pp. 9-24, 2010.

[6] A. Gancarczyk, "Real Time Measurement Systems Based On LabVIEW Graphical Environment", XI International PhD Workshop, 2009.

[7] R. Jamal and L. Wenzel, "The applicability of the visual programming language LabVIEW to large real-world applications," in Visual Languages”, Proceedings., 11th IEEE International Symposium on, 1995, pp. 99-106.

[8] Z. Isaac, "Comparison between dSPACE and NI systems based on real-time intelligent control of a teleoperated hydraulic servo system", 2014.

[9] A. H. Mohamed, "Estimation Techniques for Low Cost Inertial Navigation Systems", Chair of guidance, MTC, 2015.