Monte-Carlo Evaluation on Implementation of Autopilot HIL Simulation

Nithi Anukulsumpan
Defence Technology Institute
Ministry of Defence
Email: nithi.a@dti.or.th

Palis Ratanaswasd
Defence Technology Institute
Ministry of Defence
Email: palis.r@dti.or.th

Weerawat Khawsuk
Defence Technology Institute
Ministry of Defence
Email: weerawat.k@dti.or.th

***Abstract*—A simulation framework of a guided ballistic missile has been developed to aid the design and development of autopilot algorithm. In this framework, a hardware-in-the-loop (HIL) is utilized in the actuators that drive the control surfaces. The system dynamic engine is enhanced by receiving the position feedback of the actuator in real time allow it to be able to compute more realistic updates of system variables. A resistance torque emulator is introduced and deployed to further improve the realism of the dynamic engine. This allows the firmware control algorithm to better match the actual hardware in the real missile. An experiment is performed and results are used to compare with the simulation framework without HIL implementation.**

***Index Terms*—Simulation, Hardware-in-the-loop, Flight dynamics, Actuators, Canard**

I. Introduction

An autopilot is a critical component of a guided missile. It controls how a missile navigates to reach the target. In development process, the characteristics of control algorithms within an autopilot can be formulated based upon a mathematical model of the studied missile body. A simulation is further setup to verify this autopilot performance. Essentially, the simulation framework should contain the missiles dynamic model as well as other models such as forces, gravity, atmosphere, and etc. These models are needed for system variable computation. During simulation, the framework will step through iterations where the system variables are continually updated. Initial conditions are given to the framework to signal the starting point of a new simulation session. The simulation begins and continues until terminal conditions are met.

In a simulation, equations of motion are used to solve missile dynamics. Values of state variables are updated and monitored. Some of these variables can be mapped directly into sensor signals which are required by the autopilot during flight controls. The outputs from flight controls are then fed back to the simulation framework to update fin positions to new values. In other words, the autopilot reads sensor signals from the model, processes them, and computes fin control signals based on navigation commands. All these steps are the same process that the autopilot must go through in the actual missile.

Although a simulation is a typical method for testing and verifying functions of an autopilot, it does not always behave in the same manner as does the actual missile. If the autopilot is tuned to exactly match the simulation, problems may still arise during the actual flight due to differences between the mathematical model and the real system. It is a good idea to try to minimize the number of differences as many as possible. One possible solution is to replace some parts of the simulation framework with actual hardware. This yields a best match between the hardware conditions and the actual flight conditions during the simulation process. This method is known as the Hardware-In-the-Loop (HIL) simulation.

In this paper, we are describing a simulation framework used to simulate a ballistic missile of size 300mm in diameter. The control surfaces of this simulation framework are located at the forward location on the missile body as shown in Figure 1. The missile is driven by a solid propellant rocket motor. The burning rate and trust profile are built as a modeling part of the framework. An HIL is implemented into the autopilot.

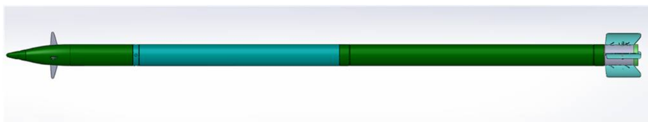

Fig. 1: A proposed configuration of the 300mm missile

This paper is organized as follows.

II. Simulation Framework

A block diagram of the simulation framework is shown in Figure 2. In this framework, a flight dynamic engine is created as a 6 Degree of Freedom (DOF) dynamic model similar to one described in [1]. The external display modules are added in order to visualize position and orientation of the missile. An autopilot receives flight information from the dynamic engine and outputs fin position commands to the fin controllers. The fin controllers then receives the commands and drives the fins to the required position. The actual fin positions are fed back to the dynamic engine. The dynamic engine uses the information as well as the information from state variables regarding the