Bulletin of Electrical Engineering and Informatics
Vol. 13, No. 5, October 2024, pp. 3121~3134
ISSN: 2302-9285, DOI: 10.11591/eei.v13i5.7798

Implementing trajectory correction strategy through model prediction control for flight vehicle missions

Herma Yudhi Irwanto[1,2], Purnomo Yusgiantoro[2], Zainal Abidin Sahabuddin[2], Romie Oktovianus Bura[2], Rika Andiarti[1], Idris Eko Putro[1], Oka Sudiana[1], Azizul Hanif[1]
[1]Research Organization of Aeronautics and Space, National Research and Innovation Agency of Indonesia, Jakarta, Indonesia
[2]Defense Science, Defense University of Republic of Indonesia, Jakarta, Indonesia

Article Info

Article history:

Received Oct 23, 2023
Revised Mar 23, 2024
Accepted Mar 28, 2024

Keywords:

Hardware-in-the-loop simulation
Model predictive control
Onboard flight control system
Software-in-the-loop simulation
Trajectory correction

ABSTRACT

Modeling a high-speed flying vehicle is imperative to ensure the success of vehicle development missions. Moreover, adherence to research protocols mandates a stepwise approach to testing the vehicle model, encompassing simulation trials using software-in-the-loop simulation (SILS), hardware-in-the-loop simulation (HILS), as well as diverse ground and environmental tests prior to flight testing. This study entailed a collaborative effort between MATLAB/Simulink and LabVIEW to seamlessly integrate the model developed in MATLAB/Simulink into LabVIEW for the implementation of model predictive control (MPC) strategy, aimed at trajectory correction (TC) missions for the vehicle. This MPC strategy was directly applied to the onboard flight control system (OBFCS) of the vehicle. Simulation results indicate the successful control of roll and pitch conditions by OBFCS in both SILS and HILS, ensuring the maintenance of flight conditions in accordance with predicted trajectories despite the presence of simulated disturbances. Notably, the simulation demonstrates the independence or absence of interference between each simultaneous MPC control for roll and pitch adjustments.

Corresponding Author:

Herma Yudhi Irwanto
Research Organization of Aeronautics and Space, National Research and Innovation Agency of Indonesia
Jakarta, Indonesia
Email: herm007@brin.go.id

1. INTRODUCTION

The trajectory development of a 6-degree-of-freedom (6-DOF) precision flying vehicle necessitates the design of an appropriate control system alongside the vehicle itself. A viable method entails employing control and navigation techniques comparable to those used in conventional flying vehicles, minimizing the need for extensive maneuvers [1], or generating flight data that closely mirrors the flight dynamics of guided missiles, which often involve intricate maneuvers [2]. Assuming the completion of the vehicle design, a control system identical to the vehicle's control strategy must be integrated to ensure stability and accurate navigation toward the target.

Engaging in research and development pertaining to high-speed unmanned aerial vehicles (UAVs) or controlled rockets (guided missiles) is inherently associated with the precise management of aerial vehicle movements, which demands meticulous accuracy, incurs substantial costs, and necessitates personnel with diverse expertise across scientific disciplines [3]. This developmental endeavor also entails significant risks, as even minor errors could lead to catastrophic outcomes given the velocity and destructive potential of the missile's warhead. Consequently, modeling high-speed UAVs or guided missiles poses substantial challenges, necessitating the utilization of suitable software capable of supporting all phases of development, ranging from design and simulation to prototyping and flight testing [4], [5]. Nevertheless, there is scant literature detailing the comprehensive system development process leading up to flight tests and the subsequent analysis of flight data.

The primary methodology for designing a control system for a flying vehicle entails simulation via software, followed by hardware implementation to assess its capability in handling diverse sensor input data and processing them using advanced control strategies, often involving intricate mathematical calculations and require quite large memory. This process involves sequential stages such as software-in-the-loop simulation (SILS) followed by hardware-in-the-loop simulation (HILS), aimed at integrating the control system directly into the hardware [6], despite this paper not having attempted applications for high-speed UAVs like rockets. This approach optimizes the development process of the flying vehicle, enhancing efficiency and accelerating the overall process while ensuring comprehensive testing prior to actual flight tests.

MATLAB/Simulink, complemented by the Aerospace Toolbox, offers a robust platform for simulating flying vehicles and their corresponding control systems [7], [8]. Conversely, LabVIEW provides enhanced integration and streamlining of hardware programming, rendering it a dependable option for crafting hardware components of vehicle control systems [9]. The combination of these two softwares for implementing avionics and flight control systems is relatively uncommon. Typically, X-Plane serves as the flight simulator in simulations, whether it is MATLAB/Simulink-X-Plane or LabVIEW-X-Plane setups. However, precise control is crucial for high-speed UAV maneuvers, necessitating alignment with the model of the flying vehicle. X-Plane, being a 'black-box' system, may yield disparate results if the system settings are directly applied during flight tests.

This research concentrates on developing a flight control simulation through the integration of MATLAB/Simulink and LabVIEW. With keen attention to the intricate flight dynamics heavily influenced by the 6-DOF, and with the objective of swiftly generating a reliable prototype with minimal design discrepancies and selecting suitable control strategies for application in the primary controller hardware of the vehicle system. The objective of this research is to integrate the aforementioned two software platforms to establish a real-time simulation. This approach has been previously explored in research pertaining to simulation analysis in control applications [10] and system control simulation systems for sounding rocket [11], [12]. The synergy between these software platforms is primed for laboratory or environmental testing and subsequent flight trials, following the SILS and HILS stages. By harnessing a flight model via MATLAB/Simulink and employing control strategies through LabVIEW, this simulation enhances precision in movements and maneuvers while preemptively addressing potential errors prior to actual flight tests, thus offering the potential to curtail development expenses and streamline scheduling and human resource allocation.

The integration of these two software in high-speed UAV or guided missile development serves as a technological mastery in vehicle model development, facilitating collaborative simulation for design and control strategy integration, and enabling prototype testing in laboratory settings.

2. METHOD

The development of this flying vehicle model progresses through stages to accommodate the SILS and HILS procedures. It commences with the simulation of the flying vehicle model using MATLAB/Simulink, adhering meticulously to its detailed specifications. Concurrently, a real-time monitoring display for the vehicle's attitude and position is crafted using LabVIEW. Subsequently, SILS facilitates the integration of these two software platforms via the user datagram protocol (UDP), a widely employed communication protocol validated for autonomous racing vehicles [13]. This development was carried out to the testing stage through the implementation of HILS, which was specifically designed for 200 mm caliber rockets equipped with canards for mission trajectory correction (TC).

2.1. Modelling using MATLAB/Simulink

The control and stability of a flying vehicle are achieved through the design of the vehicle, which includes the incorporation of fin stabilizers or control surfaces for maneuvering. As per Cook's book Flight Dynamics Principles [14], the deflections of control surfaces ($\delta 1$, $\delta 2$, $\delta 3$, $\delta 4$) of flying vehicle must undergo translation through a conversion matrix to serve as input data for the PLANT, thereby the control surfaces function as control fields called elevator, aileron and rudder (δr, δp, δy). Parameters such as thrust, inertia, and aerodynamic coefficients constitute the dynamic properties of the flying vehicle and are included in this information. The longitudinal and lateral directional mode equations, capable of being represented in two state space forms, form the basis for simulating the flight dynamics of the PLANT model rocket.

$$\begin{bmatrix}\dot{u}\\ \dot{w}\\ \dot{q}\\ \dot{\theta}\end{bmatrix}=\begin{bmatrix}x_u & x_w & x_q & x_\theta\\ z_u & z_w & z_q & z_\theta\\ m_u & m_w & m_q & m_\theta\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}u\\ w\\ q\\ \theta\end{bmatrix}+\begin{bmatrix}x_\eta & x_\tau\\ z_\eta & z_\tau\\ m_\eta & m_\tau\\ 0 & 0\end{bmatrix}\begin{bmatrix}\eta\\ \tau\end{bmatrix} \tag{1}$$

$$\begin{bmatrix}\dot{v}\\ \dot{p}\\ \dot{r}\\ \dot{\phi}\end{bmatrix}=\begin{bmatrix}y_v & y_p & y_r & y_\phi\\ l_v & l_p & l_r & l_\phi\\ n_v & n_p & n_r & n_\phi\\ 0 & 1 & 0 & 0\end{bmatrix}\begin{bmatrix}v\\ p\\ r\\ \phi\end{bmatrix}+\begin{bmatrix}y_\xi & y_\zeta\\ l_\xi & l_\zeta\\ n_\xi & n_\zeta\\ 0 & 0\end{bmatrix}\begin{bmatrix}\xi\\ \zeta\end{bmatrix} \tag{2}$$

where u is velocity at x-axis (m/s); w is velocity at z-axis (m/s); v is velocity at y-axis (m/s); q is pitch rate (deg/s); θ is pitch angle (deg); η is Elevator deflection (deg); τ is thrust (N); p is roll rate (deg/s); r is yaw rate (deg/s); ϕ is roll angle (deg); ξ is aileron deflection (deg); and ζ is rudder deflection (deg).

Utilizing the Aerospace Toolbox from MATLAB/Simulink above, the results of the dynamics simulation of the PLANT flying vehicle model depict real-time positional changes in terms of longitude, latitude, and altitude relative to the earth (Xe, Ye, Ze), as well as rotation (ϕ, θ, φ) based on the Euler formula. Additionally, the simulation illustrates translational velocities (u, v, w) and accelerations (ax, ay, az), along with rotational velocities (p, q, r).

As illustrated in Figure 1, the dynamics simulation of the rocket facilitates the formulation of mathematical models that represent the rocket's behavior. These models depict the rocket's movement in three-dimensional space, delineated through the center of mass position and Euler angles governing the rocket's rotation.

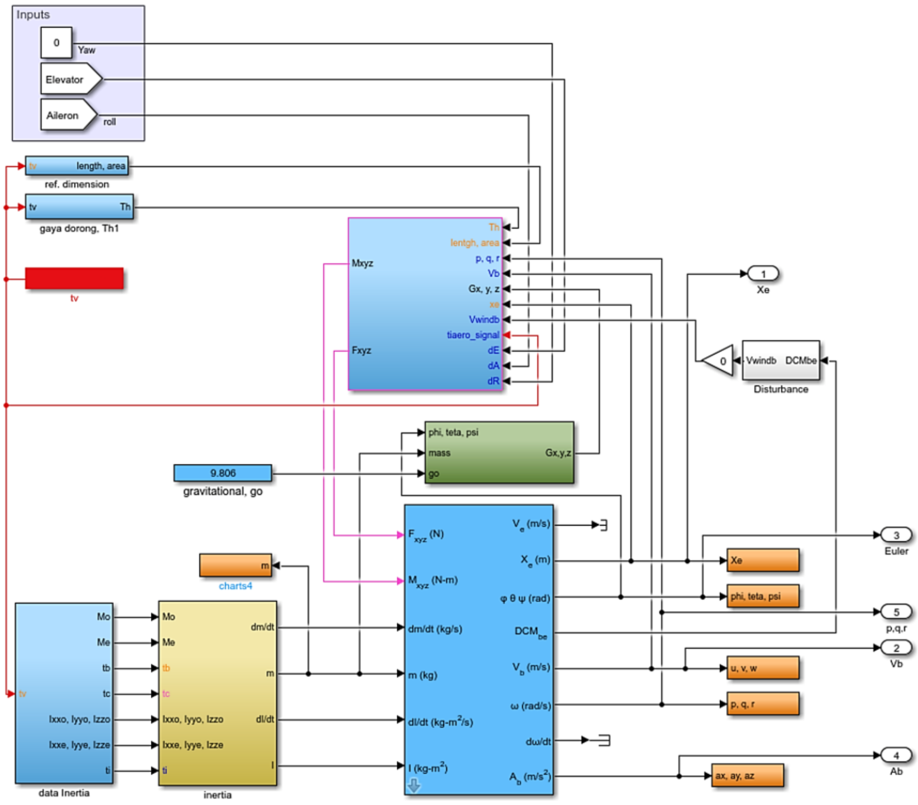
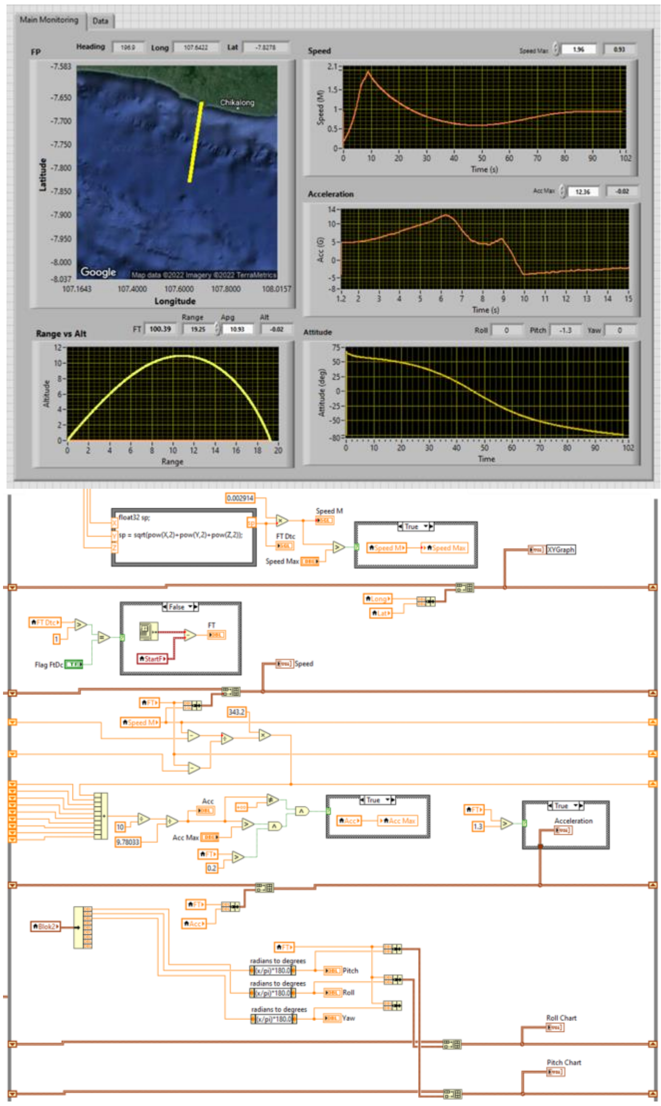


Figure 1. Flight vehicle modelling on MATLAB/Simulink

This simulation is structured into three primary segments: pre-simulation, the simulation process, and post-simulation. During the pre-simulation phase, input data entails details regarding the rocket's geometry, encompassing parameters such as thickness and other components influencing aerodynamics during flight. Additionally, the system necessitates input regarding alterations in rocket mass and propulsion characteristics, which are derived from independent static tests or the thrust profile. Subsequently, once the requisite data is inputted, the simulation process commences and persists until the calculated flight path of the rocket reaches a height of 0 meters from the local horizon surface. The simulation process entails consideration of 6-DOF in rocket flight dynamics, aerodynamic calculations, and static stability analysis. Post-simulation culminates in the generation of output data, including the rocket's flight path, fall point location, maximum acceleration and speed, maximum altitude attained, and total flight duration from launch to fall point.

2.2. Real time monitoring using LabVIEW

Continuous monitoring of the vehicle's movements and flight positions is imperative throughout its operation. Hence, tools are indispensable for real-time monitoring during flight. A user-friendly monitoring system has been devised utilizing LabVIEW. This system provides a graphical representation of the vehicle's flight location and coordinates overlaid on a Google Map interface. Additionally, it includes charts displaying altitude, apogee, and downrange positions, along with flight attitude parameters such as rocket motor thrust, velocity, acceleration, roll, pitch, and yaw, as shown in Figure 2.



Heading 196.9 Long 107.6422 Lat -7.8278

Speed Max 1.96 0.93

Acc Max 12.36 -0.02

FT 100.39 Range 19.25 Apg 10.93 Alt -0.02

Roll 0 Pitch -1.3 Yaw 0

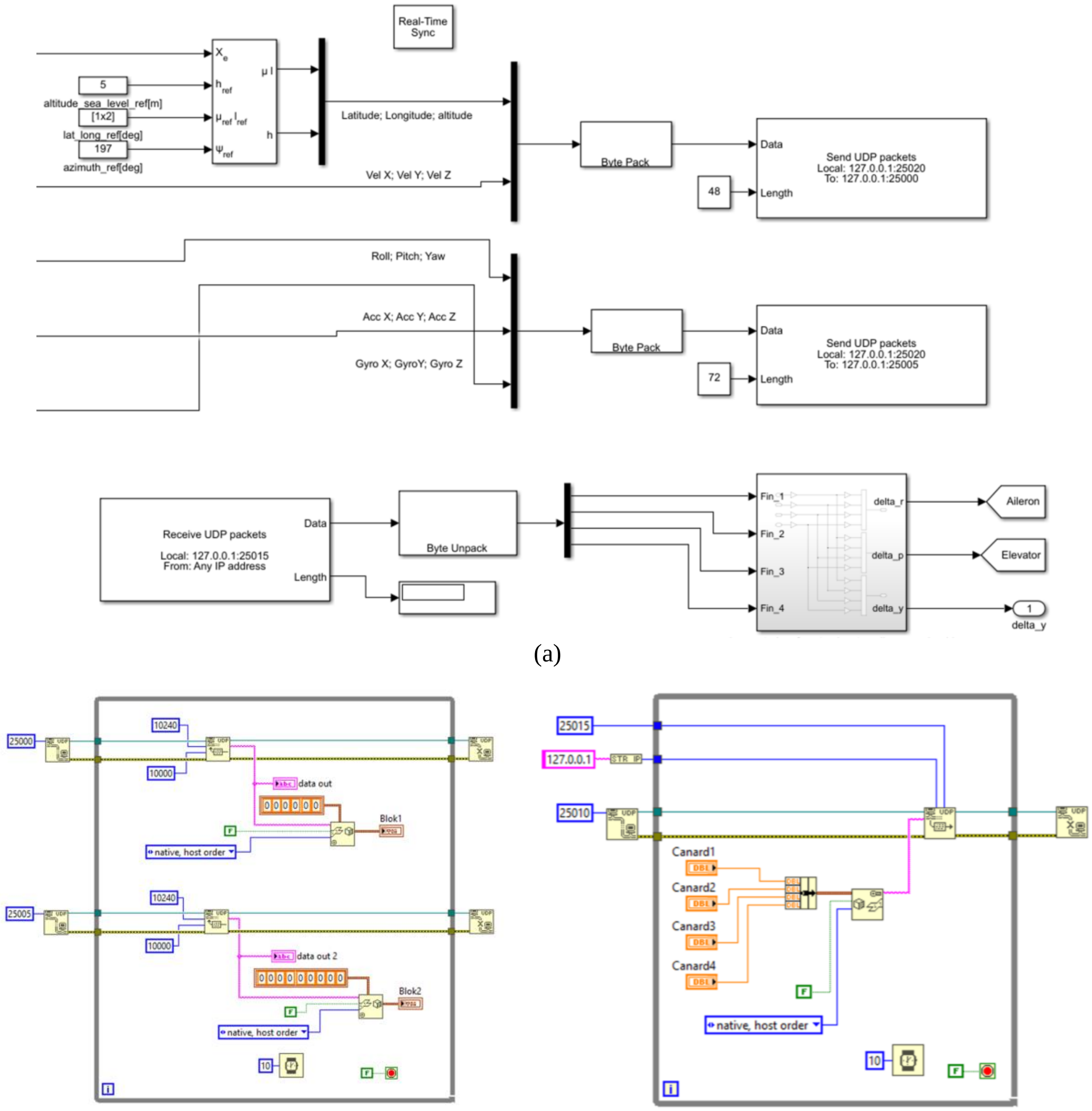
Figure 2. Programming and real time monitoring on LabVIEW

This can be easily achieved within LabVIEW by directly receiving continuous raw data from MATLAB/Simulink through UDP and presenting it directly at a specified frequency. Each chart's parameters are tailored or converted into data units for each axis, as depicted in Figure 2. Additionally, this data can be stored in excel file format for subsequent analysis.

2.3. UDP protocol connects MATLAB/Simulink and LabVIEW

Bidirectional communication between these two software platforms takes place via the UDP protocol. MATLAB/Simulink serves as the source of flight simulation data, transmitting various parameters such as latitude, longitude, altitude (Xe, Ye, Ze), speed (u, v, w), roll, pitch, yaw (ϕ, θ, φ), acceleration (ax, ay, az), and gyroscopic data (p, q, r). Conversely, LabVIEW transmits data concerning aileron, elevator, and rudder (δr, δp, δy) inputs representing motion commands, which are then converted into control signals for the vehicle's four-fin canards, enabling precise vehicle control.

The UDP protocol facilitates real-time simulation, whether internally within a single computer or across multiple computers connected via a hub/switch, utilizing designated port and IP addresses. This configuration enables both software platforms to establish real-time closed-loop communication, as in Figure 3(a) is communication programming on MATLAB/Simulink and Figure 3(b) on LabVIEW.



(a)

(b)

Figure 3. Data communication using UDP on; (a) MATLAB/Simulink and (b) LabVIEW

2.4. Strategy control uses model predictive control for trajectory corrections missions

In this study, a TC mission was executed for a launched rocket, with the objective of ensuring its flight trajectory aligns with previously calculated predictions. model predictive control (MPC) was employed as the designated control strategy [15], [16], drawing from successful high-speed UAV control simulations in developing the onboard attitude determination control system (OADCS) for roll stabilization, pitch, and yaw control [4] and real-flight tests of low-speed UAVs [17]. Its ease of configuration and rapid response to emerging disturbances enable immediate mitigation, restoring conditions to the desired trajectory and altitude settings. Consequently, in this rocket TC mission, continuous control of all motion parameters, especially pitch and yaw, is vital to maintain the rocket's alignment with the predicted conditions, as depicted in Figure 4. Emphasis is placed on sustaining roll control, as dictated by the longitudinal and lateral directional models of the vehicle under development.

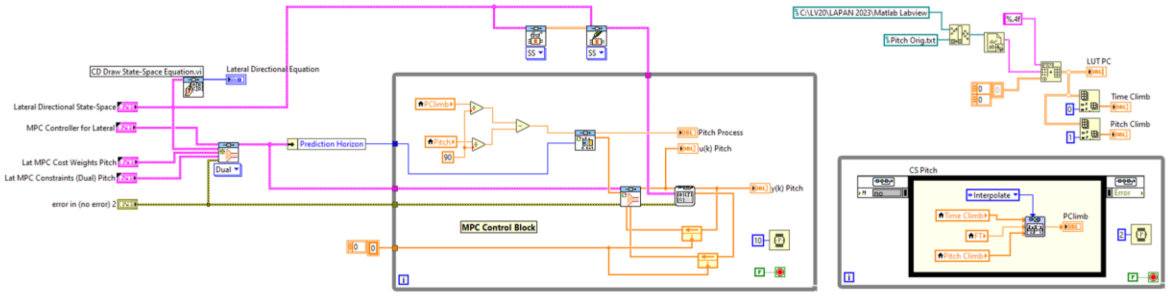


Figure 4. MPC for pitch control using LabVIEW

3. RESULTS AND DISCUSSION

The aim of developing the avionics and flight control system for the TC mission is to maintain the calculated flight trajectory. This developmental process will advance through stages including SILS, HILS, ready to flight system (RFTS) simulations, ground testing, and final flight trials.

3.1. Software-in-the-loop simulation

The incorporation of MATLAB/Simulink in collaborative research with LabVIEW aims to replace the X-Plane flight simulator [18], throughout the entire simulation phase, which encompasses attitude determination, processing of position data, and trajectory control following using SILS. The SILS method can involve either two computers with distinct ethernet addresses or a single computer with the same ethernet address, albeit with different ports, with MATLAB/Simulink and LabVIEW installed on each computer. Unlike X-Plane's 'black box' nature, which internally provides data inaccessible to researchers, MATLAB/Simulink ensures coherence between the designed vehicle model and the mathematical model utilized as input for the control strategy implemented in LabVIEW [19]. This alignment is expected to yield more precise control outcomes for trajectory adjustments during the mission.

In this research, a flying vehicle was conceptualized as a 200 mm caliber rocket measuring 4 meters in length, featuring four fixed rear fins for stabilization and four front fins as canards, as depicted in the Figure 5. With a maximum velocity approaching 2 Mach numbers and a burn time of 10 seconds, the rocket's specifications informed the derivation of mathematical models. Notably, based on specifications (1) and (2), one such model was established for the 12th-second condition of burning time, characterized by a speed of 1.81 Mach and an altitude of 3953 m. To fulfill control requirements, an additional mathematical model is indispensable, as outlined below:

- Longitudinal model:

$$\begin{bmatrix}\dot{u}\\ \dot{w}\\ \dot{q}\\ \dot{\theta}\end{bmatrix}=\begin{bmatrix}-0.083 & 0 & 0 & 5.555\\ 0 & -1.08 & 588.5 & 8.08\\ 0 & -0.638 & -1.584 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}u\\ w\\ q\\ \theta\end{bmatrix}+\begin{bmatrix}0 & 0.005\\ -0.178 & 0\\ 1.635 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\delta_e\\ T\end{bmatrix} \tag{3}$$

$$\begin{bmatrix}u\\ w\\ q\\ \theta\end{bmatrix}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0.097 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}u\\ w\\ q\\ \theta\end{bmatrix}+\begin{bmatrix}0 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\delta_e\\ T\end{bmatrix} \tag{4}$$

- Lateral directional model:

$$\begin{bmatrix}\dot{v}\\ \dot{p}\\ \dot{r}\\ \dot{\phi}\end{bmatrix}=\begin{bmatrix}-1.622 & 0 & -587 & 5.555\\ 0 & -7.646 & 0 & 0\\ 0.957 & 0 & -4.7 & 0\\ 0 & 1 & 1.455 & 0\end{bmatrix}\begin{bmatrix}v\\ p\\ r\\ \phi\end{bmatrix}+\begin{bmatrix}0 & 0.178\\ -16.287 & 0\\ 0 & 1.635\\ 0 & 0\end{bmatrix}\begin{bmatrix}\delta_a\\ \delta_r\end{bmatrix} \tag{5}$$

$$\begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \tag{6}$$

Figure 5. Flying vehicle model (rocket)

Specifically concerning the development of a 200 mm caliber rocket, a detailed mathematical model will be presented in a separate publication. The longitudinal (3), (4) and lateral directional (5), (6) models outlined above serve as inputs for implementing the control strategy, employing MPC [20], [21], with the aim of ensuring roll stability and adjusting pitch and yaw to align with flight predictions (TC). The interconnected relationship between MATLAB/Simulink and LabVIEW will facilitate the TC of the rocket based on flight prediction data integrated into the onboard flight control system (OBFCS). This process involves referencing a lookup table according to the rocket's time, pitch, and yaw conditions, with MPC ensuring continuous real-time adjustment to maintain roll stability and promptly address pitch and yaw deviations. Figure 6(a) depicts the integration of the rocket's mathematical model in LabVIEW, incorporating MPC as the control strategy, along with the outcomes of adjustments made to the presented disturbances (Figure 6(b)) (in Appendix).

In this study, two disturbance scenarios were simulated: a roll disturbance from 3 to 7 seconds, followed by MPC control to maintain roll stability after the 7th second. Additionally, a pitch-up disturbance occurred from 7 to 15 seconds, with MPC adjusting the pitch to align with predicted flight outcomes. It is evident that each MPC control action between roll and pitch operates independently, even amidst the disturbance occurring at the 7th second and beyond.

In Figure 7, a detailed examination of roll control reveals that despite efforts to maintain stability, it can take up to 20 seconds to regain control, as indicated by the gradual reduction of disturbance from 5.5 degrees to 0 degrees between the 7th and 27th seconds. However, it is evident that this roll control response is insufficiently aggressive, necessitating further adjustments to the MPC horizon and control prediction settings. Additionally, refinement of the lateral directional model is required to facilitate quicker responses in maintaining roll stability.

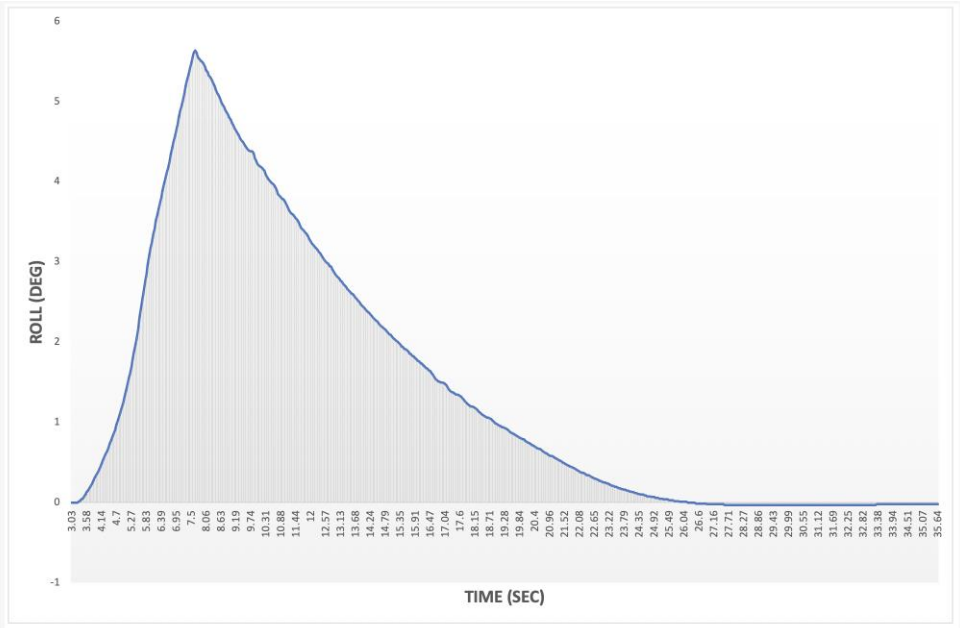


Figure 7. Disturbance with roll

Regarding pitching up disturbances affecting the canard at the X-Fin position, with degrees ranging from 1 to 1.3, Figure 8 illustrates these disturbances occurring predominantly between the 7th and 15th seconds. Notably, the rocket's velocity begins to decrease below Mach 1 after the 25th second, following the burnout of the rocket motor. Consequently, the effectiveness of the canards in influencing the rocket's pitch diminishes. However, post-apogee, as the rocket's velocity increases again, the canards resume their function. Upon touchdown, the MPC control strategy successfully adjusts the rocket's pitch positions in accordance with flight predictions.

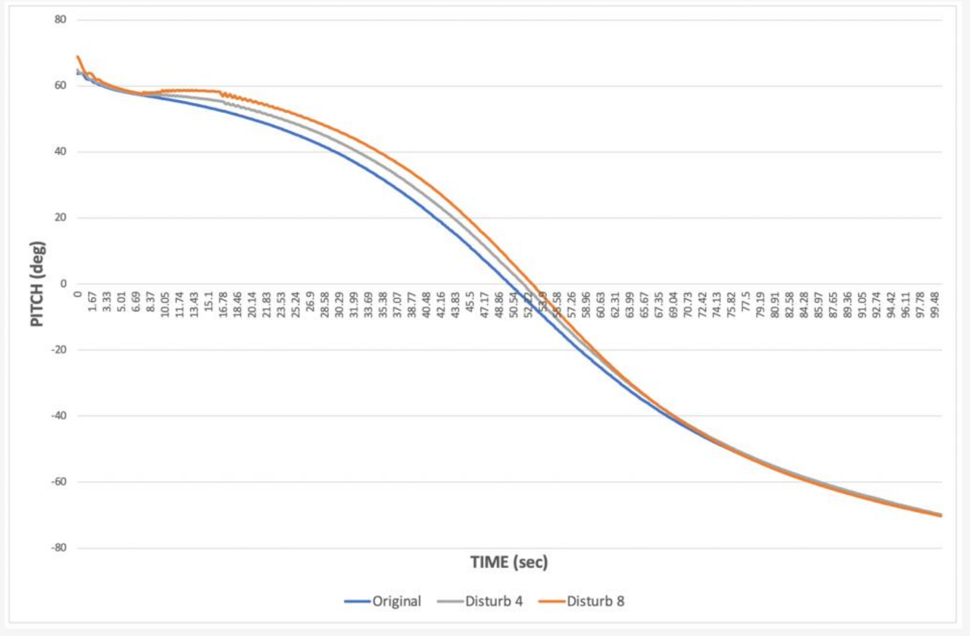


Figure 8. Disturbance with pitch

Figure 9 provides a magnified view of Figure 8, showcasing a comparison between three pitching up disturbances with 0.4, 0.8, and 1.2 degrees, in contrast to the original trajectory (undisturbed). Specifically, a 1.2 degree pitching up disturbance on the canards resulted in a deviation of the apogee from its normal position, shifting it from 10.93 km to 11.63 km (an increase of 0.7 km).

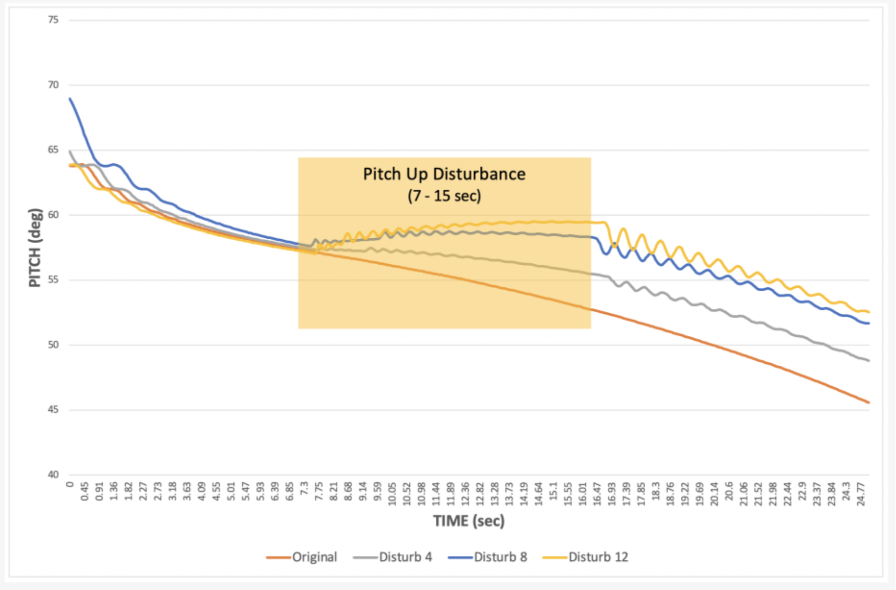


Figure 9. Zooming interference with pitch

The SILS phase has effectively simulated roll and pitch control in real-time for a TC mission, aligning with predictive calculations, by integrating two software platforms, MATLAB/Simulink and LabVIEW with each leveraging its unique strengths. MPC effectively addresses disturbances in roll and pitch throughout the TC mission, ensuring the rocket's trajectory is maintained until completion.

In the TC mission, real-world scenarios are not expected to entail extreme roll or pitch disturbances akin to those simulated above. This expectation arises from the continuous maintenance and approximation of roll and pitch conditions by the MPC, aligning them with the readings from the roll and pitch lookup table corresponding to predicted flight outcomes.

3.2. Hardware-in-the-loop simulation

The subsequent phase involves transferring the entire trajectory calculation process, previously executed on a desktop computer, to a microcontroller intended for use in the actual rocket. This phase is referred to as HILS.

This simulation phase involves the hardware implementation, specifically the integration of the vehicle/rocket avionics and flight control system within the OBFCS. For this purpose, the research employs the OBFCS featuring the National Instruments myRIO-1950 main processor [16], [22], [23], which is based on the Xilinx Zynq-7010 FPGA processor clocked at 667 MHz. It is equipped with 512 MB non-volatile memory, 256 MB DDR3 memory, and ARM Cortex A9 processor, alongside 32 ch Digital Input/Output (DIO), 8 ch analog input (AI), 4 ch analog output (AO), 2 ch TTL serial, and an 8G 3-axis accelerometer.

The movement of the four fin canards is controlled by commands from the myRIO via the actuator control system (ACS). Furthermore, the myRIO is linked to the controller computer, running LabVIEW, via shared memory. The selection of the NI myRIO-1950 is tailored to the LabVIEW software utilized in the collaborative research between MATLAB/Simulink and LabVIEW, obviating the need for an additional converter to facilitate the myRIO's role as the main controller in subsequent flight tests. The interconnectedness of computers and subsystems within the HILS series is illustrated in Figure 10.

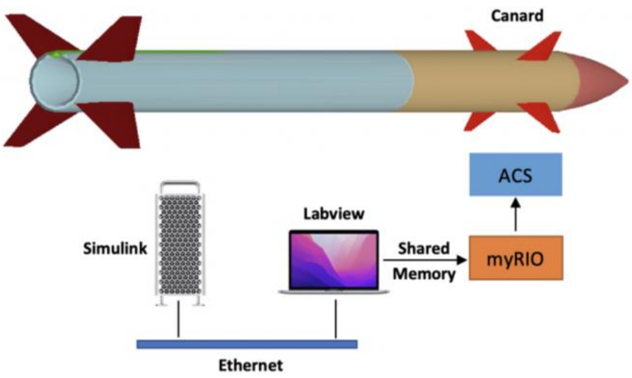


Figure 10. HILS environment

The research testing conducted using the HILS method [24]–[26] has proven successful, with the entire system operating seamlessly in real time. The close-loop interaction between the MATLAB/Simulink and LabVIEW computers directs the movement of the canards in accordance with flight predictions. Due to the rocket's nearly 2 Mach Number speed in simulation, only a subtle 1-degree movement of the canards is required, barely perceptible to the naked eye. Hence, an exceedingly precise canard control mechanism with ample torque strength to withstand air drag during flight is essential.

The canard control strategy, depicted in a mode resembling Figure 11, governs all rocket movements. The ACS issues independent signals to each canard based on roll, pitch, and yaw commands from the OBFCS in degrees. Simultaneously during HILS, OBFCS transmits canard data to MATLAB/Simulink, previously converted into aileron, elevator, and rudder inputs, as illustrated in Figures 12(a) and (b).

As a result, all subsystems within the HILS operate effectively, demonstrating the comprehensive simulation of the rocket flight test process from launch to touchdown, directly orchestrated by OBFCS, ACS, and the rocket canards in real-time. However, all of these simulations and tests must be validated through dynamic trials, namely actual flight tests. The testing process is conducted in stages, typically involving at least three events. The initial rocket flight test involves using a fixed canard, also known as without an actuator, to assess the rocket's flight stability. The second event entails a 'zero-degree canard' mission, aiming to evaluate the canard's capability to withstand rocket forces during flight. Finally, the third event encompasses the actual TC mission.

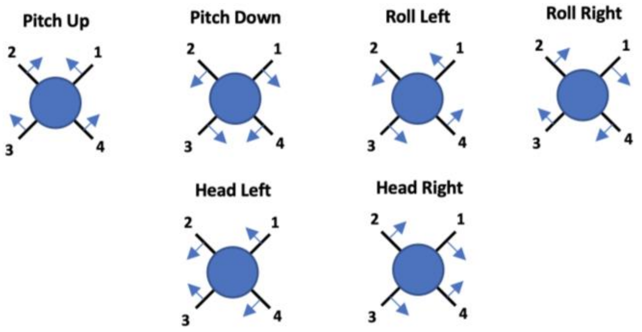
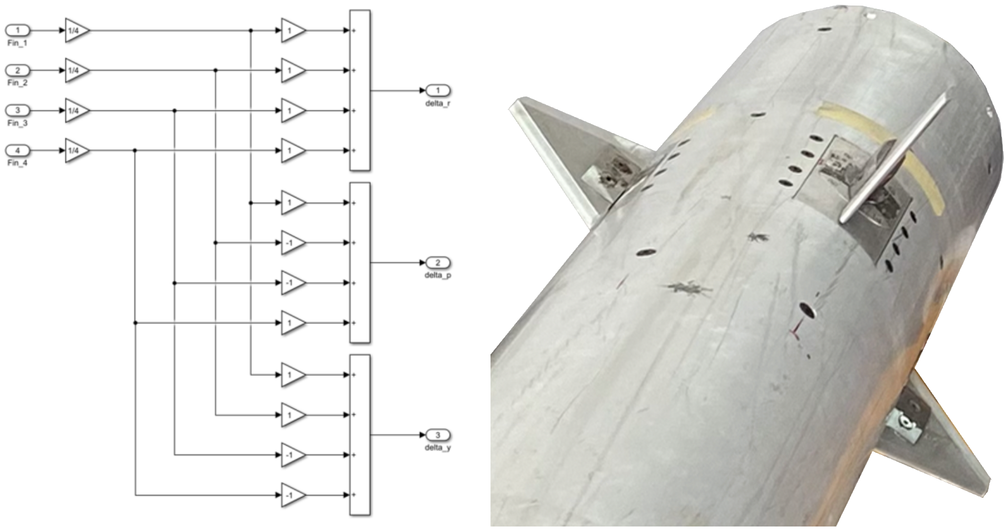


Figure 11. Canard control mode



(a) (b)

Figure 12. Canard; (a) movement conversion and (b) implementation and testing

3.3. Discussion

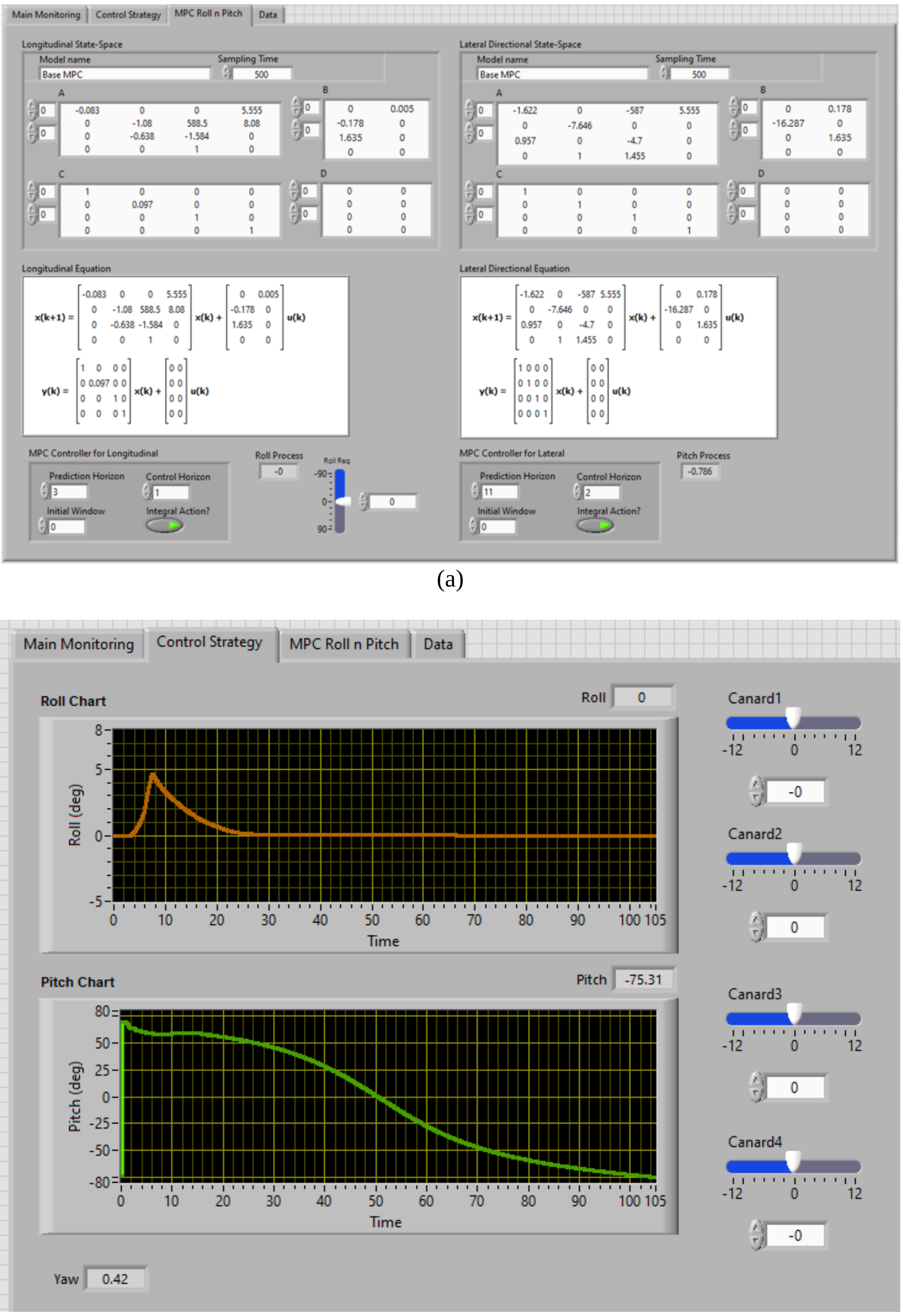
This research exemplifies that the development of intricate and costly flying vehicles can be conducted with ease and scientific rigor through the integration of software and hardware. The systematic progression through SILS and HILS stages furnishes a targeted research framework and facilitates direct applications employing real hardware. The utilization of a precise calculation model leveraging MATLAB/Simulink in conjunction with National Instruments hardware equipped with LabVIEW as the programming language not only enhances research accuracy but also expedites implementation toward tangible outcomes.

The findings of this research are applicable to various types of high-speed flying vehicles, including rockets and UAVs with turbojet engines. While existing research predominantly focuses on similar systems for low-speed (under 200 km/h) or multi-rotor UAVs, the methodologies presented herein pave the way for easier integration into high-speed controlled flying vehicles in the future. The viability of MPC as a control strategy holds promise for further enhancement or integration with advanced technologies such as reinforcement learning in subsequent research endeavors.

4. CONCLUSION

The development of vehicle avionics and flight control systems involves mathematical modeling and direct implementation into the vehicle hardware system. This process is facilitated by incremental simulation using SILS and HILS methods, ensuring the success of system development before actual flight testing commences. By employing this method, potential issues that may arise during flight tests are minimized. Moreover, the collaborative framework between MATLAB/Simulink and LabVIEW streamlines the direct implementation into the OBFCS of flying vehicles, particularly in implementing the MPC control strategy for the TC mission in this study. Further testing endeavors can then be concentrated on enhancing the precision of canard movements, which directly impacts rocket maneuvering, particularly crucial for guided missiles requiring accuracy in achieving targets.

APPENDIX



Longitudinal State-Space

Model name: Base MPC

Sampling Time: 500

$$A=\begin{bmatrix}-0.083 & 0 & 0 & 5.555\\ 0 & -1.08 & 588.5 & 8.08\\ 0 & -0.638 & -1.584 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\quad B=\begin{bmatrix}0 & 0.005\\ -0.178 & 0\\ 1.635 & 0\\ 0 & 0\end{bmatrix}$$

$$C=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0.097 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\quad D=\begin{bmatrix}0 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0\end{bmatrix}$$

Longitudinal Equation

$$x(k+1)=\begin{bmatrix}-0.083 & 0 & 0 & 5.555\\ 0 & -1.08 & 588.5 & 8.08\\ 0 & -0.638 & -1.584 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}x(k)+\begin{bmatrix}0 & 0.005\\ -0.178 & 0\\ 1.635 & 0\\ 0 & 0\end{bmatrix}u(k)$$

$$y(k)=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0.097 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}x(k)+\begin{bmatrix}0 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0\end{bmatrix}u(k)$$

MPC Controller for Longitudinal

Prediction Horizon: 3

Control Horizon: 1

Initial Window: 0

Integral Action?

Roll Process: -0

Roll Req: 0

Lateral Directional State-Space

Model name: Base MPC

Sampling Time: 500

$$A=\begin{bmatrix}-1.622 & 0 & -587 & 5.555\\ 0 & -7.646 & 0 & 0\\ 0.957 & 0 & -4.7 & 0\\ 0 & 1 & 1.455 & 0\end{bmatrix}\quad B=\begin{bmatrix}0 & 0.178\\ -16.287 & 0\\ 0 & 1.635\\ 0 & 0\end{bmatrix}$$

$$C=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\quad D=\begin{bmatrix}0 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0\end{bmatrix}$$

Lateral Directional Equation

$$x(k+1)=\begin{bmatrix}-1.622 & 0 & -587 & 5.555\\ 0 & -7.646 & 0 & 0\\ 0.957 & 0 & -4.7 & 0\\ 0 & 1 & 1.455 & 0\end{bmatrix}x(k)+\begin{bmatrix}0 & 0.178\\ -16.287 & 0\\ 0 & 1.635\\ 0 & 0\end{bmatrix}u(k)$$

$$y(k)=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}x(k)+\begin{bmatrix}0 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0\end{bmatrix}u(k)$$

MPC Controller for Lateral

Prediction Horizon: 11

Control Horizon: 2

Initial Window: 0

Integral Action?

Pitch Process: -0.786

(a)

Roll: 0

Pitch: -75.31

Canard1: -0

Canard2: 0

Canard3: 0

Canard4: -0

Yaw: 0.42

(b)

Figure 6. Flight vehicle: (a) models serve as inputs for MPC and (b) provide results disturbances

ACKNOWLEDGEMENTS

This research was made possible through funding from the Government of the Republic of Indonesia under the Indonesian Advanced Research and Innovation (RIIM - LPDP) Program Batch 1 Year 2022. Specifically, the funding was allocated for the "Two Stage Sonda Rocket Technology Dissemination for the Development of the RHAN 450 Defence Rocket with a 100 KM Flying Range" (Grant No. B-808/II.7.5/FR/6/2022 and B-9546/III.1/KS.00.00/6/2022). The author extends gratitude to colleagues in the Rocket Avionics System Program at the Research Center for Rocket Technology, ORPA, BRIN, and Cohort4 of the Doctoral Program at the Defense University of the Republic of Indonesia for their invaluable assistance, both direct and indirect.

REFERENCES

[1] R. A. Duhri and I. E. Putro, "Guidance and control of 2nd stage RKX-200EB missile using proportional navigation approach," in *AIP Conference Proceedings*, Sep. 2021, p. 030012, doi: 10.1063/5.0060045.

[2] K.-W. Jung, Y.-W. Kim, and C.-H. Lee, "Aerodynamically controlled missile flight datasets and its applications," *International Journal of Aeronautical and Space Sciences*, vol. 24, no. 1, pp. 248–260, Feb. 2023, doi: 10.1007/s42405-022-00531-x.

[3] V. Raja, A. Prakash R, A. Kumar, and D. A. de J. Pacheco, "Multi-disciplinary engineering design of a high-speed nature-inspired unmanned aquatic vehicle," *Ocean Engineering*, vol. 270, p. 113455, Feb. 2023, doi: 10.1016/j.oceaneng.2022.113455.

[4] H. Y. Irwanto, "Development of autonomous controller system of high speed UAV from simulation to ready to fly condition," *Journal of Physics: Conference Series*, vol. 962, p. 012015, Feb. 2018, doi: 10.1088/1742-6596/962/1/012015.

[5] Y. C. Paw and G. J. Balas, "Development and application of an integrated framework for small UAV flight control development," *Mechatronics*, vol. 21, no. 5, pp. 789–802, Aug. 2011, doi: 10.1016/j.mechatronics.2010.09.009.

[6] H. Y. Irwanto and E. Artono, "Correlation of hardware in the loop simulation (HILS) and real control vehicle flight test for reducing flight failures," *Journal of Physics: Conference Series*, vol. 1130, p. 012014, Nov. 2018, doi: 10.1088/1742-6596/1130/1/012014.

[7] P. Lu and Q. Geng, "Real-time simulation system for UAV based on Matlab/Simulink," in *2011 IEEE 2nd International Conference on Computing, Control and Industrial Engineering*, Wuhan, China: IEEE, Aug. 2011, pp. 399–404, doi: 10.1109/CCIENG.2011.6008043.

[8] L. R. Ribeiro and N. M. F. Oliveira, "UAV autopilot controllers test platform using Matlab/Simulink and X-Plane," in *2010 IEEE Frontiers in Education Conference (FIE)*, Arlington, VA, USA: IEEE, Oct. 2010, pp. S2H-1-S2H-6, doi: 10.1109/FIE.2010.5673378.

[9] Q. Wang and G. Yang, "Design of vehicle controller test program based on LabVIEW," in *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*, Xiamen, China: IEEE, Oct. 2019, pp. 940–943, doi: 10.1109/EITCE47263.2019.9094856.

[10] C. A. Cansalar, E. Mavis, and C. Kasnakoglu, "Simulation time analysis of MATLAB/Simulink and LabVIEW for control applications," in *2015 IEEE International Conference on Industrial Technology (ICIT)*, Seville, Spain: IEEE, Mar. 2015, pp. 470–47, doi: 10.1109/ICIT.2015.7125143.

[11] A. Nagani, M. Nasiruddin, and T. Goskula, "Simulation of control system with LabVIEW," *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)*, pp. 30–35, 2014.

[12] F. Trevisi *et al.*, "Simulation of a sounding rocket flight's dynamic," in *2017 IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, Padua, Italy: IEEE, Jun. 2017, pp. 296–300, doi: 10.1109/MetroAeroSpace.2017.7999584.

[13] E. Tramacere, S. Luciani, S. Feraco, A. Bonfitto, and N. Amati, "Processor-in-the-loop architecture design and experimental validation for an autonomous racing vehicle," *Applied Sciences*, vol. 11, no. 16, p. 7225, Aug. 2021, doi: 10.3390/app11167225.

[14] Cook Michael V., 2012. Book: Flight Dynamics Principles: A Linear Systems Approach to Aircraft Stability and Control (Aerospace Engineering), Edition 3. Publisher: Butterworth-Heinemann, ISBN: 978-0080982427

[15] I. McInerney, L. Nita, Y. Nie, A. Oliveri, and E. C. Kerrigan, "Towards a framework for nonlinear predictive control using derivative-free optimization," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 284–289, 2021, doi: 10.1016/j.ifacol.2021.08.558.

[16] N. Mehrabi, R. S. Razavian, B. Ghannadi, and J. McPhee, "Predictive simulation of reaching moving targets using nonlinear model predictive control," *Frontiers in Computational Neuroscience*, vol. 10, Jan. 2017, doi: 10.3389/fncom.2016.00143.

[17] H. Y. Irwanto, I. E. Putro, and S. Saeri, "HILS of FPV-2600 UAV using MyRIO-1950 as optimal flight control system," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 11, no. 5, pp. 1780–1786, Oct. 2021, doi: 10.18517/ijaseit.11.5.11919.

[18] S. Moon, E.-M. Oh, D.-I. You, and D. H. Shim, "Implementation of a X-plane and MATLAB/Simulink based simulation system for multiple UAVs," *Journal of Institute of Control, Robotics and Systems*, vol. 19, no. 5, pp. 442–449, May 2013, doi: 10.5302/J.ICROS.2013.13.9009.

[19] F. M. Cantri, M. H. Bisri, and H. Y. Irwanto, "Realtime simulation for rocket using visual programming," in *2022 IEEE 8th Information Technology International Seminar (ITIS)*, Surabaya, Indonesia: IEEE, Oct. 2022, pp. 150–155, doi: 10.1109/ITIS57155.2022.10010182.

[20] F. Almeida, "Model predictive control and constrained dynamic compensation for a hypersonic flight vehicle," in *AIAA Scitech 2021 Forum*, Reston, VA, USA: American Institute of Aeronautics and Astronautics, Jan. 2021, doi: 10.2514/6.2021-1566.

[21] S. Pérez-Roca *et al.*, "An MPC approach to transient control of liquid-propellant rocket engines," *IFAC-PapersOnLine*, vol. 52, no. 12, pp. 268–273, 2019, doi: 10.1016/j.ifacol.2019.11.254.

[22] A. A. Alahmadi, "Developing a wireless real-time automated home approach utilizing NI MyRIO microcontroller board and LabVIEW platform," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 3, pp. 1273–1278, Dec. 2019, doi: 10.11591/ijeecs.v16.i3.pp1273-1278.

[23] S. Bouallègue and R. Fessi, "Rapid control prototyping and PIL co-simulation of a quadrotor UAV based on NI myRIO-1900 board," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 6, pp. 26–35, 2016, doi: 10.14569/IJACSA.2016.070604.

[24] S. Braun, P. Wolze, B. Braun, M. Geiser, and F. Holzapfel, "Hardware-in-the-loop simulation in the context of the development of automatic flight control systems for UAV," in *Airtech UAV World*, Frankfurt, 2012.

[25] Y. W. Hadi and R. T. Bambang, "Development of hardware-in-the-loop simultion for rocket guidance system," in *2015 International Conference on Electrical Engineering and Informatics (ICEEI)*, Denpasar, Indonesia: IEEE, Aug. 2015, pp. 229–234, doi: 10.1109/ICEEI.2015.7352502.

[26] W. Zhao, Y. Guo, J. Yang, and H. Sun, "Hardware-in-the-loop simulation platform for fault diagnosis of rocket engines," in *2019 IEEE 10th International Conference on Mechanical and Aerospace Engineering (ICMAE)*, Brussels, Belgium: IEEE, Jul. 2019, pp. 268–273, doi: 10.1109/ICMAE.2019.8880973.

BIOGRAPHIES OF AUTHORS

Herma Yudhi Irwanto was born in Banyuwangi, Indonesia, December 30, 1967. He holds a Bachelor of Engineering (B.Eng) from the Shibaura Institute of Technology, Japan in 1992. Then a Master's degree (M.Eng) from same university, Shibaura Institute of Technology, Japan in 1994. Currently taking a doctoral course at the Defense University of Republic of Indonesia. Working as engineer in Research Center for Rocket Technology, Research Organization for Aeronautics and Space, National Research and Innovation Agency of Indonesia. In addition, he was also the Head of Guided, Control and Telemetry (2011-2015) at the Research Center for Rocket Technology. And now he is active as a researcher and as a Group Leader of Rocket Avionics and Flight Control System for rocket technology. He can be contacted at email: herm007@brin.go.id.

Purnomo Yusgiantoro was born on June 16, 1951 in Semarang, Central Java, Indonesia. He received the B.Eng. degrees from Bandung Institute of Technology, Indonesia in 1974, M.Sc., and Ph.D. degree from Colorado School of Mines, Golden, CO, USA in 1986 and 1988. Currently working as Senior Adviser to the Chief of Presidential Staff of Indonesia, has served as the Minister of Defense of the Republic of Indonesia (2009-2014), and also as Professor in Bandung Institute of Technology and Defense University of Republic of Indonesia. He can be contacted at email: pyc@pycenter.org.

Zainal Abidin Sahabuddin was born on April 24, 1957 in Rappang, South Sulawesi, Indonesia. He received the Bachelor of Engineering from Hasanuddin University, Master in Management from UPN Veteran, and Doctor in Management from Padjajaran University, Indonesia. Working as permanent lecturer and as Professor in Defense University of Republic Indonesia since May 2022. He can be contacted at email: zainal.sahabuddin@idu.ac.id.

Romie Oktovianus Bura was born on Surabaya, East Java, Indonesia. He received B.Eng (Hons) from UMIST and The Victoria University of Manchester (1999), and PhD from University of Southampton (2004). He is working as Assistant Professor of Aerospace Engineering in Bandung Institute of Technology and Permanent Lecturer in Defense Univerity of Republic of Indonesia. Currently engaged in research on high-speed aerodynamics and propulsion design. He can be contacted at email: romiebura@idu.ac.id.

Rika Andiarti was born in Sukabumi, West Java, Indonesia, Januari 30, 1967. She received Bachelor's degree from Universite Paul Sabatier, Toulouse, France in 1991, Master's and Doctor degrees from Ecole Centrale Nantes, France, in 1992 and 1995. After finishing her study, she worked as a researcher in Rocket Guidance and Control Research Group at Indonesian Space Agency. She served as Director of several positions at the Indonesian Space Agency (2010-2021) and come back as a researcher at Research Center for Rocket Technology, Research Organization for Aeronautics and Space, National Research and Innovation Agency of Indonesia. She is now appointed as Chief Engineer of the rocket technology development program. She can be contacted at email: rika001@brin.go.id.

Idris Eko Putro was born in Jember, East Java, Indonesia, 1978. He received Master of Science in Aerospace Engineering (Msc.AE.) from Konkuk University, South Korea in 2010. He hold Bachelor degree (Bsc.) from Aeronautics Engineering, ITB, Indonesia in 2002. His research interests include control design, simulation and application. He currently serves as research scientist at National Research and Innovation Agency (BRIN) Indonesia with main responsibility in aerospace vehicle design and modeling, flight dynamics and control simulation. He can be contacted at email: idris.eko.putro@brin.go.id.

Oka Sudiana was born in Jakarta, Indonesia, on July 25, 1983. He received Bachelor’s degree of Aeronautics and Astronautics Engineering from Institut Teknologi Bandung, Indonesia in 2006. Master’s degree of Aerospace Engineering from Universidad Politecnica de Madrid, Spain in 2014 and Special Master’s degree of Aerospace Engineering from Sapienza Universita di Roma, Italy in 2019. He had research visit at Institute of Flight Dynamics, TUM Germany in 2012. He worked as a Researcher in Rocket Guidance and Control Research Group at Indonesian Space Agency (2010-2021). Present, he works at Research Center for Rocket Technology, Research Organization for Aeronautics and Space, National Research and Innovation Agency of Indonesia. He can be contacted email: okas001@brin.go.id.

Azizul Hanif was born in Bekasi, West Java, Indonesia, September 29, 1996. He received bachelor degree from Institut Teknologi Bandung, Indonesia in 2018. Currently, he is working as a researcher at Research Organization for Aeronautics and Space, National Research and Innovation Agency, Indonesia. He can be contacted at email: aziz002@brin.go.id.