Automation, Control and Intelligent Systems
2018; 6(6): 62-72
http://www.sciencepublishinggroup.com/j/acis
doi: 10.11648/j.acis.20180606.11
ISSN: 2328-5583 (Print); ISSN: 2328-5591 (Online)

Flight Vehicle Autopilot System: From Design to Implementation

Bahaaeldin Gamal Abdelaty*, Ashraf Hamdy, Ahmed Nasr Ouda

Technical Research Center, Cairo, Egypt

Email address:

bahaa.blackeagel@yahoo.com (B. G. Abdelaty), ashrafhamdy1991@gmail.com (A. Hamdy), ahnasroda@yahoo.com (A. N. Ouda)

*Corresponding author

To cite this article:

Bahaaeldin Gamal Abdelaty, Ashraf Hamdy, Ahmed Nasr Ouda. Flight Vehicle Autopilot System: From Design to Implementation. *Automation, Control and Intelligent Systems.* Vol. 6, No. 6, 2018, pp. 62-72. doi: 10.11648/j.acis.20180606.11

Received: February 19, 2019; **Accepted**: March 30, 2019; **Published**: April 26, 2019

Abstract: Recently, the tendency to reduce the human role is becoming an important step to overcome a human error during firing process in the military systems that may cause dangerous situations, especially anti-tank guided missile (ATGM) systems. Therefore, the researchers start to evaluate the automatic digital guidance and control unit before a real physical system integration in order to save their time, effort, money, and safety. This paper is dedicated to designing and analysis performance of the proposed anti-tank guided missile autopilot system and then moving to digital implementation on an embedded Linux system (ELS). Moreover, a developed procedure is carried out to confirm accurate digital implementation on an embedded system through the non-real time processor-in-The loop (PIL) approach. The intended missile modeling system is presented in the MATLAB environment. The proposed autopilot, in digital form, is implemented on the Raspberry Pi (RPI) system and connected to the main flight simulation environment through a serial communication protocol. The results confirm that the digital autopilot implementation on the embedded system is correct and the performance of the controlled plant is achieved all system requirements successfully.

Keywords: Digital Autopilot Implementation, X-in-The loop Test, System on Chip, Raspberry Pi

1. Introduction

The objective of the guidance and control unit in the missiles is changing the missile attitude during its flight trajectory to strike the target at the terminal stage by the steering control signal [1]. The steering control signal is shaped according to the error signal between the actual attitude and the required one. The efficiency of the guidance and control unit (Autopilot Unit) is measured by achieving time response requirements, overcoming different sources of disturbances and measurement noise as well as achieving minimum miss distance although the target maneuver [2].

A typical problem with the design of a feedback autopilot is to achieve at the same time a high performance for both time response and robustness and, from the computation point of view, the digital embedded implementation [3]. Therefore, for the autopilot design, although the majority of the controllers used in engineering systems especially in flight guidance and control system are still the classical Proportional-Integral-Derivative (PID) controllers, 2DOF PID is introduced as the optimal solution to get the advantages of classical PID and overcome its disadvantages [4, 5].

In addition, for digital implementation, the software engineers are used a development approach to reduce the cost, time, effort, and producing a rapid and reliable product in a short time development cycle before a real experimental test, which called X-in-The Loop Test. Each test provides some advances and reduces the gap in the development process that initiates with the mathematical model and ends at the firmware running in a stand-alone microprocessor platform. These stages carried out to have a green light to complete system test, in addition increasing the operator experience in interfacing with different analog and digital circuits and saving time and money, especially for military physical systems [6].

Recently, system-on-Chip (SoC) technology has large extension applications on different platforms, especially for unmanned guided systems such as robots, UAV's, missiles.

SoC provides great setoff facilities from high processor speed, large RAM, more communication interfaces, and so on. All of these features are added a great space for engineers to design and analyze their design's before the real-world interact, especially digital control implementations [7, 8].

2. Problem Formulation

The main problems of the missile systems have emerged because of the obsolescence of the electronic and mechanical parts, sharp changes in the external environment, and the human error during the firing process. These reasons have led to an increased chance of incorrect performance through the emergence of deviations in the expected path of the missile during flight trajectory. In addition, high cost and great effort are required for having a successful autopilot system for a guided plant such as missiles [1].

The present work is concerned with design and analysis an appropriate automatic autopilot system for the intended missile system, which means upgrade the current manually missile system to the automatic guided missile system in order to reducing the human-in-the-loop role in the firing process and enhancement of overall system performance. In addition, investigate a proposed autopilot performance on the hardware platform before starting Hardware-in-Loop (HIL) experimental tests through PIL development approach. Python programming language is used to program the RPI embedded system and serial communication protocol is used to transmit and receive the data during the test.

3. The Intended Flight Simulation Model

The intended system is representing one of the first generation ATGM, surface-to-surface type, manually tracking and manually guided, Thrust Vector Control (TVC) type, which means that the missile is guided via changing the generated thrust direction from the actuating system nozzle to correct the trajectory path during flight [9].

Modeling and simulation of the intended system are indispensable for design and analysis using either analogue and/or digital computers/processors. The first problem facing the designer in simulating a missile guidance system is the translation of its tactical characteristics into design specifications. These specifications carried out through a complete set of equations representing the missile motion and yielding the mathematical model [2]. This model constitutes the six degrees of freedom (6DOF) equations, which break down into those describing kinematics, dynamics (aerodynamics, thrust, and gravity), command guidance generation systems, and autopilot (electronics, instruments, and actuators) [3]. The input stimuli to this model are launch conditions, target motion, and target trajectory characterization and the outputs are the missile flight data (speed, acceleration, range etc.) during an engagement. Conducting the flight simulation helps to draw contributions about the reliability of conceptual hardware design and its effectiveness with minimum cost and no need for the very expensive flight trials, especially at the starting phase [10].

Thus, this section is devoted to present the complete set of equations representing the missile motion. Then, a simulation model of the intended missile is developed, with computer code written in the MATLAB environment in the form of different modules. Each module simulates a separate subsystem in the considered guidance loop with different engagement scenarios.

The digital simulation represents 6DOF model of the intended anti-tank missile systems, with comprehensive modelling of the aerodynamics, rocket motor, command generation system, inertial reference frames, autopilot, body mounted sensors (position gyro), moving control actuators, wind effect, and atmospheric condition. The simulation model can be broken down into major parts: missile-target geometry, guidance, autopilot, airframe, and kinematics are shown in Figure 1 [11].

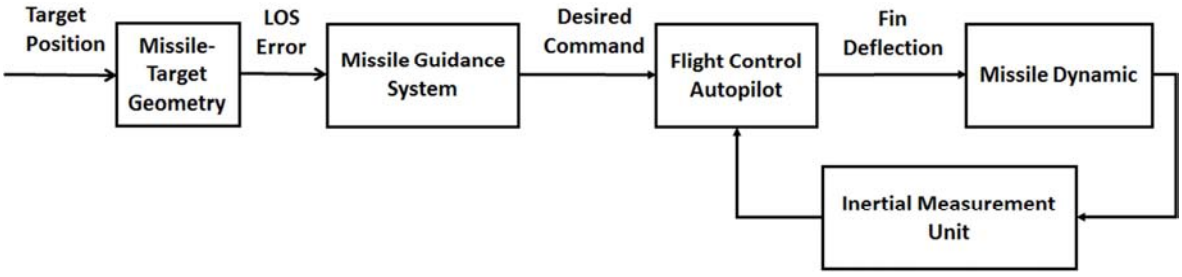

Figure 1. *Block Diagram of the Flight Simulation Model.*

A computer code written in the MATLAB environment is employed to solve the mathematical model of the intended missile system. The mathematical model has been structured such that a series of modules are defined which can be individually developed and if necessary more modules incorporated. Each module simulates part in the missile system and all modules connected together to complete the missile closed loop function. Runge Kutta 4 is used to solve numerically differential equations handled in the simulation program. A flowchart illustrating the simulation process sequence of missile-target engagement is shown in Figure 2.

Figure 3 reveals the flow data path in the intended flight simulation model clearly. Starting, with the missile target geometry and given the motion of the designated target, the relative motion computer determines the deviations from the desired trajectory. These errors are detected by the missile

sensor and yield error signals, which are applied to the control system. Then, the control system changes the actuating system nozzle position and receives feedback from the actual missile motion to allow for maneuvers. The new nozzle deflection causes change in thrust force components leading to the required flight course corrections.

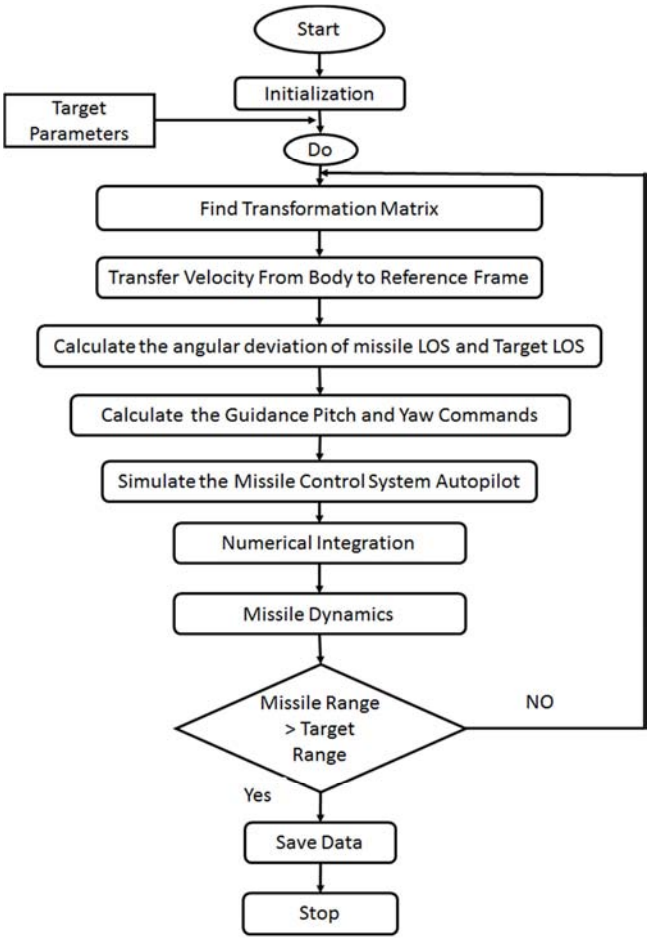

Figure 2. *Block Diagram of the Flight Simulation Model.*

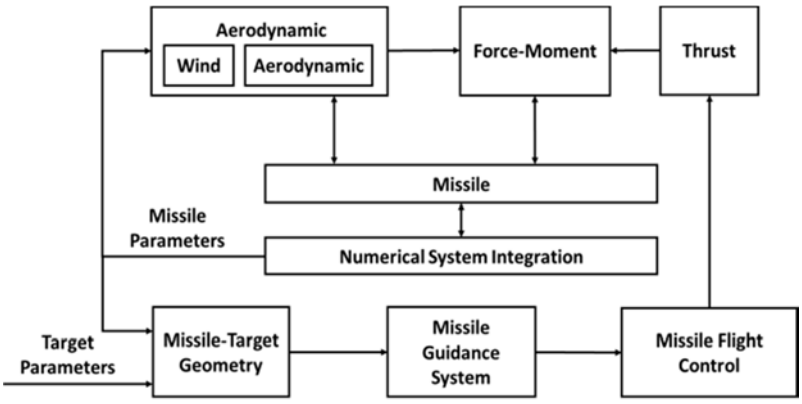

Figure 3. *6DOF Simulation Diagram.*

For having an ability to deal with the flight vehicle problems well, the description of the various dynamical parameters including position, velocity, acceleration, forces, and moments should be considered correctly. In addition, the forces acting on the missile, including weight, thrust, and aerodynamic forces, have different mother frames of reference and consequently coordinates transformation from a frame to another is indispensable. This transformation is carried out using Euler's angles transformation method.

3.1. Reference Frames and Coordinates Transformations

A coordinate system must be established to describe the vehicle position in space uniquely in the form of range and body attitude with respect to a specified reference frame. The mathematical model for the system under consideration,

consist of different parameters describing the missile position, velocity, acceleration, forces, and moments [2, 10]. The forces that act on the missile are weight, thrust, and aerodynamic forces. The necessity of having more than one coordinate system is attributed to the fact that these forces originated in the different coordinate systems employed. Thus, formulas must be available for transforming these parameters from one frame to another. This transformation carried out using Euler's angles transformation method.

3.1.1. Ground-Body Coordinate Systems

The coordinates' transformation from the body into the ground coordinate system using Euler's angles can be carried out using the following transformation matrix, based on Figure 4:

$$T_{bg} = \begin{bmatrix} c\Psi\, c\theta & c\Psi\, s\theta\, s\phi - c\phi\, s\Psi & s\theta\, c\phi\, c\Psi + s\phi\, s\Psi \\ s\Psi\, c\theta & s\Psi\, s\theta\, s\phi + c\Psi\, c\phi & c\phi\, s\theta\, s\Psi - s\phi\, c\Psi \\ -s\theta & s\phi\, c\theta & c\phi\, c\theta \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} = T_{bg} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad (2)$$

Where x_1, y_1, and z_1 are the vectors components along the board system axes whereas x_g, y_g, and z_g are the vectors components along the ground reference coordinate system axes.

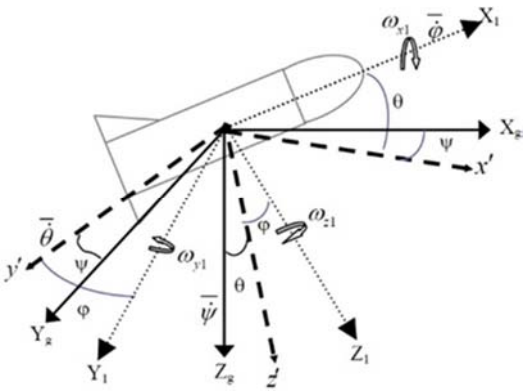

Figure 4. *Angles between Ground and Body Axes.*

3.1.2. Ground-Velocity Coordinate Systems

The coordinates' transformation from ground into the velocity coordinate system can carried out using the following transformation matrix, based on Figure 5:

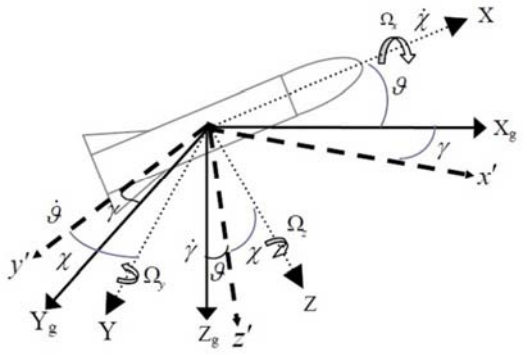

Figure 5. *Angles between Ground and Velocity Axes.*

$$T_{vg} = \begin{bmatrix} c\gamma\, c\vartheta & c\gamma\, s\vartheta\, s\chi - c\phi\, s\gamma & s\vartheta\, c\chi\, c\gamma + s\chi\, s\gamma \\ s\gamma\, c\vartheta & s\gamma\, s\vartheta\, s\chi + c\gamma\, c\phi & c\chi\, s\vartheta\, s\gamma - s\chi\, c\gamma \\ -s\vartheta & s\chi\, c\vartheta & c\chi\, c\vartheta \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} = T_{vg} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (4)$$

Where x, y, and z are the vectors components along the velocity system axes; where as x_g, y_g, *and* z_g are the vectors components along the ground reference coordinate system axes.

3.1.3. Velocity-Body Coordinate Systems

The coordinates' transformation from velocity into the body coordinate system can carried out using the following transformation matrix, based on Figure 6:

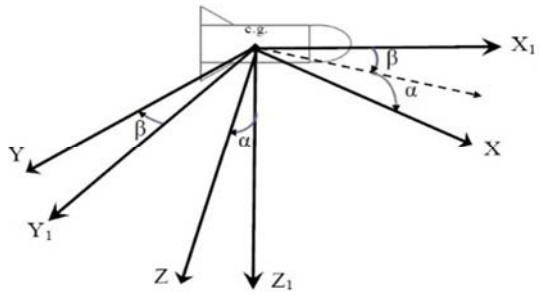

Figure 6. *Angles between Velocity and Body Axes.*

The thrust forces that act on the missile are inclined by angles δ_{jp} and δ_{jy} in the pitch and yaw planes. The thrust forces and moments are shown in Figure 7.

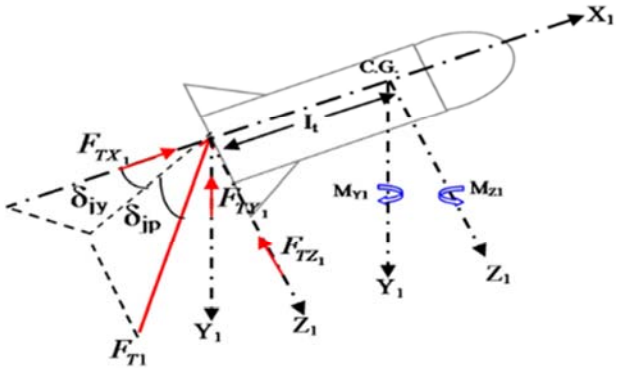

Figure 7. *Thrust Forces and Moments Acting on the Missile.*

3.2. Equation of Motion

The equations of missile motion include three translational equations and three rotational equations. The simulation uses these nonlinear-coupled differential equations that describe the behavior of a rigid missile can be summarized as follows:

$$\sum F_x = mg_x + F_{Tx} + F_{AX} = m\dot{v_m}$$

$$\sum F_y = mg_y + F_{Ty} + F_{Ay} = mv_m\Omega_z$$

$$\sum F_z = mg_z + F_{Tz} + F_{Az} = -mv_m\Omega_y \quad (5)$$

$$J_X = \dot{v_m}$$
$$J_Y = v_m \Omega_z$$
$$J_Z = v_m \Omega_y \quad (6)$$

$$M_{x_1} = I_{xx}\dot{\omega_{x_1}} + (I_{zz} - I_{yy})\omega_{z_1}\omega_{y_1}$$
$$M_{y_1} = I_{yy}\dot{\omega_{y_1}} + (I_{xx} - I_{zz})\omega_{z_1}\omega_{x_1}$$
$$M_{z_1} = I_{zz}\dot{\omega_{z_1}} + (I_{yy} - I_{xx})\omega_{x_1}\omega_{y_1} \quad (7)$$

$$\dot{\theta} = \omega_{y_1}\cos\phi - \omega_{z_1}\sin\phi$$
$$\dot{\Psi} = (\omega_{z_1}\cos\phi + \omega_{y_1}\sin\phi)/\cos\theta$$
$$\dot{\phi} = \omega_{x_1} + (\omega_{z_1}\cos\phi + \omega_{y_1}\sin\phi)\tan\theta \quad (8)$$

Where Eq. (5) represents force components in the velocity reference frame, Eq. (6) represents the acceleration components in the velocity reference frame, Eq. (7) represents moment components in body reference frame, Eq. (8) represents missile rotation around its center of gravity (c.g.).

3.3. Flight Simulation Model

A flight simulation model depends on the mathematical model, which is developed to obtain the flight trajectory of the missile. A block diagram that shows the flow sequence of data among the various subsystems can be broken down into the following major parts: Ground Station (Manually target tracking, operator intervention, Guidance system, Command signal generator), On-board missile system (Control unit, attitude sensor, Actuating system, missile airframe), Dynamics and kinematics of the missile as shown in Figure 8.

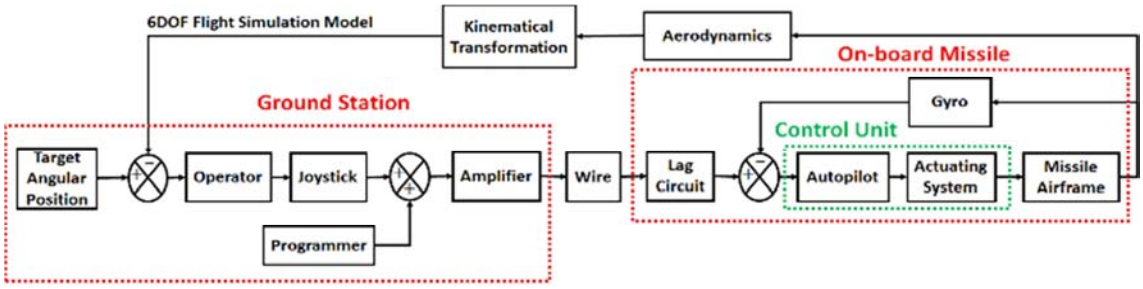

Figure 8. *Block Diagram of the Intended 6DOF Flight Simulation Model.*

3.4. Flight Simulation Model Evaluation

For the evaluation standpoint, the flight path trajectory, the output of the intended flight simulation model, is compared against the real reference data for a target at distance 2500[m] with different thrust profiles. Figure 9 reveals that the consistency of the simulation model, which is programmed under MATLAB environment and the real reference data.

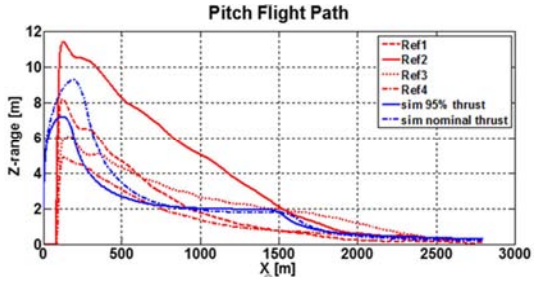

Figure 9. *The Consistency between Simulation Model and Real Reference Data.*

4. The Intended Missile Control System

For the intended system, the autopilot control system is one of the major building blocks of the platform model as shown in Figure 8. However, due to the nature of the missile maneuver, it is has a nonlinear aerodynamic characteristic that can be linearized at some conditions and consequently, their transfer function can be obtained to design and analysis the proposed autopilot.

This module simulates the current missile control system. It converts the derived guidance signal, with some feedbacks, into the control signal that causes the actuating system nozzle deflection. The output applied to the thrust module for calculating the thrust force of the missile as shown in Figure 8. The following transfer functions extracted from experience gained from previous searches and work with the system. The transfer function of the wire in pitch and yaw planes:

$$\frac{i_w}{i_{tot}} = \frac{1}{0.001062\,s^2 + 0.077\,s + 1} \quad (9)$$

Where i_{tot} is the total command current, which is the summation of drive amplifier current and programmable ground unit current, and i_w is the command current through the wire, then the transfer function of the lag circuit in pitch and yaw planes:

$$\frac{V_{lag}}{i_w} = \frac{0.022}{0.008\,s + 1} \quad (10)$$

The transfer function of the gyros in pitch and yaw planes:

$$\frac{V_g}{r_{\theta,\Psi}} = \frac{0.044}{0.002\,s + 1} \quad (11)$$

Where V_s is the electronic-pack voltage, and V_e is the error voltage between lag voltage and gyro voltage. The transfer function of the jetevator servo in pitch and yaw planes:

$$\frac{\delta}{V_s} = \frac{2.85}{0.0000633\,s^2 + 0.0079\,s + 1} \quad (12)$$

Where δ is the jet deflection angle.

Figure 10 shows the scope in the control section of the flight simulation model and its input and output of each sub-element to identify the operation sequence during flight control process. Where V_g free gyro voltage output, i_{tot} total current sending from the control station to missile in wire, V_{lag} lag circuit output voltage according to i_w (wire current), V_e error voltage between desired and actual output, V_s output of autopilot to generate accelerated force to change actuating system position with angle (δ).

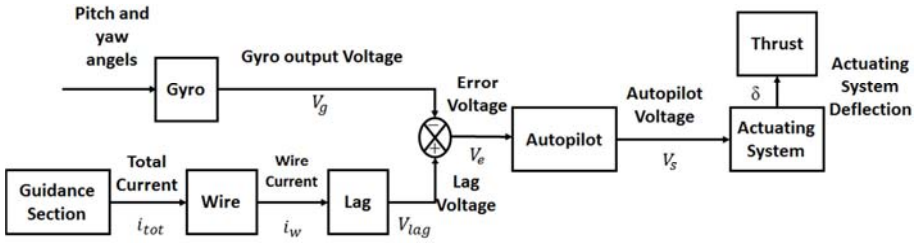

Figure 10. *Block Diagram of Control Section.*

Thus, the overall plant transfer function obtained as:

$$\frac{\theta}{\Delta_{p_{command}}} = \frac{-1.362*10^5\, s - 5.533*10^4}{s^5 + 392.6\, s^4 + 4.935*10^4\, s^3 + 4.246*10^6\, s^2 + 2.005*10^6\, s} \tag{13}$$

The next step is improving the overall system performance by redesign autopilot depend on modern control techniques, that called 2DOF PID control theory.

5. Autopilot Design and Analysis

In many control applications, the plant can be considered linear within defined regions of operation, which may lead to a set of linearized models at specific set points, or trim condition [12-14]. Therefore, the nonlinear capabilities, e.g. position, rate and acceleration limits, of actuation device are adequate for the application is being considered to prevent any uncontrollable or unstable condition developing [15].

The main problem of achieving at the same time a high performance for both time response and robustness can be solved by designing a 2DOF PID control architecture, namely, a combined feedforward/feedback control law that classical PID controller does not achieve [16, 17]. The previous analysis was developed to ensure the proposed autopilot achieve all system performance requirements especially in set point following and load disturbance rejection because of 2DOF PID controller has advantages of classical PID controller also including the solution of its problems [11, 18].

First, adding the desired autopilot in the linearized actuating system model, where the autopilot consists of pre-filter used to smooth the output of the lag circuit and then summation occurs with the feedback gyro voltage to produce the error voltage signal as shown in Figure 11. It is possible to obtain the overall autopilot transfer function of the system through a suitable combination utilizing some basic rules of block diagram transformation to reduce the original diagram [16]. In order to represent the desired controller in one block diagram to test under software-in-loop pre-filter block diagram shifted after summation point and the gyro simulated feedback voltage is modified to modified voltage via multiplication by the inverse of the pre-filter circuit as shown in Figure 12.

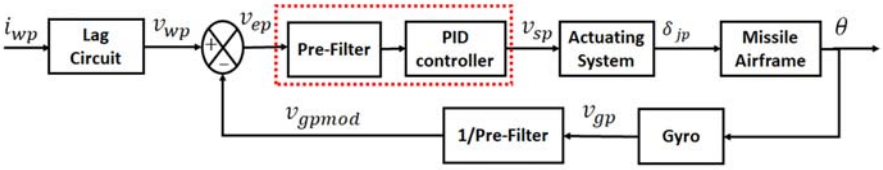

Figure 11. *2DOF PID Autopilot Controller.*

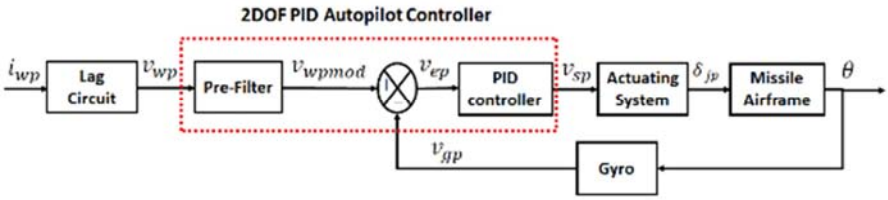

Figure 12. *Equivalent Autopilot Diagram.*

The equivalent transfer function of autopilot represented in s-domain as:

$$\frac{v_{sp}}{v_{ep}} = \frac{13244\, s + 89080}{946\, s + 4454} \tag{14}$$

Figure 13 is clarified that the designed controllers have a faster transient response than the original one. In addition, the increasing set point weighting effect appeared as a reduction of overshoot and increasing the rising time of the system. However, the designed controller with different set point weighting value has a lower control effort at the steady state compared to the classic controller.

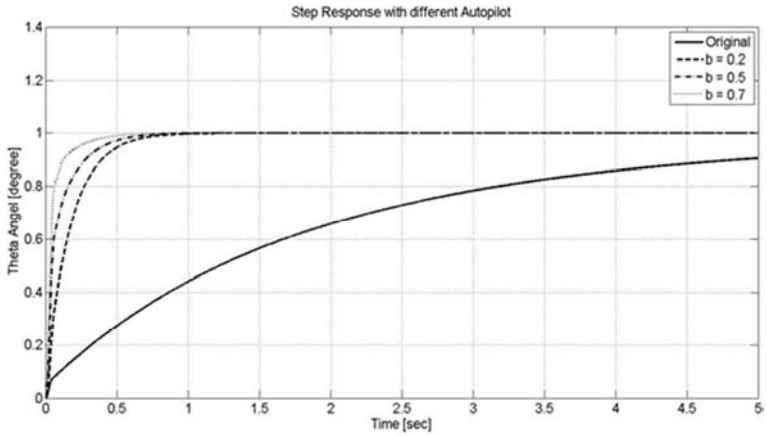

Figure 13. *Step Response of the Original and Obtained Controllers.*

In addition, applying white Gaussian noise to the gyro output, the step response with different setpoint weighting is shown in Figure 14, which clarify that the designed controller is less sensitive to additive noise, compared to conventional one. In addition, applying disturbance to the actuating system output, the obtained step response of closed-loop system shown in Figure 15, which clarifies that the convergence using the designed controller, is the best compared to the classic controller as it rejects 50% within 0.1 sec and 95% within 0.25 sec.

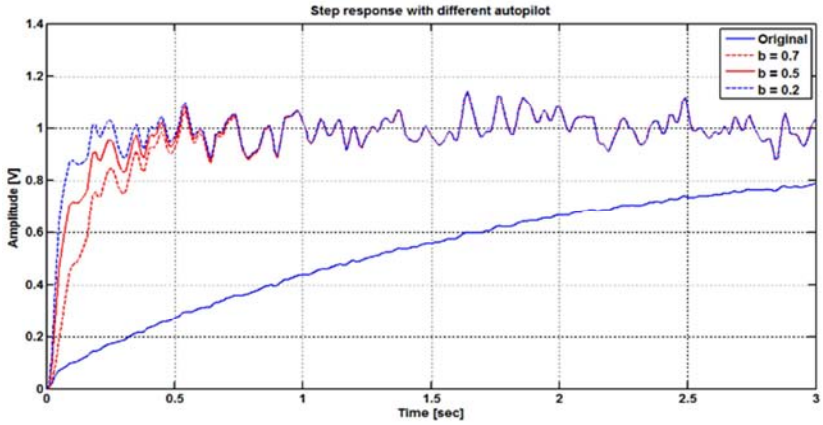

Figure 14. *Step Response with Applying Measurement Noise.*

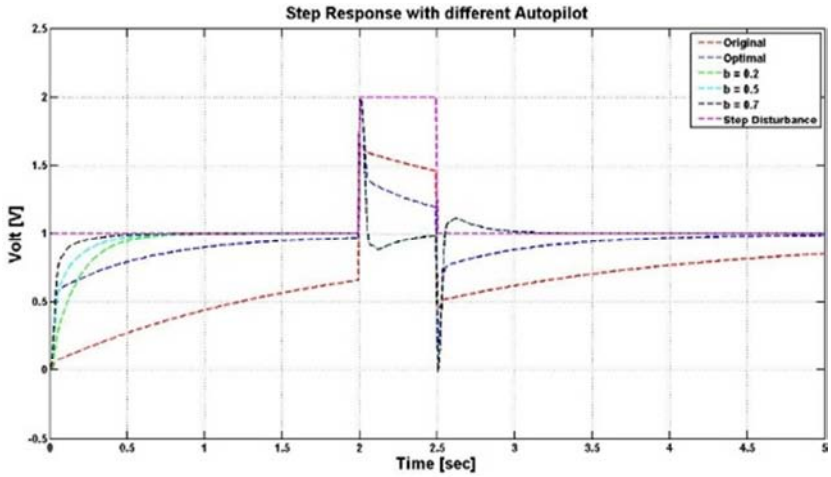

Figure 15. *Step Response of the Original and Obtained Controller with Disturbance.*

In non-linear flight simulation model, the flight path evaluation will be considered w.r.t. different viewpoints to evaluate design autopilot performance under different target scenarios, degradation in thrust, aerodynamic variation, and wind effect as an external disturbance source [13, 19, 20]. The proposed controller is evaluated with the flight trajectory at the minimum and maximum tactical data (500 [m], 2800 [m]), respectively. The flight trajectory with conventional and proposed autopilot for a fixed target at the different distance is shown in Figure 16 and Figure 17 respectively. The miss-distance and variance of the control effort signal of the above evaluations summarized in Table 1.

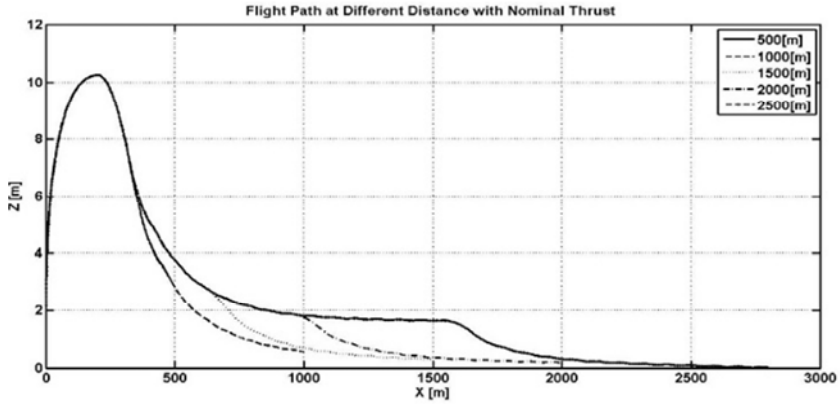

Figure 16. *Trajectory Obtained with Conventional Autopilot.*

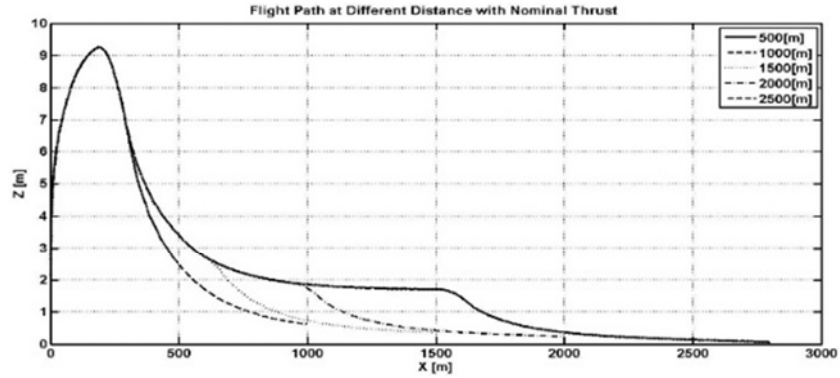

Figure 17. *Trajectory Obtained with the proposed Autopilot.*

Table 1. *Designed Autopilots Evaluation via Miss-Distance and Variance of Control Effort.*

Fixed target at Nominal Thrust	Original Controller		2DOF PID Controller	
	Miss-Distance [m]	Variance of Control Effort	Miss-Distance [m]	Variance of Control Effort
500 [m]	2.475	69.11	2.18	483.3
1000[m]	0.616	102.44	0.55	442.2
1500[m]	0.359	174.73	0.305	372.15
2000[m]	0.235	193.21	0.1708	312.9
2500[m]	0.132	189.94	0.0989	276.02

In addition, the proposed autopilot is evaluated with the flight trajectory against classical autopilot using different thrust values with different target position. Also with 80% of nominal thrust value variation, against perturbations in the aerodynamic coefficient of about ±20%, and effect of the wind appeared as the input disturbance to the system during the flight path trajectory for minimum and maximum tactical rang 500[m], 2800[m] respectively. All of the different scenarios of the proposed autopilot is achieved all control system requirements with the accepted flight path and accepted minimum miss-distance.

Next step is converted autopilot equivalent transfer function from continuous time description to discrete time description using a suitable discretization mechanism for digital representation.

6. Autopilot Implementation

The implementation of the processor in the loop done by sequential steps. The designed controller, originally designed for continuous time systems, must be adapted to discrete time application. The appropriate selection of the sampling period T is a crucial factor in digital controller design since if this period is too large there are problems in the signal reconstruction, and if it is too small, system instability and processing capacity problems can occur.

The equivalent transfer function of autopilot is converted from s-domain (continuous time) to z-domain (discrete time) with an approporate sampling time (0.0313[sec.]). According to sampling time and order of the equivalent transfer function, zero order method is used to discretized autopilot with the same analog prototype behavior.

$$\frac{u(z)}{e(z)} = \frac{A\,z - B}{C\,z - D} \tag{15}$$

Moreover, the more applicable one represented as:

$$\frac{u(z)}{e(z)} = \frac{14\,z - 11.26}{z - 0.863} \tag{16}$$

In order to represent the discrete transfer function in different embedded system must be change from differential equation to difference equation as shown in the following:

$$e(z) * (A - B\,z^{-1}) = u(z) * (C - D\;z^{-1}) \tag{17}$$

$$A * e(z) - B * e(z) * z^{-1} = C * u(z) - D * u(z) * z^{-1} \tag{18}$$

When B=1, the final equation can represented as:

$$u(n) = A * e(n) - C * e(n-1) + D * u(n-1) \tag{19}$$

7. X-in-The Loop Development Approach

The rapid development in the usage of the embedded systems leads researchers all over the world to have a development approach to save their time, money, and efforts during design, analysis, and implementation real physical systems, especially for military physical systems like missiles [21].

This X-in-The Loop development approach provides four tests, which are MIL (Model-In-The-Loop), SIL (Software-In-The-Loop), PIL (Processor-In-The-Loop) and HIL (Hardware-In-The-Loop) configuration respectively. The main objectives of these tests reduce the time of development of embedded systems, reliable prototype, and rapid development processing cycle in a short time.

First, MIL is based on design and analysis of the integrated controller model with the underlying system model. The objective of this test generates reference results of the proposed controller that can validate the controller behavior.

Second, in the SIL test, the simulated controller is replaced by an executable code running at the same computer. This test provides a correct choice of the variables memory size and a solid background in software programming.

Third, PIL test introduces the real connection with a hardware component, as well as, good experience with dealing with the external processors rather than PC or main platform environment. This test is a non-real time test; therefore, the objective of the test breaks the full simulation environment and confirm the controller firmware will run correctly in the standalone processor platform.

At the last one, HIL is the final step to have the full correct embedded controller system. The embedded controller is connected with a real interface with the electrical emulation of sensors and actuating systems in a real-time target platform to evaluate the real-time controller performance at the real situation before installing to real physical plant.

8. Processor-in-The Loop Test

The SIL simulation is made using the MATLAB software and after, the PIL simulation is carried out using a Raspberry Pi Platform to the implantation of the autopilot system while the underlying system dynamics is implemented in PC platform and coded by MATLAB software [11, 22, 23].

PIL requires drivers to communicate the computer platform with the aimed hardware. The resulting object code generated in the PC links with other test-management functionality and then downloaded, typically to an off-the-shelf evaluation board with the target processor. The simulation tool, running on the PC machine, then communicates with the downloaded software, typically via a serial communication link.

Flight simulation model in the main simulation environment is connected to the raspberry pi system with a serial communication protocol. The designed 2DOF PID autopilot implemented on the raspberry pi system as a processor-in-Loop part to evaluate the digital autopilot description on the embedded hardware system. The experimental procedure carried out for system evaluation and validation as shown in Figure 18.

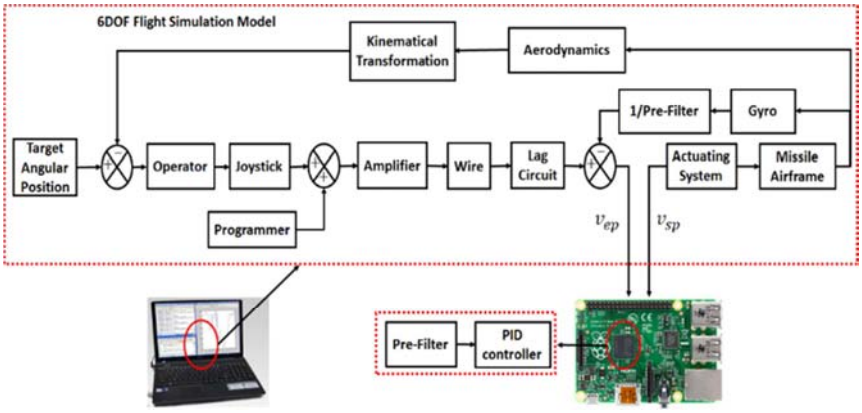

Figure 18. *Processor-in-The Loop Experimental Setup.*

The design autopilot evaluated with the flight path trajectory in the pitch plane against simulated designed autopilot and PIL embedded plate form hardware at the different target position. Figure 19 shows the flight path trajectory minimum and maximum tactical rang 500[m], and 2800[m] respectively. In addition, the error signal between the required position and actual position is shown in Figure 20.

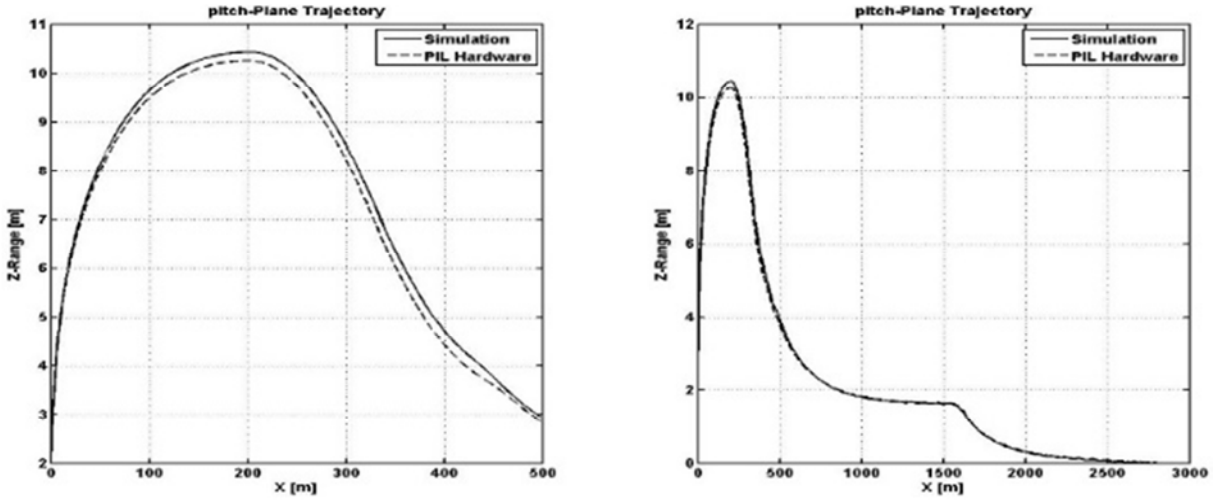

Figure 19. *Missile Trajectory in Pitch Plane.*

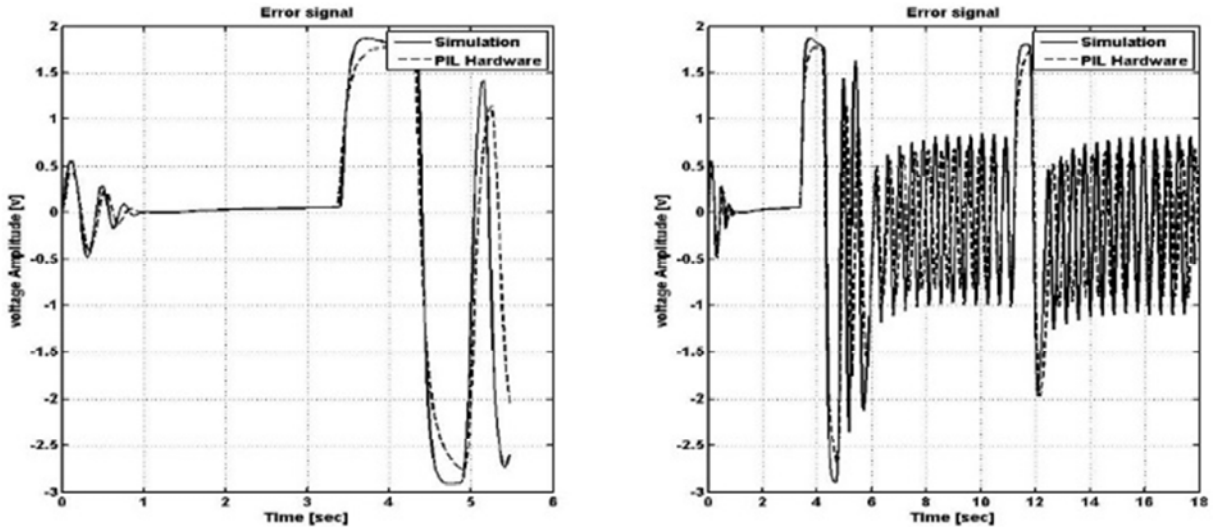

Figure 20. *Error signal.*

Table 2 shows the experimental results between autopilot simulation in flight simulation model and Processor-in-Loop experimental test.

From the above results, the processor-in-loop experimental test carried out to evaluate digital autopilot description, which simulates the continuous autopilot on flight simulation model, based on the raspberry pi embedded system. From the results, the digital autopilot is achieved the tactical specification with the accepted flight path and accepted miss distance. The miss-distance of experimental lower than original autopilot, on the other hand, the variance of control effort is higher than the original autopilot. The simulation-designed autopilot has lower miss-distance than experimental due to precise of data transfer from different plate form using serial communication protocol. The control effort value is calculated using fast Fourier transformation technique [24].

Table 2. *Processor-in-loop experimental results.*

Experimental with Fixed Target	Target Distance [m]	Miss-Distance [m] (< 3[m])	Variance of Control Effort
Original Controller	500	2.4755	69.1095
	2800	0.0609	183.1107
Designed Controller	500	2.18	483.3027
	2800	0.0087	271.2906
PIL Hardware	500	2.2908	628.2199
	2800	0.0550	361.2107

9. Conclusion

Having an digital autopilot systems is a great step to increase the over all missile system performance and reduce the human intervention for firing process. Moreover, the advanced control techniques provide an optimal solution to have approporate control system specifiction from robustness and time tresponse, in addition, computational standpoint.

2DOF PID control theory reveals the flexibility to get the advantages of classical linear feedback control theory and have a approporate processing time for system like missiles. Moreover, the autopilot introduce great effectiveness to overcome the unlinearity of the intended sytems, overcome the uncertinities, different disturbance sources, and the measurement noises.

The X-in-The Loop development approach provides a clear, easy, and rapid development of applications for embedded systems implementation, in addition, design and analysis of controllers by sequential process immensely. As well as open the way to studies more realistic physical problems, which depend on several factors, including communication protocol between the controlled system and the embedded system processor, data exchange rate, time delay, reliability of transmitting and receiving data, and processing time of the controller.

References

[1] N. Iyer, "Recent Advances in Antitank Guided Missile Systems," *Defence Science Journal,* vol. 45, p. 187, 1995.

[2] P. Zarchan, "Tactical and strategic missile guidance," *Progress in astronautics and aeronautics,* 2002.

[3] A. E. Özcan, "Autopilot and guidance for anti-tank imaging infrared guided missiles," *Yüksek Lisans Tezi, ODTÜ Elektrik ve Elektronik Mühendisliği Bölümü,* 2008.

[4] B. G. Abdelaty, A. H. Ahmed, and A. N. Ouda, "Fixed Set Point Weighting 2DOF PID Controller for Control Processes," 2018.

[5] A. Visioli, *Practical PID control*: Springer Science & Business Media, 2006.

[6] V. Garousi, M. Felderer, Ç. M. Karapıçak, and U. Yılmaz, "What we know about testing embedded software," *IEEE Software,* vol. 35, 2018.

[7] L. Zhang, F. Deng, J. Chen, Y. Bi, S. K. Phang, X. Chen*, et al.*, "Vision-Based Target Three-Dimensional Geolocation Using Unmanned Aerial Vehicles," *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS,* vol. 65, 2018.

[8] B. E. Gamal, A. N. Ouda, Y. Z. Elhalwagy, and G. A. Elnashar, "Embedded target detection system based on raspberry pi system," in *Computer Engineering Conference (ICENCO), 2016 12th International*, 2016, pp. 154-157.

[9] C.-M. Lin, "Fuzzy-logic-based CLOS guidance law design," *Aerospace and Electronic Systems, IEEE Transactions on,* vol. 37, pp. 719-727, 2001.

[10] C.-F. Lin, *Modern navigation, guidance, and control processing* vol. 2: Prentice Hall Englewood Cliffs, 1991.

[11] T. Ashish, "Modern control design with Matlab and simulink," *Indian Institute of Technology, Kanpur, India, John Wiley & Sons,* 2002.

[12] S. Cong, G. Li, and X. Feng, "Parameters identification of nonlinear DC motor model using compound evolution algorithms," in *Proceedings of the World Congress on Engineering*, 2010.

[13] I. Mizumoto, H. Tanaka, and Z. Iwai, "2 DOF adaptive PID control with a parallel feedforward compensator for nonlinear systems," in *Networking, Sensing and Control, 2009. ICNSC'09. International Conference on*, 2009, pp. 261-266.

[14] A. S. Morse, "Supervisory control of families of linear set-point controllers-Part I. Exact matching," *IEEE transactions on Automatic Control,* vol. 41, pp. 1413-1431, 1996.

[15] V. M. Alfaro and R. Vilanova, "Control System Evaluation Metrics," in *Model-Reference Robust Tuning of PID Controllers*, ed: Springer, 2016, pp. 21-28.

[16] Y. Chen and D. P. Atherton, *Linear feedback control: analysis and design with MATLAB* vol. 14: Siam, 2007.

[17] R. Vilanova and O. Arrieta, "PID design for improved disturbance attenuation: min max Sensitivity matching approach," *IAENG International Journal of Applied Mathematics,* vol. 37, 2007.

[18] G. J. Silva, A. Datta, and S. P. Bhattacharyya, "New results on the synthesis of PID controllers," *Automatic Control, IEEE Transactions on,* vol. 47, pp. 241-252, 2002.

[19] V. Alfaro, R. Vilanova, and O. Arrieta, "Robust tuning of Two-Degree-of-Freedom (2-DoF) PI/PID based cascade control systems," *Journal of process control,* vol. 19, pp. 1658-1670, 2009.

[20] P. Ananthababu, B. A. Reddy, and K. R. Charan, "Design of Fuzzy PI+ D and Fuzzy PID Controllers Using Gaussian Input Fuzzy Sets," in *Emerging Trends in Engineering and Technology (ICETET), 2009 2nd International Conference on*, 2009, pp. 957-961.

[21] H. Min, Z. Guoqiang, Y. Hong, and T. Yafeng, "Processor-in-the-loop demonstration of coordination control algorithms for distributed spacecraft," in *Information and Automation (ICIA), 2010 IEEE International Conference on*, 2010, pp. 1008-1011.

[22] W. C. Messner, D. M. Tilbury, and A. P. R. Hill, *Control Tutorials for MATLAB® and Simulink®*: Addison-Wesley, 1999.

[23] A. K. Singh and A. Pandey, "Intelligent PI Controller for Speed Control of SEDM using MATLAB," *International Journal of Engineering Science and Innovative Technology (IJESIT),* vol. 2, 2013.

[24] A. Ouda, "A robust adaptive control approach to missile autopilot design," *International Journal of Dynamics and Control,* vol. 6, pp. 1239-1271, 2018.