



Autopilot Implementation Synthesis via Processor-in-Loop Test

Bahaaeldin Gamal Abdelaty^{*}, Ahmed Nasr Ouda[†], Yehia Zakaria Elhalwagy[‡]
 Gamal Ahmed Elnashar[§]

Abstract: This paper presents design and implementation of 2DOF PID autopilot capable of dealing with the issue related to the missile aging, uncertainty in aerodynamic coefficient calculations, and different disturbance sources that affects the missile tactical specifications. 2DOF PID autopilot is carried out to achieve both control objective of follow reference signal and system robustness, which better than traditional PID. This paper presents redesign a 2DOF PID appropriate autopilot capable of dealing with the issue related to the missile aging, uncertainty in aerodynamic coefficient calculations, different disturbance sources and thrust degradation that affects the missile tactical specifications, which carried out with linearized mathematical model of actuating system and missile airframe and nonlinear flight simulation model under MATLAB environment. The overall block diagram of the underlying missile system have to be presented with mathematical model of the linearized underlying missile airframe and actuating system. The 2DOF PID autopilot has to be evaluated to clarify the merits and demerits of it and implemented based on embedded raspberry pi system. At the end, processor-in-loop experimental test is used to validate the proposed digital autopilot performance, which is correctly described on embedded system and achieved the required system response more than the conventional one with accepted flight path and minimum miss-distance.

Keywords: Autopilot Design, 2DOF PID Controller, Disturbance Rejection, Raspberry pi

1. Introduction

Performance of antitank guided missile systems measured through the minimum miss distance and the capability of the missile to overcome target maneuver and different sources of disturbance and measurement noise. The objective of guidance process is to correct the missile trajectory through its flight and to overcome the external and internal sources of disturbance by moving actuators according to the steering control signal (autopilot output) to change the missile attitude [1].

A typical problem with the design of a feedback autopilot is to achieve at the same time a high performance in the set point following and in the load disturbance rejection. The majority of the controllers used in engineering systems especially in flight guidance and control system and are still the Proportional-Integral-Derivative (PID) controllers [2].

^{*} Technical Research Center, Egyptian Armed Force, Egypt; bahaa.blackageel@yahoo.com

[†] Technical Research Center, Egyptian Armed Force, Egypt; ahnasroda@yahoo.com

[‡] Military Technical College, Egyptian Armed Force, Egypt; ehalwagy@hotmail.com

[§] Military Technical College, Egyptian Armed Force, Egypt; g.elnashar@mtc.edu.eg

This is due to their simple structure from both mathematical and computation point of view, easy implementation, and adequate performances. However, tracking with conventional PID controllers over the operating range of highly nonlinear uncertain systems is difficult to achieve with excellent performances. In other words, for such systems conventional PID controllers lack their credibility, reliability, and robustness.

The 2DOF PID controller is considered as a suitable solution to overcome the traditional PID controller disadvantages by set point weighting parameters [2, 3]. In addition, evaluation and validation of the proposed digital autopilot must be achieved using development approach methods. One of these methods is X-in-loop development approach, which have different stages to reach complete system test finally [4]. These stages carried out to have a green light to completely system test, in addition increasing the operator experience in interfacing with different analog and digital circuits and saving time and money especially for digital autopilot [5]. In addition, the system on chip (SOC) technology have a large extension application to be used for implementation on different plate form such as raspberry pi system [6, 7].

2. Problem Formulation

The main problems have emerged after the life span of the missile are obsolescence of the electronic and mechanical parts, and sharp changes in the external environment has led to an increased chance of incorrect performance due to emergence of deviations in the expected path of the missile during flight trajectory. The present work is concerned with improving the performance of underlying missile system via redesign an appropriate autopilot to deal with different disturbance sources by advanced synthesis of control system. In addition, implemented the proposed autopilot on embedded system and validated via appropriate suitable development approach.

3. Generic underlying Platform Description

The underlying missile system is represented one of the first generation, surface-to-surface guided missile, wire command-link, and thrust vector control to correct the trajectory path during flight. The block diagram of 6DOF flight simulation model is shown in figure (1) [1].

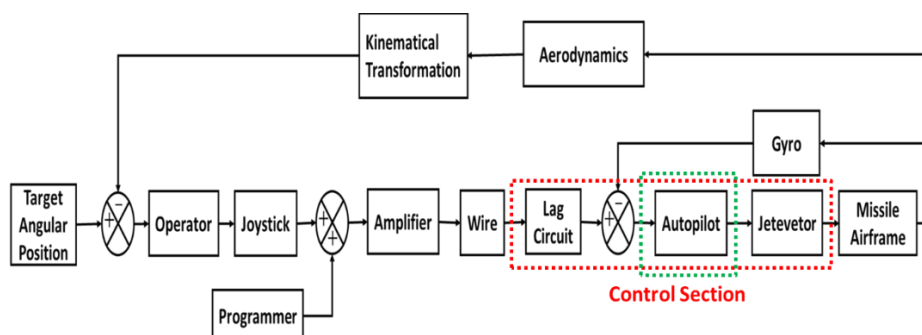


Fig. (1) Non-linear flight simulation model block diagram

For the underlying missile guidance system, the autopilot control system is the major building block that necessitate mature designs as shown in figure (1). However, due to the nature of the missile flight, its nonlinear aerodynamic characteristics can be linearized at some conditions and consequently their transfer function obtained for design and analysis [1].

The present work is concerned with pitch plane as a case study. The overall pitch airframe transfer function as a nominal airframe which have a moderate frequency response compared

with the remainder set points is the jetevator ($\delta_{jp}/\Delta_{p_{command}}$) cascaded with the airframe (θ/δ_{jp}) as shown in figure (2) [1, 8]. The proposed autopilot has designed based on suitable operating point to evaluate the system performance.

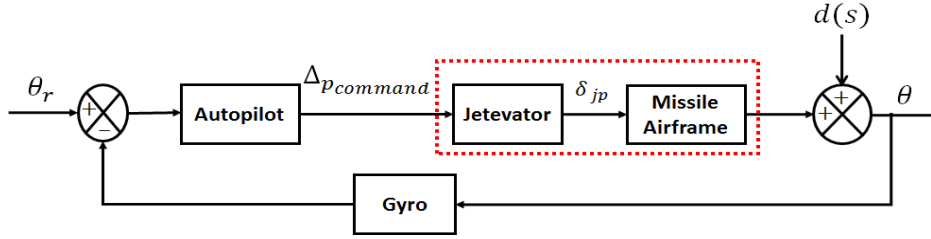


Fig. (2) Block diagram of missile airframe

Thus, the overall plant transfer function obtained as:

$$\frac{\theta}{\Delta_{p_{command}}} = \frac{-1.362 * 10^5 s - 5.533 * 10^4}{s^5 + 392.6 s^4 + 4.935 * 10^4 s^3 + 4.246 * 10^6 s^2 + 2.005 * 10^6 s} \quad (1)$$

The present work is concerned with improving the overall system performance by redesign autopilot depend on modern control techniques.

4. Autopilot Design and Analysis

This problem can be approach by designing a 2DOF control architecture, namely, a combined feed forward/feedback control law. It improves reference track response by providing additional tuning parameters (b) that allows independent control of the impact of the reference signal on the proportional action. In figure (3) 2DOF PID controller structure shown. There is a feedback controller C and a feed-forward filter F [3]. The 2DOF PID control law can be obtain as:

$$u(s) = K_p \left[(b r(s) - y(s)) + \frac{1}{sT_i} (r(s) - y(s)) + \frac{sT_d}{1 + \frac{sT_d}{N}} (cr(s) - y(s)) \right] \quad (2)$$

$$F(s) = \frac{1 + b * T_i * S + T_i * T_d * S^2}{1 + T_i * S + T_i * T_d * S^2} \quad (3)$$

Parameters b and c called set-point weights and constitute a simple way to obtain a 2DOF controller.

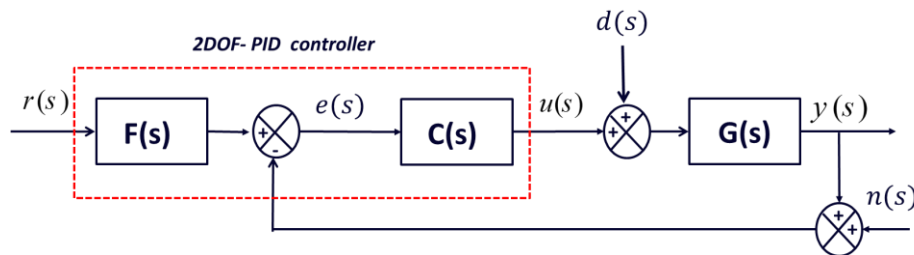


Fig. (3) 2DOF PID Controller Structures

It appears that the load disturbance rejection decoupled from the set point following one and obviously does not depend on the weight b . Thus, the PID parameters can be select to achieve a high load disturbance rejection and then the set point following performance can be

recover by suitably selecting the value of the parameter b . Where, in general, it is $0 \leq b \leq 1$ and $0 \leq c \leq 1$, although the value of c usually either 0 (the derivative action is entirely applied to the process output) or 1 (the derivative action is entirely applied to the control error). The previous one is usually call a PID controller in ISA form [2, 3, 9].

Figure (4) shows the scope in the control section of 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 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 jetevator position with angle (δ).

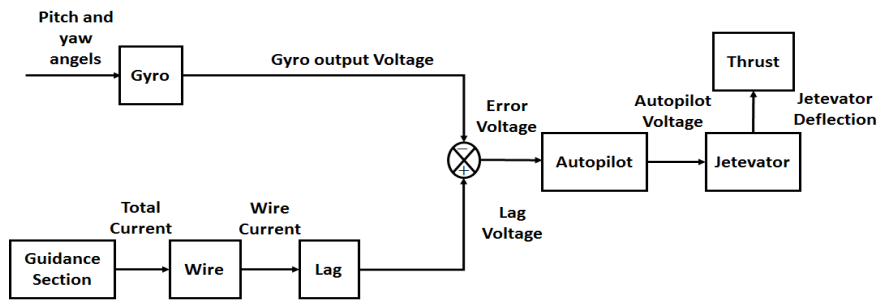


Fig. (4) Block diagram of control section

The implementation of the processor in the loop done by sequential steps. First one adding the desired autopilot in the nonlinear flight simulation 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 (5) [10-12]. 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. 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 (6).

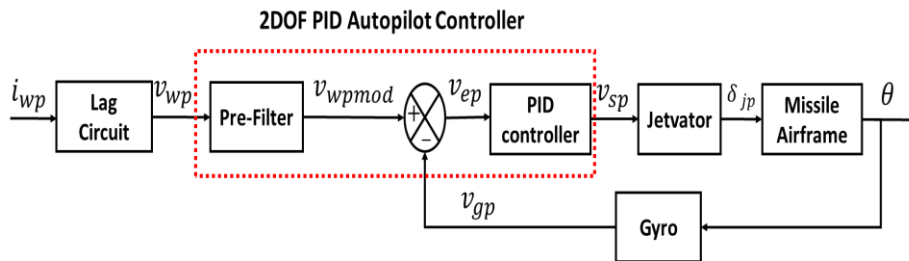


Fig. (5) 2DOF PID autopilot controller

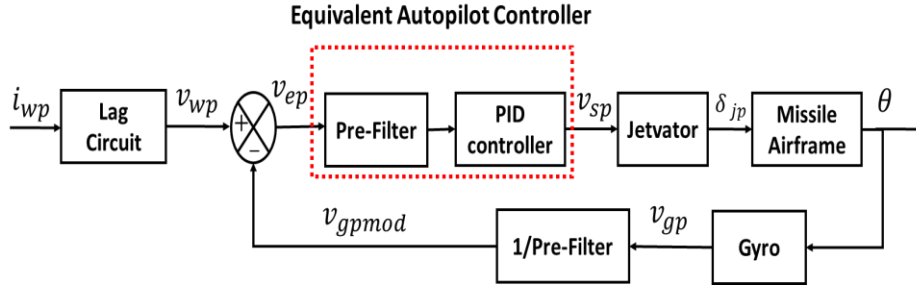


Fig. (6) Equivalent autopilot diagram

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

$$\frac{u(s)}{e(s)} = \frac{13244s + 89080}{946s + 4454} \quad (4)$$

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

5. Autopilot Analysis with Linearized Missile Airframe

In many control applications the plant can be considered linear within defined regions of operation, which may led to a set of linearized models at specific set points, or trim condition. 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. The present analysis was developed to insure the proposed autopilot achieve all system performance requirements especially in set point following and load disturbance rejection.

The designed controller analyses at the different set point weighting values and clarified the robustness of these designed controllers. The obtained results summarized in figure (7), which clarifies that the designed controllers has faster transient response than classical one. In addition, the set point weighting effect appeared as reduction the overshoot system and increasing the rising time of the system. However, the obtained controller with different set point weighting value have a lowest control effort at the steady state compared to classical controller as shown in figure (8).

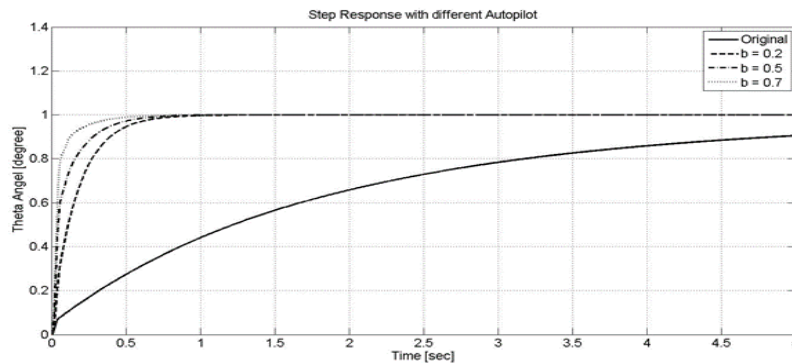


Fig. (7) Step response of the original and obtained controllers

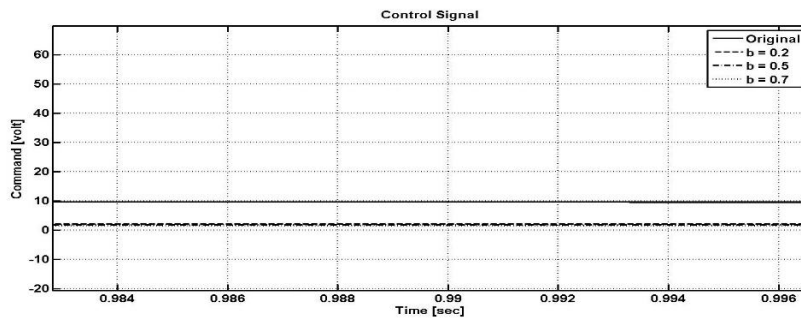


Fig. (8) Control signal of the original and obtained controllers

In addition, Applying white Gaussian noise to the gyro output, the step response with different set point weighting is shown in figure (9), 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 (10), which clarifies that the convergence using designed controller, is the best compare to classical controller as it rejects 50% within 0.1 sec and 95% within 0.25 sec. Also, the control effort shown in figure (11). Which reveals that this controller has the lowest control effort after applying disturbance.

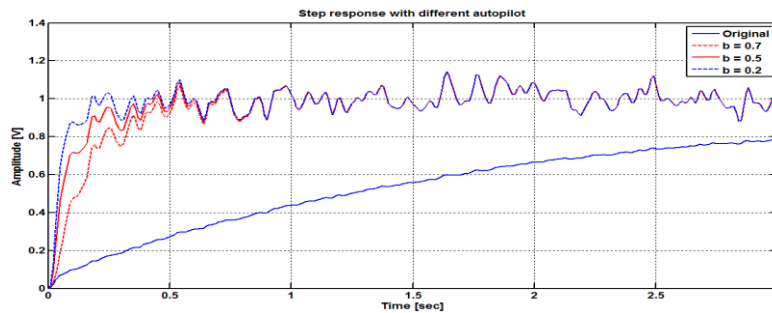


Fig. (9) Step response with applying measurement noise

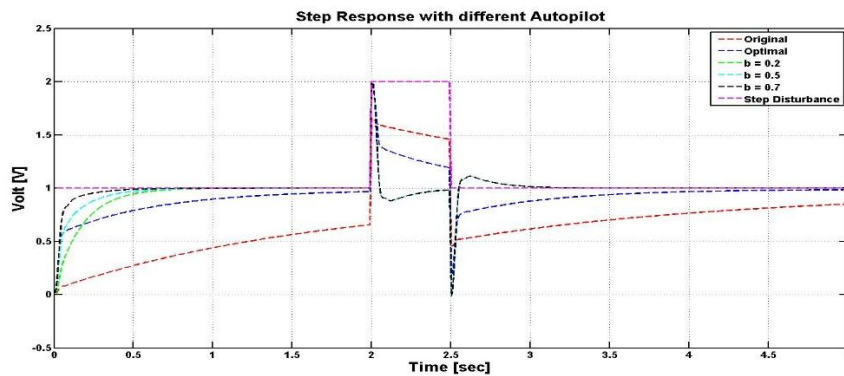


Fig. (10) Step response of the original and obtained controller with disturbance

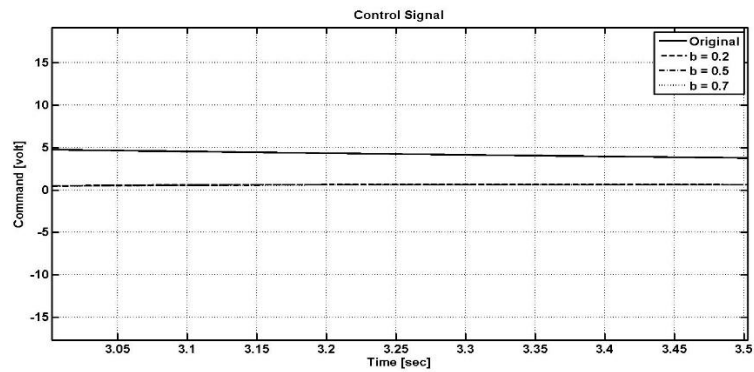


Fig. (11) Control signal of the original and obtained controller with disturbance

6. Autopilot Analysis with Non-linear Missile 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. 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 fixed target at different distance shown in figure (12) and (13) respectively. The miss-distance and variance of the control effort signal of the above evaluations summarized in table (1).

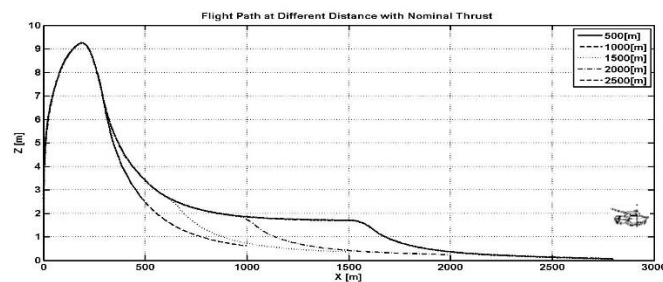


Fig. (12) Trajectory obtained with conventional autopilot

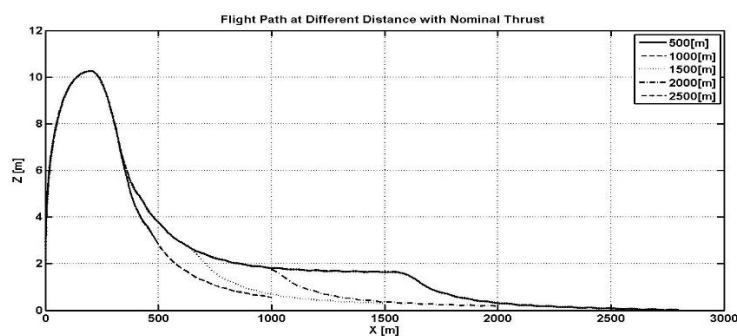


Fig. (13) Trajectory obtained with designed autopilot

Table (1) Designed autopilots evaluation via mis-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 at range to 80% of nominal thrust value, against perturbations in the aerodynamic coefficient of about $\pm 20\%$, 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 with accepted flight path and accepted minimum miss-distance.

7. Autopilot Implementation

The equivalent transfer function of autopilot is converted from s -domain (continuous time) to z -domain (discrete time) with specified 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 [13, 14].

$$\frac{u(z)}{e(z)} = \frac{A z - B}{C z - D} \quad (5)$$

Moreover, the more applicable one represented as

$$\frac{u(z)}{e(z)} = \frac{14 z - 11.26}{z - 0.863} \quad (6)$$

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

$$\begin{aligned} e(z) * (A - B z^{-1}) &= u(z) * (C - D z^{-1}) \\ A * e(z) - B * e(z) * z^{-1} &= C * u(z) - D * u(z) * z^{-1} \end{aligned} \quad (7)$$

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

$$u(n) = A * e(n) - C * e(n - 1) + D * u(n - 1) \quad (8)$$

8. Processor-in-Loop Experimental Test

The design of an autopilot requires a mathematic modeling followed by the adjusting of some model parameters. However, to overcome the autopilot to a single piece of hardware, involves the codification of this mathematical based model into an appropriate firmware description suited to work correctly in a specific platform. Recently a model driven development approach, firstly defined by system engineers, frequently used as way to reduce the time of

development of embedded systems, producing rapid and reliable product in a short time development cycle. This model driven approach used for test and is known as X in-Loop [4]. These tests provide four levels of testing configurations: MIL (Model-in-Loop), SIL (Software-in-Loop), PIL (Processor-in-Loop) and HIL (Hardware-in-Loop). Each of the configuration levels provides some advances, reduces the gap in the development process that initiate with the mathematical model, and ends at the firmware running in a stand-alone microprocessor platform. The following test consists in the PIL (Processor-in-Loop) that goes beyond the PC platform. This step introduces some hardware features that permit to achieve situations that are more realistic where the control algorithm will run. In PIL, the target processor is a non-real time environment and the communication with the external processors given by using specific functions installed in a simulation-integrated environment installed in the host PC. 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 main simulation environment connected to raspberry pi system with serial communication protocol. The designed 2DOF PID autopilot implemented on raspberry pi system as a processor-in-Loop part to evaluate the digital autopilot description on embedded hardware system. Experimental procedure carried out for system evaluation and validation as shown in figure (14).

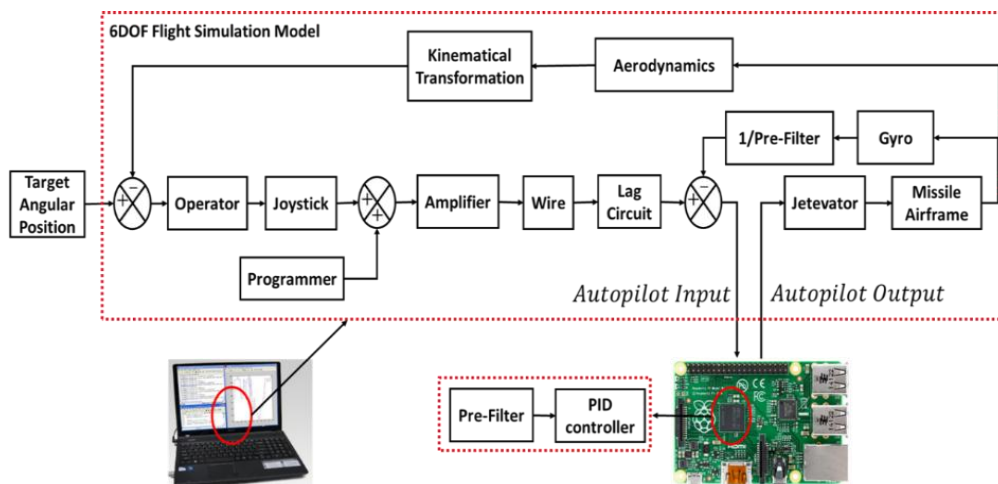


Fig. (14) Processor-in-loop experimental setup

The design autopilot evaluated with the flight path trajectory in pitch plane against simulated designed autopilot and PIL embedded plate form hardware at different target position. Figure (15) shows the flight path trajectory minimum and maximum tactical rang 500[m], and 2800[m] respectively. In addition, the error signal between the require position and actual position is shown in figure (16).

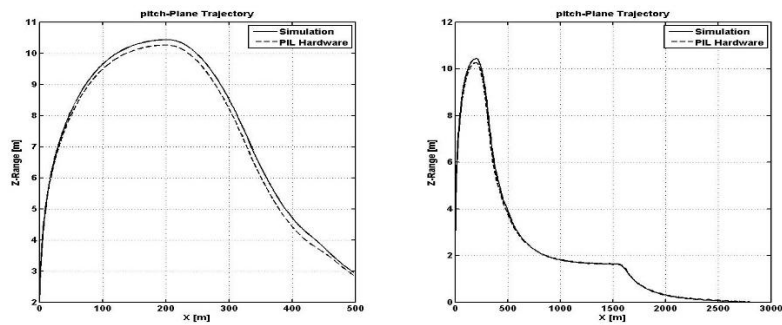


Fig. (15) Missile pitch plane trajectory

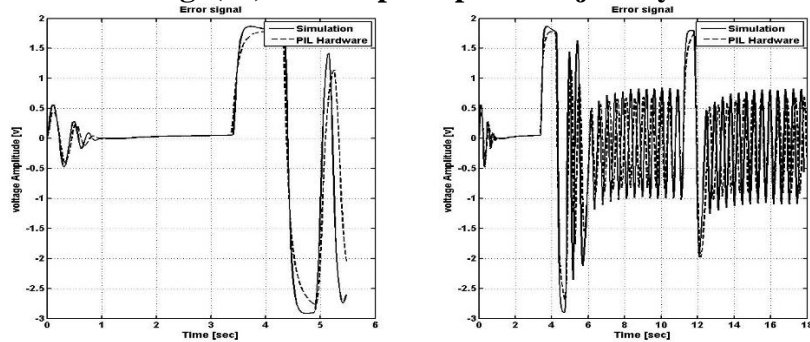


Fig. (16) Error signal

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

Table (2) Processor-in-loop experimental results

Experimental with Fixed Target	Target Distance [m]	Miss-Distance [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

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

9. Conclusion

2DOF PID autopilot is a suitable solution to overcome the demerits of traditional PID of achieve at the same time control system objectives. Sampling time and order of the mathematical representation of autopilot are main factors to have the same digital autopilot behavior like analog prototype. Raspberry pi system is a suitable plate form to increase experience to deal with system on chip plate form and to be used for hardware experimental test.

References

- [1] A. N. El-Din, "Performance Investigation of Adaptive Guidance Algorithms and its Effectiveness," PHD, Chair of Guidance, Military Technical College, Cairo, 2012.
- [2] K. J. Åström and T. Hägglund, "The future of PID control," *Control engineering practice*, vol. 9, pp. 1163-1175, 2001.
- [3] A. Visioli, *Practical PID control*: Springer Science & Business Media, 2006.
- [4] L. S. Martins-Filho, A. C. Santana, R. O. Duarte, and G. A. Junior, "Processor-in-the-Loop Simulations Applied to the Design and Evaluation of a Satellite Attitude Control," 2014.
- [5] D. Soares Pires and G. L. De Oliveira Serra, "Fuzzy digital PID controller design based on robust stability criteria," in *Industrial Informatics (INDIN), 2014 12th IEEE International Conference on*, 2014, pp. 654-659.
- [6] J. Manasa, J. Pramod, S. Jilani, and M. S. J. Hussain, "Real Time Object Counting using Raspberry pi."
- [7] S. Yamanoor and S. Yamanoor, *Raspberry Pi Mechatronics Projects HOTSHOT*: Packt Publishing Ltd, 2015.
- [8] P. Zarchan, "Tactical and strategic missile guidance," *Progress in astronautics and aeronautics*, 2002.
- [9] R. Vilanova and O. Arrieta, "PID design for improved disturbance attenuation: min max Sensitivity matching approach," *International Journal of Applied Mathematics*, vol. 37, pp. 1-6, 2007.
- [10] M. Araki and H. Taguchi, "Two-degree-of-freedom PID controllers," *International Journal of Control Automation and Systems*, vol. 1, pp. 401-411, 2003.
- [11] K. J. Åström, C. C. Hang, P. Persson, and W. K. Ho, "Towards intelligent PID control," *Automatica*, vol. 28, pp. 1-9, 1992.
- [12] M. Chidambaram, "Set point weighted PI/PID controllers," *Chemical Engineering Communications*, vol. 179, pp. 1-13, 2000.
- [13] M. A. Mazidi, J. G. Mazidi, and R. D. McKinlay, "The 8051 microcontroller and embedded systems," *New Delhi*, 2000.
- [14] B. G. A. Elaty, "Enhancement of On-board Electronic Package Using Modern Control Strategies," Master, Chair of Guidance, Military Technical College, Cairo, 2017.