Artificial Neural Network-based Control Architecture for Aircraft Autopilot Design

Shreehan Kate, National Institute of Technology-Tiruchirappalli, 620015, shreehan1912@gmail.com

***Abstract*— This research presents a unique method to design robust autopilot systems for Aircraft by using Artificial Neural Networks (ANN) as controllers. Through this study, we have investigated the capability of ANNs to control the behavior of a dynamic system or a process and to maintain a desired output or performance.**

The ANN can replicate the responses of the control system after it is trained on a dataset that consists of the inputs and output values of the PID controller. Upon replacing the PID controller in the closed-loop feedback control system with the trained ANN, the ANN can adapt to the aircraft's non-linear dynamics. The results obtained in this study suggest that the ANN-based controller can be a viable alternative to conventional PID controllers.

I. INTRODUCTION

Autopilot design for aircraft is a critical aspect of aerospace engineering. It involves developing a robust system for aircraft followed by evaluating the performance and safety of the system before its deployment. The design process involves extensive usage of existing control methodologies and algorithms that can manage the complexities of aircraft dynamics. One advanced method used in this field is the Linear-Quadratic Regulator (LQR) and Pole Placement approach [1], which offers a systematic way to design non-linear feedback controllers. Traditional controllers such as the Proportional-Integral-Derivative (PID) and Proportional-Derivative (PD) controllers have long been utilized for their simplicity in regulating non-linear behavior to an extent [2]. However, as technology advances and the need for more sophisticated control strategies grows, researchers have turned to Artificial Intelligence-based control schemes such as Fuzzy Logic-based control, Model Predictive Control (MPC), Reinforcement Learning, and Genetic Algorithms. Artificial Neural Networks were used in developing the Intelligent Autopilot system (IAS), a fully autonomous control strategy where large airliners were piloted by ANNs trained on the experiences of human pilots [9].
These control strategies come under the category of Intelligent Control. In literature about the LQR approach, Intelligent Control proves to be dependable in choosing Q and R weighing matrices for optimization purposes [3].
This research paper makes use of the state-space equations of Aircraft autopilot that are described in this research [2] and aims to compare the performances of PID controllers and Artificial Neural Networks (ANNs) as controllers for controlling the pitch angle of the Aircraft autopilot. ANNs are modeled after the structure of the human brain, comprising interconnected nodes known as artificial neurons. They can adapt and learn from complex, non-linear relationships, delivering better performance compared to traditional Controllers.

The upcoming section i.e., Section 2 discusses the non-linear coupled differential equations and the subsequent formulation of the state-space model of a Boeing commercial aircraft used in this paper.

II. AUTOPILOT MODEL

An aircraft is modeled by six complicated non-linear coupled differential equations [2].
From the aircraft model given in [2], the following velocity vector of the aircraft is obtained:

$$V = \begin{bmatrix} U \\ V \\ W \\ P \\ Q \\ R \end{bmatrix} \tag{1}$$

where,

U is the longitudinal forward velocity vector
V is the lateral velocity
W is the vertical velocity
P is the roll rate
Q is the pitch rate
R is the yaw rate

Similarly, we define the forces, and the moments as follows [2].

$$F = \begin{bmatrix} X \\ Y \\ Z \\ L \\ M \\ N \end{bmatrix} \tag{2}$$

where,

X is the longitudinal force
Y is the transverse force
Z is the vertical force
L is the roll moment
M is the pitch moment and N is the yaw moment

An aircraft is a system with six degrees of freedom. It can move forward, sideways, down, and rotate about the axes with yaw, pitch, and roll. Developing the state-space model in this case requires the values of all six variables and it further necessitates the decoupling and linearization of the equations of motions into longitudinal and lateral equations as mentioned in [2]. The assumptions made while formulating the state-space model required in this study are as follows:

i. Aircraft must cruise at constant altitude and velocity. Thrust, drag, weight, and lift forces must be balanced in the X and Y direction [2], [4].
ii. A change in pitch angle will not change the speed of the aircraft under any circumstances [2], [8].
iii. The input will be elevation differential (δ) and the output will be the pitch angle (θ_p), Angle of Attack (α), and Pitch Rate (q) of the aircraft [2], [4].

We will keep the assumptions mentioned in mind as we establish the following parameters as state variables for the state-space model [2]. They are listed in the table below.

Table 1: State variables used in the state-space model

State Variable	Name
α	Angle of Attack
q	Pitch Rate
θ_p	Pitch Angle

The input will be the elevator deflection (δ) and the output will be the pitch angle θ_p, pitch rate q, Angle of Attack α. Differential equations obtained for the aircraft (Boeing) are given below [2]:

$$\dot{\alpha} = 0.313\alpha - 56.7q + 0.232\delta \quad (3)$$

$$\dot{q} = 0.0319\alpha - 0.426q + 0.0203\delta \quad (4)$$

$$\dot{\theta}p = 56.7q \quad (5)$$

The standard form of state-space equations is as follows

$$\dot{x} = Ax + Bu \quad (6)$$

$$y = Cx + Du \quad (7)$$

where,

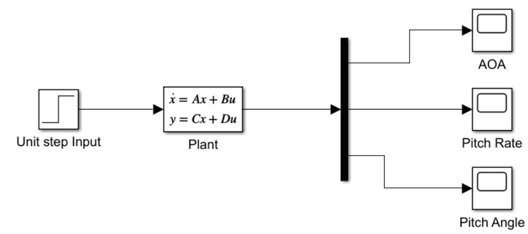
$$A = \begin{bmatrix} -0.313 & 56.7 & 0 \\ -0.0139 & -0.426 & 0 \\ 0 & 56.7 & 0 \end{bmatrix}; B = \begin{bmatrix} 0.232 \\ 0.0203 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad D = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (8)$$

II.B. Building the state-space model

In this section, the development of the state-space model on MATLAB and Simulink software will be discussed.
In Simulink, the State-Space block is used and the matrices A, B, C, and D as described in equation (8) are entered. The initial state is set to zero and state names are given as per Table 1.
The parameters of the state-space block are set.
A control system is developed by taking the output of the pitch angle as feedback. The block diagram of the control loop is given in the image below,

Image 1: Open loop block diagram of the aircraft model



To observe the behavior of the open loop system, the system is triggered with a unit step input, and the following responses are obtained.

Image 2: Open loop response of Angle of Attack

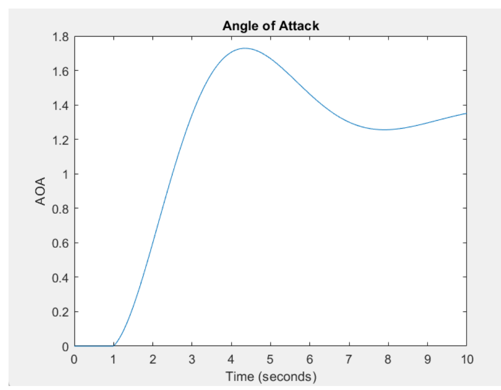


Image 3: Open loop response of Pitch Rate

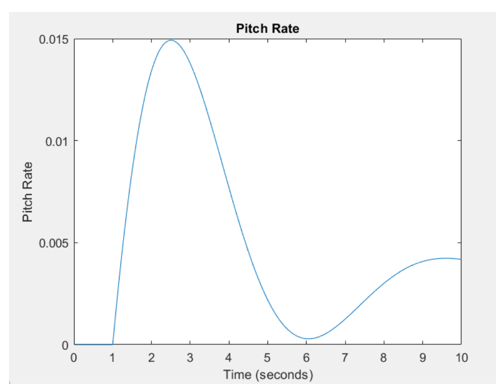
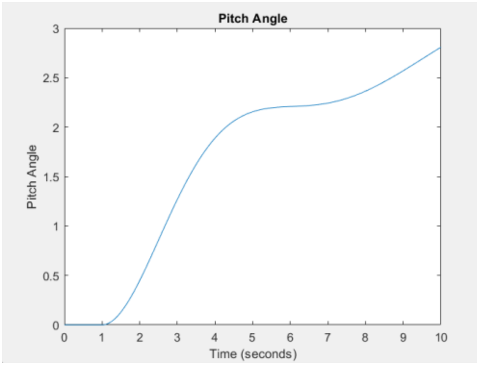


Image 4: Open loop response of Pitch Angle



Upon analyzing images 2, 3, and 4, it has been observed that the open loop responses exhibit unstable behavior for the given parameters. To effectively control these parameters, the development of a suitable controller becomes crucial. The subsequent section will delve into a detailed discussion of ANN-based controller development, along with thorough analysis and simulations.

III. CONTROLLER DEVELOPMENT

Observing the system's closed-loop performance (without controller), the output of the pitch angle is taken as the negative feedback, and it is observed at 31.6 seconds, the pitch angle settles down. To control the settling time and overshoot of the system, a Proportional-Integral-Derivative (PID) controller is used. The tuning of the PID controller is carried out with the help of the PID tuner app built within the Simulink environment [5].

III.A Proposed Method

The proposed method involves the development of the neural network architecture for the linearized system developed in Section II

To increase the control accuracy and flexibility, an Artificial Neural Network-based controller is used. Such types of controllers are flexible and are capable of handling non-linearity in a much better way. An artificial Neural Network controls the process by mimicking the working principle of a human brain.

The basic structure of an Artificial Neural Network consists of an input layer, an output layer, and hidden layers. The primary function of each layer is to receive data from the outside world, learn and analyze the data transmitted, and finally process the findings or inferences into valuable information for the output layer.

Below is the development workflow of the Artificial Neural Network-based controller [6].

i. Generate a dataset comprising input-output parameters of the PID controller
ii. The input values comprise the values of the error signal and the derivative of the error signal at each instant.
iii. The input values are taken to be the predictor and the output values of the PID controller are taken to be the response values.
iv. The neural network has been trained, validated, and tested using the provided dataset. The Mean Square Error will be used to assess the performance of the Artificial Neural Network.

Training and performance evaluation of the Neural Network will be discussed in the next section.

III.B Training the neural network

The neural network fitting app present in MATLAB will be used in the development of the neural network architecture along with its training, testing, and validation. The neural network will be trained on a predictor set containing a given number of observations with two (error and error derivative) features and a response set containing one (PID coefficient). The steps to implement the neural network in the Neural Network fitting app in MATLAB are as follows:

- The dataset comprising the values of the error signal, the derivative of the error signal, and the output values of the PID controller are gathered and sent to the MATLAB workspace.
- The Neural Network fitting application is invoked by entering the command “nftool” on the command window of the MATLAB workspace.
- The dataset is imported into the app and the entire dataset is divided into training, validation, and testing data.
- The neural network used is a two-layer feedforward network with sigmoid hidden neurons and linear output neurons.

Table 2: Parameters used in ANN:

Parameters	Value
Epochs	1000
Number of hidden layers	1
Number of neurons in the hidden layer	10
Activation function	Sigmoid
Error	Mean Square Error
Coefficient of determination	R-squared
Optimization	Levenberg Marquardt

The Levenberg Marquardt (LM) optimization technique is used to train artificial neural networks. Initially, the LM algorithm was developed to solve non-linear equations. LM algorithm combines two minimization techniques: the Gradient Descent method and the Gauss-Newton Method. The LM algorithm has a high computation speed and can be used with small datasets [7].

IV. SIMULATIONS AND RESULTS

The Artificial Neural Network required for the controller is trained successfully and is ready for deployment. A few changes need to be made in the existing control system to incorporate the ANN-based controller. In this section, the results obtained from the control loop containing the PID controller as well as the ANN-based controller will be discussed, and inferences will be made.

Image 5: Aircraft Autopilot with ANN controller.

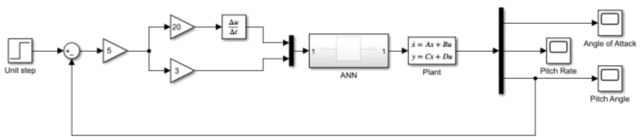


Table 3: Performance of ANN

	Observations	MSE	R
Training	1945	1.0369	0.99689
Validation	417	0.1476	0.9996
Testing	417	0.3451	0.99986

Regression plots are an effective way of understanding the neural network's performance. The Regression (R) value suggests how close the predicted value is to the actual value. Mean-square error is defined as mean squared differences between outputs and responses.

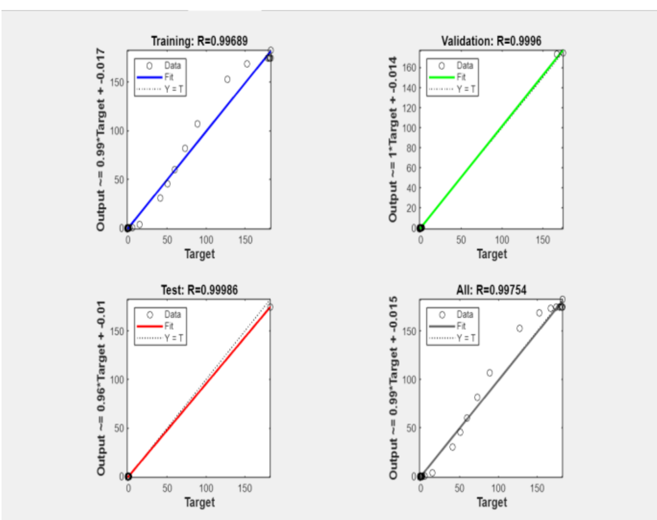


Image 6: Regression Plots for Training, Testing and Validation

Once the system with the ANN-based controller is set up, it is triggered with a unit step command. The waveforms for Angle of Attack, Pitch angle, and Pitch Rate are obtained and compared with those obtained with PID controllers.

Image 5: Waveform of Pitch Angle (PID and ANN)

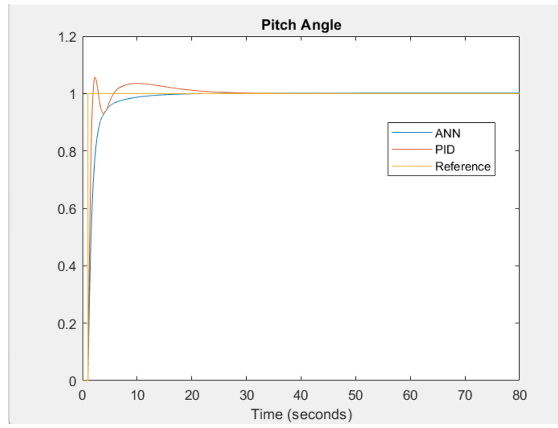


Image 6: Pitch Rate response (PID and ANN)

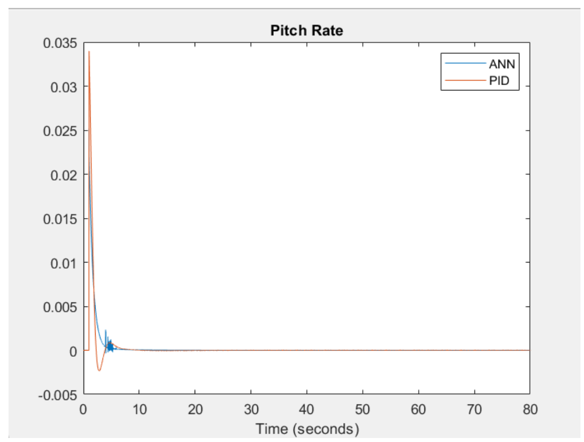
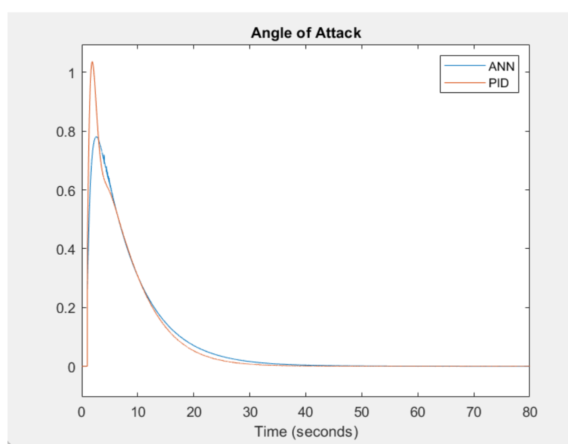


Image 7: Angle of Attack response (PID and ANN)



Time domain specifications for the responses in the case of PID controller and ANN-based controller are recorded and tabulated below (based on Image 5):

Table 4: Time-domain specifications

	PID	ANN
Rise Time (s)	0.6462	2.1526
Peak overshoot (%)	5.6912	0
Settling time (s)	16.4210	8.2448
Peak Time (s)	2.2918	80
Peak	1.0569	1.0015

V. CONCLUSION AND FUTURE SCOPE

This paper explores the advantages of an Artificial-based Neural Network control architecture for Aircraft Autopilot design. Neural Networks have been long used in the development of autonomous systems.
One such example is the use of convolutional neural networks in self-driving cars [10].
Through this paper, it has successfully been demonstrated that intelligent control techniques are a suitable replacement for traditional controllers. The ANN controller offers robustness and adaptability to non-linear dynamics. Further optimization can improve the controller's performance when it comes to the waveform of the Angle of Attack. Future research and improvement in autopilot design can include the use of intelligent control techniques such as model predictive control (MPC), sliding mode control (SMC), or adaptive control techniques, and machine learning techniques, such as deep learning or reinforcement learning, to optimize autopilot control based on real-time feedback and improve performance and adaptability. These avenues of research aim to enhance autopilot systems, making them safer and more efficient in various control applications.[6]

VI. ACKNOWLEDGEMENTS

MATLAB and Simulink were extensively used for conducting simulations and for the generation of results.
Gemini, a large language model powered by Google AI was used to improve the structure of the texts.
Grammarly, a writing assistant was used to improve the readability of the manuscript.

VII. REFERENCES

[1] V.S.N. Murthy Arikapalli, Shiladitya Bhowmick, P.V.R.R. Bhogendra Rao, and Ramakalyan Ayyagari, “Missile Longitudinal Dynamics Control Design using Pole Placement and LQR Methods – A Critical Analysis in Defence Science Journal, Vol. 71, No. 5, September 2021, pp. 5-8.

[2] Lubna Moin, Aman-uz-Zaman Baig, Vali Uddin, “State Space Model of an Aircraft Using Simulink” in International Journal of System Modeling and Simulation.

[3] Sanjay Joseph Chacko a, Neeraj P.C. b, Rajesh Joseph Abraham, “Optimizing LQR controllers: A comparative study” in Results in Control and Optimization. pp. 7-9.

[4] Ju Jiang; Mohamed S. Kamel, “Pitch Control of an Aircraft with Aggregated Reinforcement Learning Algorithms” in 2007 International Joint Conference on Neural Networks.

[5] Htet Soe Paing, Schagin Anatolii, Zaw Myo Naing, Han Myo Htun, “Designing, Simulation and Control of Autopilot using PID Controller” in 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus). pp. 3-4.

[6] Pritish Sharma, Dr. Jyoti Ohri, “ANFIS Based PID Control of Antilock Braking System Model” in 2023 7th International Conference on Computer Applications in Electrical Engineering-Recent Advances (CERA). pp. 3-4, 6

[7] Everton Gomede, PhD “Optimizing Neural Networks with Levenberg-Marquardt: An Effective Approach for Small Datasets (Updated)” published in The Modern Scientist, Medium.

[8] Shuguo Liang, Xiaoping Chen, Bo Zhu, “A Nonovershooting Pitch-Angle Regulator for Aircraft with Conventional Aerodynamic Configuration”, 2012 Fifth International Conference on Intelligent Computation Technology and Automation, transaction 2012 IEEE.

[9] Haitham Baomar & Peter J. Bentley, “Autonomous flight cycles and extreme landings of airliners beyond the current limits and capabilities using artificial neural networks”, Applied Intelligence (2021), pp 1-2, 5-8.

[10] Syed OwaisAli Chishti; Sana Riaz; Muhammad Bilal, Zaib; Mohammad Nauman, “Self-Driving Cars Using CNN and Q-Learning”, “2018 IEEE 21st International Multi-Topic Conference (INMIC)”.