An Intelligent Autopilot System Using Artificial Neural Network

Ashwini Khandal (Student)[1], Mr. Atul Kumar Verma (Asst. Prof)[2]
[1, 2] Dept of Computer Science
[1, 2] Poornima College of Engineering, Jaipur, India

Abstract- *An Intelligent Autopilot System (IAS) is the one that can learn piloting skills by observing and imitating expert human pilots. IAS is a potential solution to the current problem of Automatic Flight Control Systems of being unable to handle flight uncertainties and the need to construct control models manually. A robust Learning by Imitation approach uses human pilots to demonstrate the task to be learned in a flight simulator while training datasets are captured from these demonstrations. The datasets are then used by Artificial Neural Networks to generate control models automatically. The control models imitate the skills of the human pilot when performing piloting tasks including handling flight uncertainties such as severe weather conditions and flight emergencies such as engine(s) failure or fire, Rejected Take Off (RTO), and emergency landing, while a flight manager program decides which ANNs to be fired given the current condition.*

Keywords- Intelligent Autopilot System (IAS), Artificial Neural Networks(ANNs), Automatic Flight Control Systems (AFCS).

I. INTRODUCTION

There are numerous crisis circumstances and flight vulnerabilities, for example, extreme climate conditions or framework disappointment which may emerge amid various periods of the flight. All the human pilots are all around prepared to take legitimate activities and handle these circumstances. Interestingly, Automatic Flight Control Systems (AFCS/Autopilot) are profoundly restricted, equipped for performing negligible steering undertakings in non-crisis conditions. For Example, solid turbulence can make the autopilot withdraw or even endeavor an undesired activity which could influence flight wellbeing. The autopilots require consistent checking of the framework and the flight status by the flight team to respond rapidly to any undesired circumstance or crises. Then again, endeavoring to discover everything that could turn out badly with a flight, and fusing that into the arrangement of principles or control models in an AFCS is infeasible. There have been reports either discussing the limitations of current autopilots [3] [4] such as the inability to handle severe weather conditions, or blaming autopilots for a number of aviation catastrophes.

This work plans to address this issue by examining an Intelligent Autopilot System (IAS) that can gain from human pilots by the Learning by Imitation idea with Artificial Neural Networks. By utilizing this approach we expect to broaden the abilities of present day autopilots and empower them to independently respond to various situations from typical to crisis circumstances.

II. BACKGROUND

A. Automatic Flight Control Systems

Currently used autopilots uses the Control Theory. Modern autopilots are based on controllers such as the Proportional Integral Derivative (PID) controller, and Finite-State automation [5]. Many recent research efforts focus on enhancing flight controllers, through the introduction of various methods such as a non- adaptive Backstepping approach [6], Dynamical Inversion flight control approach based on Artificial Neural Network Disturbance Observer to handle the dynamical inversion error factor [7], an L1 adaptive controller which is based on piecewise constant adaptive laws [8], a multi-layered hybrid linear/non-linear controller for biologically inspired Unmanned Aerial Vehicles [9], and a fault-tolerant control based on Gain-Scheduled PID [10].

In any case, physically planning and building up all the essential controllers to deal with the entire range of flight situations and vulnerabilities going from ordinary to crisis circumstances won't not be the perfect strategy because of attainability limitations, for example, the trouble in covering every single conceivable consequence.

B. *Fault/Failure Tolerant Systems for Flight Control*

Many recent research efforts focus on enhancing flight controllers by adding fault/failure tolerant capabilities. With respect to flight control systems, a fault is “an unpermitted deviation of at least one characteristic property of the system from the acceptable, usual, standard condition.”

[11], while failure is "a permanent interruption of a system's ability to perform a required function under specified operating conditions." [11].

To handle faults and disappointments, late research endeavors have been concentrating on planning Fault Detection and Diagnosis (FDD) systems that can either stream data to ground team individuals particularly on account of UAVs, or encourage fault tolerant systems that are fit for handling system faults.

The primary kind of such systems are known as the Passive Fault Tolerant Controllers which can handle direct faults, for example, parameters deviations by utilizing a hearty criticism controller. Nonetheless, if the faults are past the abilities of such controllers, another sort of fault tolerant systems turns into a need. This write is known as an Active Fault Tolerant control system which incorporates a different FDD system that includes an expanded and upgraded level of fault resilience capacities [12]. In the event of crisis circumstances, primarily motor disappointment, motor fire, flight instruments disappointment, or control surface harm or disappointment, proceeding to fly turns out to be either incomprehensible or can represents a genuine risk to the security of the flight. In such conditions, a constrained or crisis landing on a reasonable surface, for example, a level field turns into an absolute necessity particularly on the off chance that it isn't conceivable to return securely to the runway [13].

C. Artificial Neural Networks

In gadgets building and related fields, artificial neural networks (ANNs) are scientific or computational models that are motivated by a human's focal sensory system (specifically the mind) which is equipped for machine learning and also design acknowledgment. Artificial neural networks are by and large introduced as systems of exceedingly interconnected "neurons" which can process esteems from inputs. With the assistance of these interconnected neurons all the parallel preparing is being done in body and the best case of Parallel Processing is human or creature's body. Numerical examination has tackled a portion of the puzzles postured by the new models however has left numerous inquiries for future examinations. There is no compelling reason to state, the investigation of neurons, their interconnections, and their part as the mind's rudimentary building squares is a standout amongst the most unique and vital research fields in current universe of gadgets and software engineering.

III. THE INTELLIGENT AUTOPILOT SYSTEM

The Intelligent Autopilot System (IAS) in this paper can be seen as a disciple that watches the exhibit of another errand by the accomplished educator, and then plays out a similar undertaking autonomously. An effective speculation of Learning by Imitation should mull over the catching of low-level models and abnormal state models, which can be seen as fast and dynamic sub-activities that happen in divisions of a moment, and activities overseeing the entire procedure and how it ought to be performed deliberately. It is imperative to catch and impersonate the two levels keeping in mind the end goal to handle flight vulnerabilities effectively. The IAS is made of the accompanying parts: a pilot training program, an interface, a database, a flight chief program, and Artificial Neural Networks. The IAS execution technique has three stages: A. Pilot Data Collection, B. Training, and C. Autonomous Control. In each step, different IAS components are used.

A. Pilot Data Collection

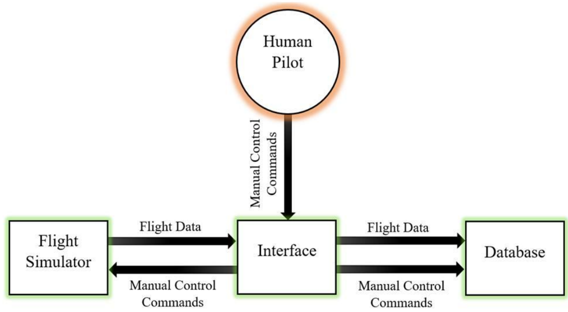

Fig 1: Block diagram illustrating the IAS components used during the pilot data collection step.

1). Flight Simulator

Prior to the IAS can be prepared or can take control, we should gather data from a pilot. This is performed utilizing X-Plane which is a propelled pilot test program that has been utilized as the test system of decision in numerous examination papers, for example, [15] [16] [17].

X-Plane is utilized by different associations and ventures, for example, NASA, Boeing, Cirrus, Cessna, Piper, Precession Flight Controls Incorporated, Japan Airlines, and the American Federal Aviation Administration. 1 X-Plane can speak with outside applications by sending and accepting flight status and control commands data over a system through User Datagram Protocol (UDP) parcels. For this work, the test system is set up to send and get parcels involving wanted data each 0.1 second. In X-Plane, it is conceivable to reenact various flight crises to train pilots. Crises extend from extreme climate conditions to system disappointment, for example, motor disappointment or fire.

2). The IAS Interface

The IAS Interface is in charge of data stream between the pilot training program and the system in the two bearings. The Interface contains control command catches that give a streamlined yet adequate flying machine control interface which can be utilized to perform fundamental errands of piloting an air ship, for example, take-off and landing in the test system while having the capacity to control different systems, for example, fuel and fire systems. It likewise shows flight data got from the test system.

Data collection is begun promptly before exhibit, at that point; the pilot utilizes the Interface to play out the piloting assignment to be educated. The Interface gathers flight data from X-Plane over the system utilizing UDP bundles, and gathers the pilot's activities while playing out the undertaking, which are additionally sent back to the test system as manual control commands. The Interface sorts out the gathered flight data got from the test system (inputs), and the pilot's activities (yields) into vectors of information sources and yields, which are sent to the database each 1 second.

3). Database

A SQL Server database stores all data caught from the pilot demonstrator and X-Plane, which are gotten from the Interface. The database contains tables intended to store: 1. Flight data as sources of info, and 2. Pilot's activities as yields. These tables are then utilized as training datasets to prepare the Artificial Neural Networks of the IAS.

B. Training

After the human pilot data collection step is completed, Artificial Neural Networks are used to generate learning models from the captured datasets through offline training.

Ten feedforward Artificial Neural Networks comprise the core of the IAS. Each ANN is designed and trained to handle specific controls and tasks. The ANNs are: Taxi Speed Gain ANN, Take Off ANN, Rejected Take Off ANN, Aileron ANN, Rudder ANN, Cruise Altitude ANN, Cruise Pitch ANN, Fire Situation ANN, Emergency Landing Pitch ANN, and Emergency Landing Altitude ANN. The inputs and outputs which represent the gathered data and relevant actions, and the topologies of the ten ANNs are illustrated in Fig. 3.

Before training, the datasets are normalized, and retrieved from the database. Then, the datasets are fed to the ANNs. Next, Sigmoid (1) [18] and Hyperbolic Tangent (Tanh) (2) [18] functions are applied for the neuron activation step, where f(x) is the activation function for each neuron, and x is the relevant input value:

$$f(x) = \frac{1}{1+e^{-x}} \tag{1}$$

$$f(x) = \frac{e^{2x}-1}{e^{2x}+1} \tag{2}$$

The Sigmoid activation function (1) is used by the Taxi Speed Gain ANN, Take Off ANN, Emergency Landing Altitude ANN, Rejected Take Off ANN, and the Fire Situation ANN, while (2) is used by the rest since their datasets contain negative values.

Next, Back propagation is applied. Based on the activation function, (3) [19], or (4) [19] are applied to calculate the error signal(δ) where t_n is the desired target value and a_n is the actual activation value:

$$\delta_n = (t_n - a_n)a_n(1 - a_n) \tag{3}$$

$$\delta_n = (t_n - a_n)(1 - a_n)(1 + a_n) \tag{4}$$

Finally, coefficients of models (weights and biases) are updated using (5) [20] where $\delta w_{i,j}$ is the change in the weight between nodes j and k.

$$w_{i,j} = w_{i,j} + \delta w_{i,j} \tag{5}$$

When training is completed, the learning models are generated, and the free parameters or coefficients represented by weights and biases of the models are stored in the database.

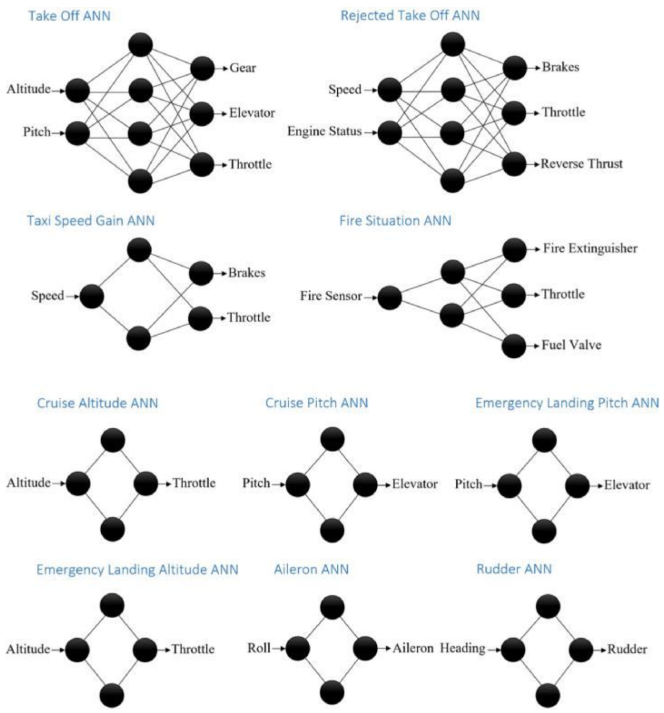

Fig 2: Inputs, outputs, and the topologies of the ten ANNs representing the core of the Intelligent Autopilot System. Each ANN is designed and trained to handle a specific task.

C. Autonomous Control

Once prepared, the IAS would now be able to be utilized for autonomous control. Fig.3 delineates the parts utilized amid the autonomous control step.

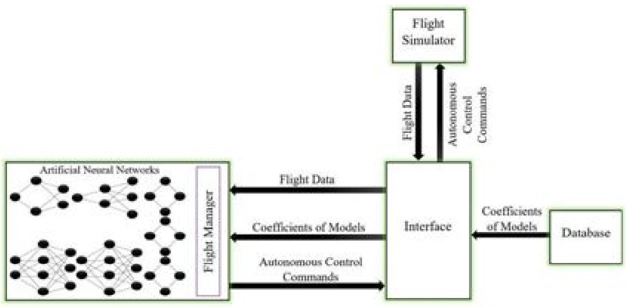

Fig 3: Block diagram illustrating the IAS components used during autonomous control.

1). The IAS Interface

Here, the Interface recovers the coefficients of the models from the database for each prepared ANN, and gets flight data from the pilot training program each 0.1 second. The Interface sorts out the coefficients into sets of weights and biases, and arranges data got from the test system into sets of contributions for each ANN. The applicable coefficients, and flight data input sets are then encouraged to the Flight Manager and the ANNs of the IAS to deliver yields. The yields of the ANNs are sent to the Interface which sends them to the pilot test program as autonomous control commands utilizing UDP parcels each 0.1 second.

2). The Flight Manager Program

The reason for the Flight Manager is to deal with the ten ANNs of the IAS by choosing which ANNs are to be utilized all the while at every minute. The Flight Manager begins by accepting flight data from the pilot training program through the interface of the IAS, at that point it identifies the flight condition and stage by looking at the got flight data, and chooses which ANNs are required to be utilized given the flight condition (typical/crisis/fire circumstance) and stage (taxi speed pick up/take off/journey/crisis landing).

3). Artificial Neural Networks

The pertinent arrangement of flight data inputs got through the Interface is utilized by the ANNs' information neurons alongside the applicable coefficients to foresee control commands given the flight status by applying (1) and (2). The estimations of the yield layers are sent to the Interface which sends them to the pilot test program as autonomous control commands. Taxi Speed Gain ANN is utilized while on the runway just before take off to foresee the appropriate brakes and throttle command esteems. Take Off ANN is utilized after a specific take off speed is accomplished to anticipate rigging, lift, and throttle command esteems. Rejected Take Off ANN is utilized to prematurely end take off if essential by foreseeing brakes, throttle, and turn around throttle command esteems. Aileron ANN is utilized to control the air ship's roll quickly after take off. Rudder ANN is utilized to control the airplane's taking before take off, and yaw when airborne on the off chance that one motor comes up short and makes drag. Journey Altitude ANN is utilized to control the flying machine's coveted cruising elevation by anticipating the throttle command esteem. Journey Pitch ANN controls the pitch while cruising by anticipating the lift command esteem. Fire Situation ANN is utilized as a part of instance of flame by anticipating fuel valve and fire quenching control commands. Crisis Landing Pitch ANN keeps up a specific pitch amid crisis landing to lose speed without slowing down and to keep a nose first crash. Crisis Landing Altitude ANN controls the throttle if there should arise an occurrence of a solitary motor disappointment.

IV. EXPERIMENTS AND RESULTS

In research works [1] and [2] various experiments have been conducted to see if the discussed IAS model was capable of imitating the human pilot's actions and behaviour. The experiments which were performed are: autonomous taxi speed gain, take off, climb, and applying rudder and aileron to correct heading and roll deviations under normal and severe weather conditions and Rejecting take off, Emergency landing, Maintaining a cruising altitude, and Handling single engine failure/fire while airborne.

Each experiment was composed of 20 attempts by the IAS to perform autonomously under the given conditions.

The experiments exhibited extremely alluring outcomes. The IAS was equipped for impersonating the human pilot's activities and conduct with striking precision, and solid consistency. They delineate the capacity of the IAS to perform superior to the human pilot educator due to the accomplished solid match of the learning models. . The IAS was fit for utilizing the effectively learned models to keep flying while bit by bit losing height. Despite the fact that the air ship's standard autopilot kept up a superior elevation for the time being, by forcefully expanding motor push it improves the probability of motor disappointment in the rest of the motor, with possibly disastrous outcomes.

The system could impersonate numerous human pilot's abilities and conduct subsequent to being given exceptionally constrained illustrations. This is because of the approach of dividing the issue of autonomous piloting while at the same time handling vulnerabilities into little squares of undertakings, and doling out various ANNs uniquely outlined and prepared for each errand, which brought about the age of very precise models.

V. CONCLUSION & FUTURE WORK

In this work, a robust approach is discussed to "teach" autopilots how to handle uncertainties and emergencies with minimum effort by exploiting Learning by Imitation also known as Learning from Demonstration. The experiments showed the ability of the IAS to capture high-level tasks such as coordinating the necessary actions to reject take off and extinguish fire.

Breaking down the piloting tasks, and adding more Artificial Neural Networks enhanced performance and accuracy, and allowed the coverage of a wider spectrum of tasks.

The aviation industry is currently working on solutions which should lead to decreasing the dependence on crew members. The reason behind this is to lower workload, human error, stress, and emergency situations where the captain or the first officer becomes incapable, by developing autopilots capable of handling multiple scenarios without human intervention. We anticipate that future Autopilot systems which make of methods proposed here could improve safety and save lives.

Future effort will focus on giving the IAS the ability to learn how to fly a pre-selected course, and land safely in an airport. The IAS should be capable of avoiding no-fly zones that are either pre-identified, or detected during the flight such as severe weather systems detected by the aircraft's radar.

The Flight Manager program should be redesigned to utilize Artificial Neural Networks to classify the situation (normal or emergency), and predict the suitable flight control law or mode given the situation.

The problem of sensor fault and denial should be investigated to test the feasibility of teaching the IAS how to handle such scenarios.

REFERENCES

[1] H. Baomar and P. J. Bentley, "An Intelligent Autopilot System that learns piloting skills from human pilots by imitation," 2016 International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA,USA,2016

[2] H. Baomar and P. J. Bentley, " An Intelligent Autopilot System that Learns Flight Emergency Procedures by Imitating Human Pilots".

[3] Baumbach, J., Marais, A. and Gonçalves, D. (2015). Losing the Boxes:Fragmentation as a Source of System Complexity. Proc. of the 11th SA INCOSE Conference.

[4] Sherry, L., Mauro, R. (2014). Controlled Flight into Stall (CFIS): Functional complexity failures and automation surprises. Integrated Communications, Navigation and Surveillance Conference (ICNS), vol., no., pp.D1-1-D1-11.

[5] Nelson, Robert C. Flight stability and automatic control. Vol. 2. WCB/McGraw Hill, 1998.

[6] Sartori, D., Quagliotti, F., Rutherford, M. and Valavanis, K. (2014). Implementation and Testing of a Backstepping Controller Autopilot for Fixed-wing UAVs. Journal of Intelligent & Robotic Systems, 76(3-4), pp.505-525.

[7] Chen, M., Jiang, C. and Wu, Q. (2011). Disturbance-Observer-Based Robust Flight Control for Hypersonic Vehicles Using Neural Networks. adv sci lett, 4(4), pp.1771-1775.

[8] Capello, E., Guglieri, G., Quagliotti, F. and Sartori, D. (2012). Design and Validation of an ${\mathcal{L}}_{1}$ Adaptive Controller for Mini-UAV Autopilot. Journal of Intelligent & Robotic Systems, 69(1- 4), pp.109-118.
[9] Ratti, J. and Vachtsevanos, G. (2011). Inventing a Biologically Inspired, Energy Efficient Micro Aerial Vehicle. Journal of Intelligent & Robotic Systems, 65(1-4), pp.437-455.
[10] Sadeghzadeh, I. and Zhang Y. (2013). Actuator fault-tolerant control based on Gain-Scheduled PID with application to fixed-wing Unmanned Aerial Vehicle. Control and Fault-Tolerant Systems (SysTol), pp.342-346.
[11] G. J. J. Ducard, “Fault-Tolerant Flight Control and Guidance Systems: Practical Methods for Unmanned Aerial Vehicles,” Springer, 2009.
[12] Sadeghzadeh, I. & Zhang, Y. (2011). A Review on Fault-Tolerant Control for Unmanned Aerial Vehicles (UAVs). St. Louis, Missouri, USA: The American Institute of Aeronautics and Astronautics.
[13] Rao, Faheem Muhammad et al. "UAV Emergency Landing Site Selection System Using Machine Vision". J MACH INTELL 1.1 (2016): n. pag. Web.
[14] A.Vukovic, “All-optical Networks – Impact of Power Density”, Panel
on “Challenges of Electronic and Optical Packaging in Telecom and
Datacom Equipment”, Maui, Hawaii, USA, July 2003.
[15] Wei, F., Amaya-Bower, L., Gates, A., Rose, D. and Vasko, T. (2016). The Full-Scale Helicopter Flight Simulator Design and Fabrication at CCSU. 57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference.
[16] Jirgl, M., Boril, J., Jalovecky, R. (2015). The identification possibilities of the measured parameters of an aircraft model and pilot behavior model on the flight simulator. International Conference on Military Technologies (ICMT), vol., no., pp.1-5.
[17] Kaviyarasu, A. and Senthil Kumar, K. (2014). Simulation of Flapping-wing Unmanned Aerial Vehicle using X-plane and Matlab/Simulink. Defence Science Journal, 64(4), pp.327-331.
[18] Heaton, J. (2005). Introduction to neural networks with Java. St. Louis: Heaton Research.
[19] McClelland, J. (2015). Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises (2nd ed.). Stanford.
[20] Tveter, D. (1995). Chapter 2, The Backprop Algorithm.
[21] K. Winter, I. J. Hayes and R. Colvin, "Integrating Requirements: The Behavior Tree Philosophy," 2010 8th IEEE International Conference on Software Engineering and Formal Methods, Pisa, 2010, pp. 41-50.