Defence Science Journal, Vol. 71, No. 4, July 2021, pp. 448-455, DOI : 10.14429/dsj.71.16164

Software in Loop Simulation based Waypoint Navigation for Fixed Wing UAV

A. Kaviyarasu*, A. Saravanakumar, and M. Logavenkatesh

Madras Institute of Technology, Anna University, Chennai - 600 044, India

E-mail: isrokavi@gmail.com

ABSTRACT

Software in loop simulation (SILS) based waypoint navigation test platform being presented in this paper for fixed wing unmanned aerial vehicle. The proposed platform helps to test waypoint navigation algorithm before implementing into real time environment. Matlab/Simulink and X-plane flight simulator are chosen for the proposed platform. The interface between these two platforms are done by using user datagram protocol (UDP). The waypoint navigation which is to be tested is run in Matlab/Simulink environment where as fixed wing model runs in X-plane simulator. Inverted T tail fixed wing unmanned aerial vehicle configuration is chosen for this research work to verify both its inner loop (attitude control) and outer loop (navigation control). Navigation algorithm executed in Matlab/Simulink compares difference between current and desired latitude longitude position to command flight simulator to reach its desired waypoint. Navigation towards a desired waypoint will be achieved by varying inner loop attitude command of an unmanned aerial vehicle. Finally results are observed and performances are verified in X-plane simulator.

Keyword: X-plane; Waypoint navigation; Unmanned aerial vehicle; User datagram protocol; UDP

1. INTRODUCTION

In recent days autonomous unmanned aerial vehicles have grown rapidly in the field of robotics of its ability which includes surveillance, rescue, traffic monitoring, land surveying and disaster management[1]. Development of unmanned aerial vehicle has gained a tremendous growth both in commercial and defense field of its potential application. Fixed wing unmanned aerial vehicle have an advantages of flying longer time whereas rotary wing have advantage of hovering, takeoff and landing without any runway. The heart of any type unmanned aerial vehicle is its controller generally called as an autopilot.

Autopilot is a device which is mainly used to guide unmanned aerial vehicle without any human assistance. Autopilots are initially developed for missile and later its application is extended to both ships and planes[2]. The most critical process in any autopilot design is tuning of both inner loop and outer loop gain which needs several flight simulation and trails. This may be expensive and time consuming. It helps to makeover gain parameters and control algorithm, therefore a platform such as Software in Loop System (SILS) and Hardware in Loop System (HILS) are needed. This will permit fast development, safety and reduce minimum real-time experimental flights[3-4]. Once the autopilot is tuned in simulation environment the gain which was adopted in simulation has been used in real flight which ensures the performance of autopilot in real time. In recent years different path planning algorithm and navigation algorithm are proposed by different researchers around the world.

Received : 13 August 2020, Revised : 22 October 2020
Accepted : 04 April 2021, Online published : 01 July 2021

Santana[5], *et al.* proposed AR parrot based outdoor navigation for rotary wing unmanned aerial vehicle. Layered control architecture is used to generate reference signal for the navigation. Vision based waypoint trajectory following approach proposed by souza[6], *et al*. Artificial network based method is used in order to recognition and template matching. Indrawati[7], *et al.* developed fuzzy logic based waypoint navigation to realise the performance of control loop by three different flight schemes. Reactive based decision tree model for navigation and deliberative based obstacles avoidance model was proposed by wang[8], *et al.* Song[9], *et al.* proposed vision based method to classify waypoint selection from an aerial image. Supported Vector Machine (SVM) method is used to classify waypoint from the saliency map. To reduce the computational load during waypoint trajectory generation, A Dubin's based optical algorithm is proposed by Arrif[10], *et al.* Islam[11], *et al.* developed fusing method to combine GPS information along with vehicle motion sensor to improve accuracy for real time navigation.

A hybrid control based neighborhood waypoint and its approaching is proposed by Smith[12], *et al.* Low level navigation and high level robot communication for robot navigation has proposed by Hayajneh[13], *et al.* With the proposed method user can fed waypoint input throughout smart phone and web based portal. Particle Swarm Optimisation (PSO) based waypoint navigation and logical decision based re-route of UAV path planning method proposed by Krishnan[14], *et al.* A graph on shortest path planning algorithm from aerial image autonomous waypoint navigation is proposed by Kwak[15], *et.al.* Great circle based on vector algebra to solve great circle sailing proposed by

Chen[16], *et al.* The proposed method helps to calculate latitude and longitude of waypoints more effectively than conventional approaches. Software in loop simulation uses Matlab/Simulink to develop flight control algorithm proposed by Nguyen[17], *et al.* It can be compactable to interface with flight simulators to test and verify flight control algorithms.

Based on literate review, most of current researches mainly focuses on path planning and waypoint navigation for both robotic and unmanned aerial vehicle application. There are plenty of navigation algorithm that are available across research community to improve its performances and accuracy. Mostly these proposed algorithms are simulated and tested by its mathematical model. Testing of navigation algorithm is a tedious process, needs lot of simulation and flight trails to tune the algorithm. Most of the testing are expensive and takes time to validate the algorithms. Sometimes, improper tuning of algorithms lead to crash unmanned aerial vehicle. So there is a need of platform to check and evaluate the performance of algorithm before implementing into real time. From the literature review it is noted that there is no platform available to check and validate the navigation and path planning algorithms. The research work proposed on the above planning to check and evaluate navigation and path planning using Matlab/Simulink and X-plane flight simulator

2. SOFTWARE IN LOOP SIMULATION ARCHITECTURE

Figure 1 shows the architecture of proposed platform for test and demonstrating waypoint navigation through software in loop simulation environment. Matlab/Simulink has been chosen for this research work of its compatibility with X-plane flight simulator to exchange data through its User Datagram Protocol. This platform consists of Matlab/Simulink which executes flight control and navigation algorithm. The output of algorithm is fed to X-plane simulator where its runs the fixed wing unmanned aerial vehicle model. Outputs from X-plane simulator are fed to the Matlab/Simulink through User Datagram Protocol (UDP). The updated output signal from the controller is applied to X-plane Simulator through UDP. This process is being repeated between simulator and controller to make the loop close. This will help to move the unmanned aerial vehicle to reach its destiny. The platform also has a provision to interface a microcontroller in order to monitor and verify the operation of unmanned aerial vehicle. During simulation it is

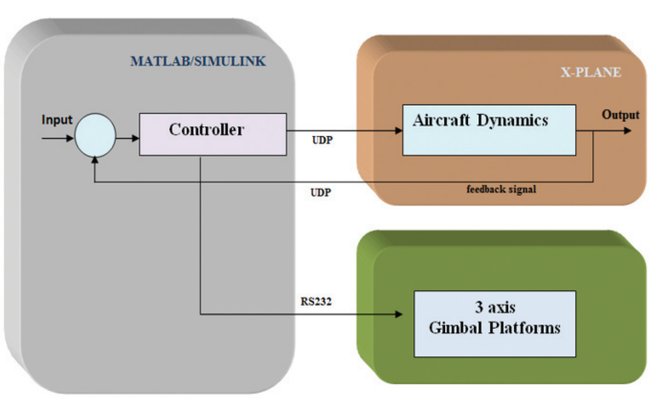

Figure 1. Architecture of proposed SILS platform.

observed that 3 axis gimbal platforms start deflecting as that of X-plane simulator.

Figure 2 represents position controller for fixed wing unmanned aerial vehicle model. Position control for an unmanned aerial vehicle has been achieved by varying its attitude commands of inner loop such as pitch, roll and yaw. Varying the inner loop commands changes its outer loop such as latitude and longitude in 3D space environment. Initially UAV model fly in an arbitrary position, when command being initiated in the current latitude and longitude the desired position and error signal are generated. Depending upon the error signal, position controller in the loop generates attitude inner loop command to guide unmanned aerial vehicle reach its desired position. The above process is repeated until unmanned aerial vehicle attains its desired latitude, longitude position. Sequences of desired position commands maybe upload in Matlab/Simulink controller in order to attain required navigation patterns like circle, zigzag, square based on user application.

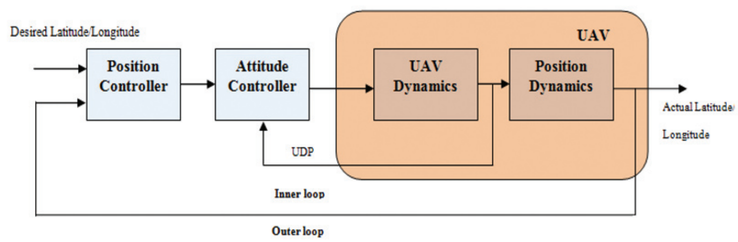

Figure 2. Position controller for fixed wing UAV model.

3. X-PLANE

X-plane is most popular flight simulator among flight community because of its flexible nature and supporting packages. The X-plane community provides plane maker tool setup which is used to design our customised unmanned aerial vehicle depending upon users need. Airfoil maker tool in X-plane package helps to design customised experimental airfoil for unmanned aerial vehicle. The designed flight model in plane maker has been easily imported and executed in X-plane environment. It also allows to create own scenario to replicate exact geo location and easily simulate their model as that of real time.

X-plane uses blade elementary theory where as other simulators like microsoft flight simulator and flight gear simulator uses empirical data in the predefined lookup tables to build the dynamics of an aircraft model. This method has some flaws when aerodynamics and other critical datas are not available to create aircraft model. Blade elementary theory handles this situation by breaking aircraft model into small elements, applying forces and moments over it. The responses from each element are finally applied to the whole aircraft model to make simulator more realistic. X-plane simulator also supports highly developed aerodynamics and comprehensive environment model which may help to aid the scientific researchers to develop flight control system[18]. In addition, X-plane also helps to connect with external world via its User Datagram Protocol (UDP). UDP is a communication protocol which may help to establish low latency and loss tolerating connection between applications. The main advantage of using User Datagram Protocol is to exchange data information with high rate compared with hypertext transfer protocol (HTTP). This will help making the simulator exchange data rapidly and make simulator more realistic when compared to others[18-19].

Data exchange set block in X-plane simulator is shown in figure 3, where important data values such as pitch, roll, yaw, current latitude and longitude information are exchanged through User Datagram Protocol. Selections of data set in the tab are responsible to sent data to Matlab/Simulink through User Datagram Protocol. These datas are generally called as packets; each packet consists of 41 bytes ASCII data information. First four bytes of the packets represents the word "DATA" modulated in the form of ASCII character. Among the byte fifth byte is dedicated for future use. The next bytes D11, D12, D13, and D14 which is known as parametric label helps to identify type of data sent from the X-plane. Followed by remaining 32 bytes of information are encoded in single precision data format[19-20].

Figure 4 shows input/output data values in X-plane simulation environment. Critical flight parameters are usually displayed in simulator which is selected in data tab. All data's in X-plane are encoded in IEE747 format. The encoded packets has collection of datas, first bytes of each data represents positive or negative value. Following bits representing biased exponent and mantissa. X-plane supports up to 131 words which carries most of the flight parameters needed to connect

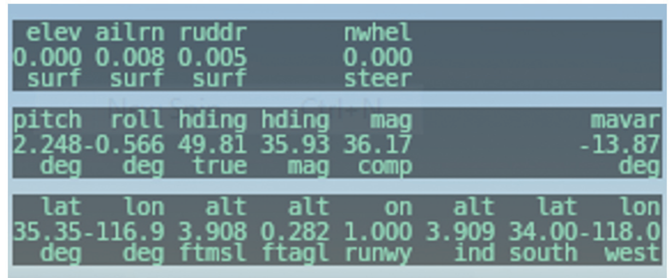

Figure 4. Input/ Output values in X-plane.

with any computational software. The user can get information for user application sthrough its UDP port[21]. UDP port has assigned an unique IP address where all data can be sent and received. In addition X-plane flight simulator allows users to connect their iPhone as well as iPod. This will help users and students to fly their unmanned aerial vehicle model by using iPad or iPhone as a joystick.

Figure 5 represents data establishment in X-plane environment. IP address of system which runs Matlab/Simulink has been set in the tab to transfer data between simulator and computational software. Correspondingly IP address of receiver and transmitter has been set in instrumentation toolbox of Matlab/Simulink environment. If both Matlab/Simulink and X-plane simulator runs in local host IP address 127.0.0.1 is used between the data transmission. However, it is advised to run simulator in separate system to avoid computational load. The data transmission between softwares has been

Figure 3. Data exchange set block in X-plane.

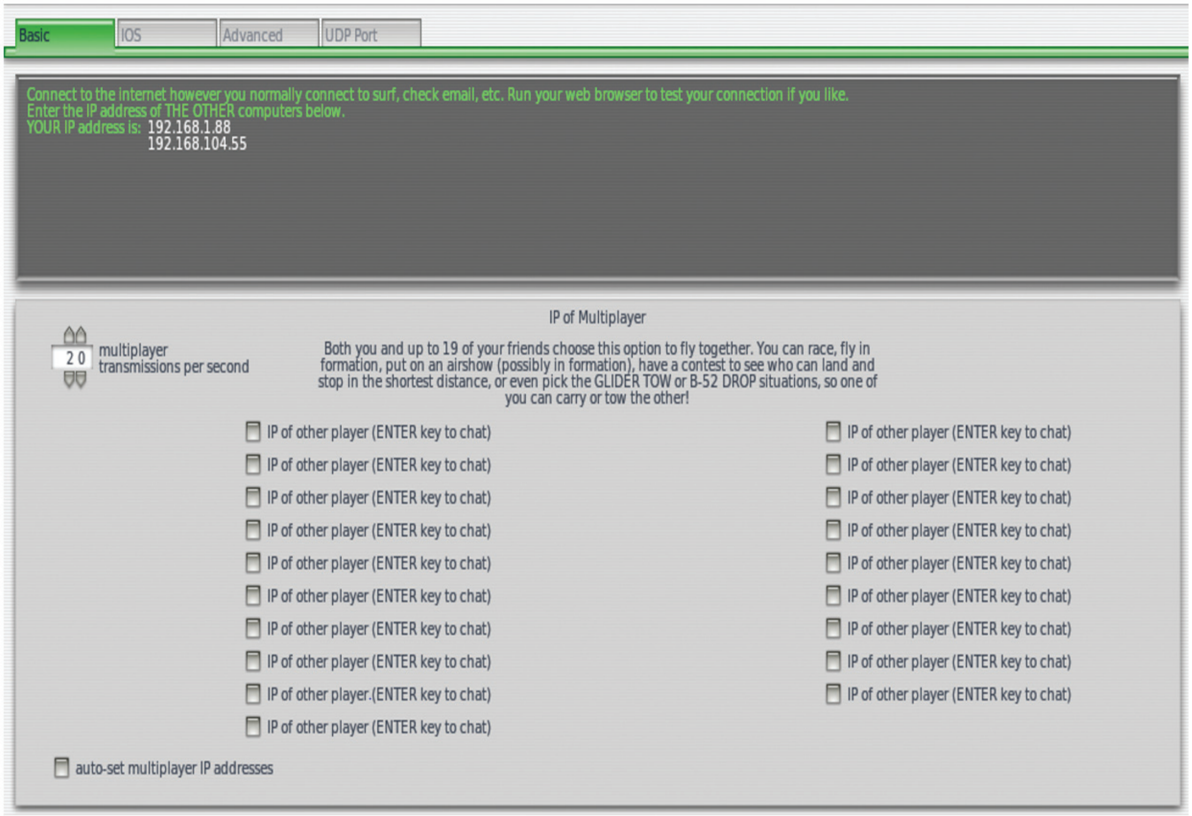

Figure 5. Data establishment in X-plane environment.

done by using UDP connection. In addition to that, X-plane also provides multi-screen external visual setup to enhance simulator reality.[22-23]

4. DISTANCE AND BEARING

Figure 6 shows representation of unmanned aerial vehicle heading and waypoint heading. Heading of unmanned aerial vehicle is measured with help of GPS or magnetometer. Magnetometer is mostly preferred more than GPS, because of less noise and high accuracy. Relative bearing refers to the angle between UAV forward direction with respect to target heading. The distance between any two points in Euclidean space is length between them, but on sphere there are no such straight lines. In space with curvature, such lines are replaced by geodesics are also called as great circle. The shortest path between any two waypoints on sphere called orthodrome under the family of great circle. The shortest distance between any two points on sphere surface is calculated with the help of haversine formula and it is expressed in Equ(3). Bearing between any two waypoint on sphere surface is called as waypoint bearing and its is expressed in Equ(4).

$$a = \sin^2\left(\frac{\nabla\varphi}{2}\right) + \cos\varphi 1 \cdot \cos\varphi 2 \cdot \sin^2\left(\frac{\nabla\lambda}{2}\right) \quad (1)$$

$$c = 2 \cdot a\tan 2\left(\sqrt{a}, \sqrt{(1-a)}\right) \quad (2)$$

$$distance = R \cdot c \quad (3)$$

where R-radius of earth,

$\nabla\varphi = latitude1 - latitude2$

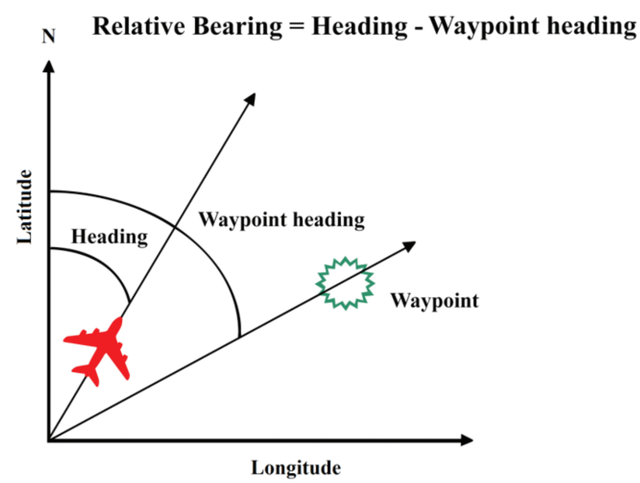

Figure 6. Representation of UAV heading and waypoint heading.

$\varphi 1 = latitude1$

$\varphi 2 = latitude2$

$\nabla\lambda = longitude1 - longitude2$

Bearing between Waypoint

$$Waypoint\ bearing = a\tan 2(\sin(\nabla\lambda) \times \cos(\varphi 2), \cos(\varphi 1) \times \sin(\varphi 2) - \sin(\varphi 1) \times \cos(\varphi 2) \times \cos(\nabla\lambda)) \quad (4)$$

Figure 7 represents flow chart of waypoint navigation for proposed test platform. The algorithm starts with calculating

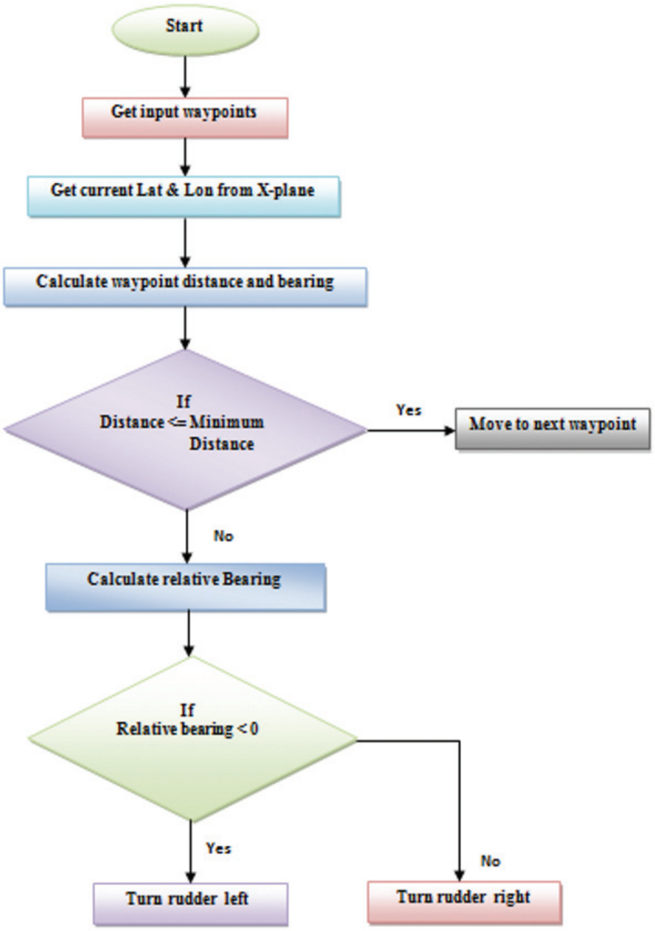

Figure 7. Flowchart of waypoint navigation.

current distance and waypoint bearing between desired latitude longitude position. If error between current position and desired position is zero, the unmanned aerial vehicle moves towards the next waypoint otherwise, relative bearing (angle between UAV true heading and waypoint heading) will calculated and its control surface either rudder or aileron will adjusted to turn the unmanned aerial vehicle fly in desired waypoint. This process is repeated until all other waypoint in the queue are completed, once the vehicle reaches all waypoint it will executes return to home function for safe landing.

5. MATLAB/SIMULINK

Matlab/Simulink plays an important role to execute waypoint navigation under SILS environment. The platform used in this research work sends an error signal to the X-plane simulator by comparing desired latitude longitude with current latitude longitude position through its User Datagram Protocol. Navigation towards the waypoint is being achieved by varying attitude commands of inner loop such as roll, pitch and yaw. Current position and attitude information of an UAV are received from the X-plane software UDP receiver block in Matlab/Simulink. All the received datas from X-plane are in the form of complete packages called as sentences. Each sentence consists of 42 bit of ASCII datas which carries attitude, heading and position information (Latitude and Longitude) of an unmanned aerial vehicle.

Figure 8 shows data transmit section in Matlab/Simulink environment. Data transmit to X-plane simulator is similar to receiving data from X-plane simulator except 5 bit which is set to be zero in the packet. Following byte of information are index which input commands containing inner loop and outer

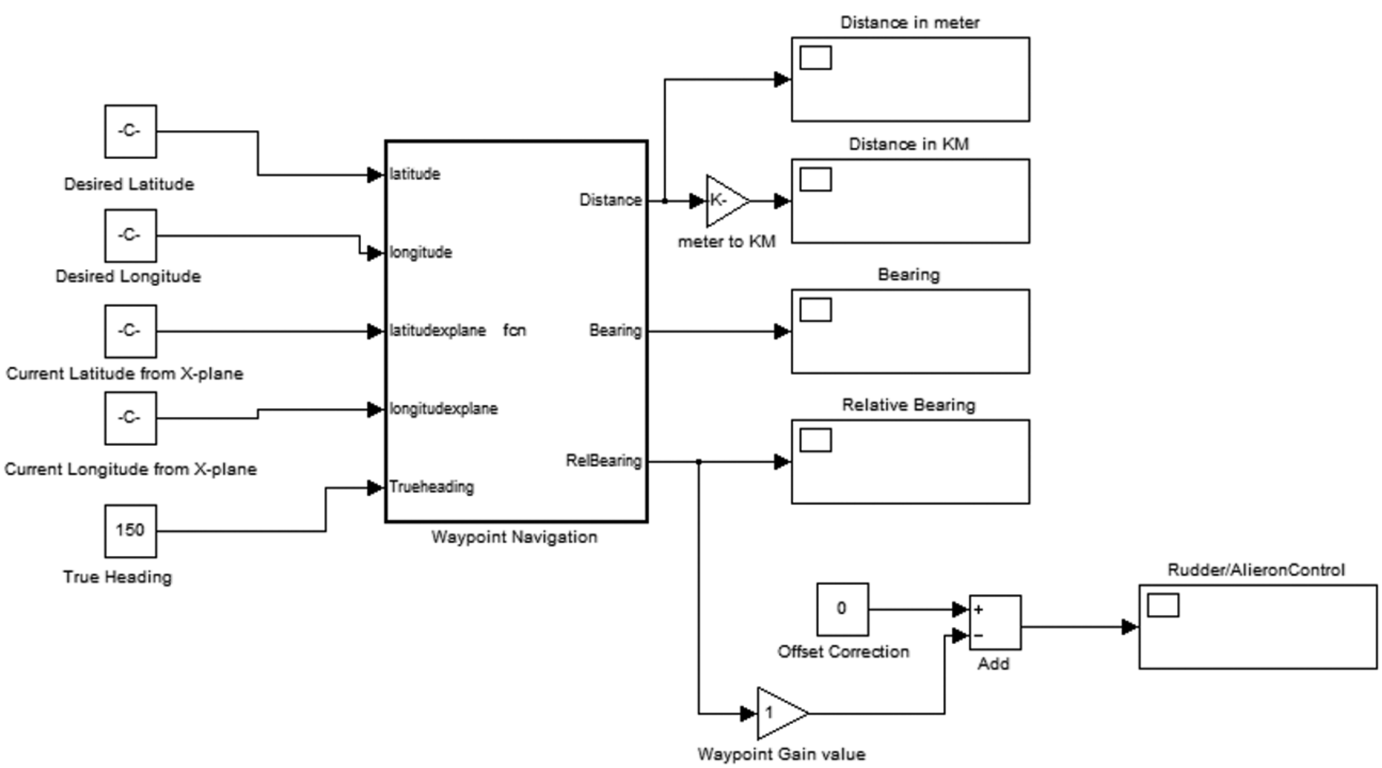

Figure 8. Data transmit in Matlab/simulink environment.

loop command input to the simulator. Depending upon input command signal, unmanned aerial vehicle adjusts its control surface and tries to reach its desired latitude longitude position. Distance and relative heading of unmanned aerial vehicle plays a major role to successful execution of waypoint navigation which has been calculated by haversine formula Eqn(2) and waypoint bearing Eqn(3)**.** Once the unmanned aerial vehicle reaches its desired waypoint with an accuracy of about one meter the next waypoint will execute and move towards the successive waypoint.

6. SOFTWARE IN LOOP SIMULATION EXPERIMENT SETUP

Figure 9 represents simulation platform setup to carry out software in loop simulation experiment. The proposed platform consists of two separate systems which contains both Matlab/Simulink and X-plane software. Matlab/Simulink runs control algorithm whereas X-plane runs its 6DOF flight simulation. Output from navigation algorithm which runs in Simulink has been fed via ethernet UDP port to flight simulator. Similarly, output data from X-plane flight model are also exchanged through UDP. These exchange of data makes system to work in closed loop environment throughout the simulation. Various commands like arming, takeoff, loiter towards waypoint and return to home functions has been executed either by RC Joystick or through Matlab/Simulink switch commands.

Waypoint bearing of an unmanned aerial vehicle is being calculated by comparing current latitude longitude position with desired latitude longitude position. The difference between true heading and waypoint heading provides relative bearing of an unmanned aerial vehicle**.** Depending upon relative bearing the control algorithm will adjust its rudder/aileron surface to turn unmanned aerial vehicle to reach its desired waypoint. This process is easily visualised in X-plane environment, once the user set desired latitude longitude position, the control surfaces produces necessary control deflections to reach its desired waypoint. In addition to that, X-plane simulator provides multi screen interface facility, that enables the students and researchers to look various flight parameters and datas with dedicated display. Further the inbuilt software in X-plane allows logging flight data which may help to tune waypoint navigation algorithm.

7. RESULT AND DISCUSSION

Figure 10 show SILS based waypoint navigation output of an unmanned aerial vehicle model under X-plane simulation environment. Once the waypoint input command is received from user, the control algorithm in Matlab/Simulink executes waypoint navigation algorithm which is served as an outer loop. Navigation towards waypoints has been achieved by varying attitude command of input loop. Haversine formula helps to calculate relative distance and angle bearing between waypoint. Once direction of an unmanned aerial vehicle is identified from its true north, the corresponding rudder/aileron will be adjusted in a closed loop environment to guide the unmanned aerial vehicle towards next waypoint. Meanwhile desired altitude of an UAV will be achieved by controlling its control surface elevator. Once the unmanned aerial vehicle has reached its designated waypoint, the next waypoint will be executed automatically and the process is repeated until it reaches all waypoint. In addition to that, the autopilot also executes some important functions like loiter, circle and return to launch waypoint in order to complete the mission successfully. From the above figure it is observed that, mission start with an auto takeoff reaching an altitude about 200 meters, moving towards second waypoint by executing 90 degree turn on right heading towards third and fourth waypoint. Finally, the controller executes Return To Launch (RTL) command and it will proceed to return to its original launch position.

Figure 11 shows various SILS based outputs run in X-plane environment. It is noted from the above figures, as soon as the take off command has been fed from Matlab/Simulink environment, plane in the X-plane simulator will start and takeoff from the runway and reach 200 meters altitude by heading towards second waypoint rolling its wing towards right.Once it reaches the desired heading, the unmanned aerial vehicle heads towards the second waypoint in a steady and levelled position and turns left/right side to

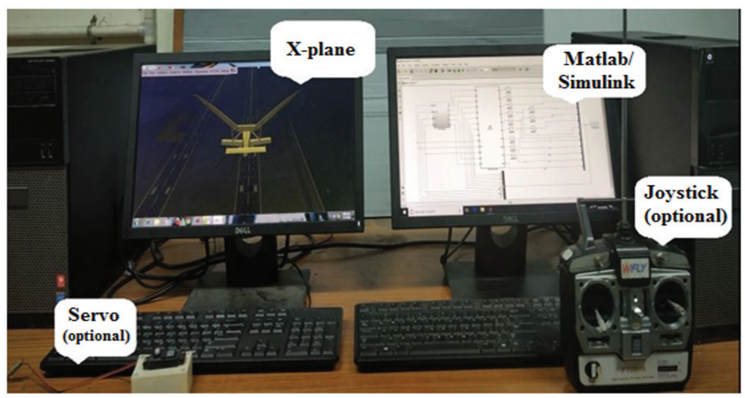

Figure 9. Simulation platform setup.

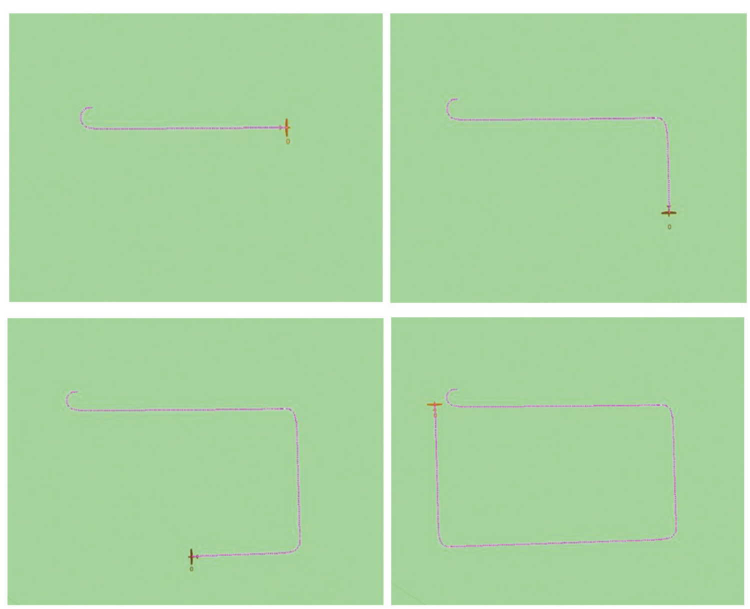

Figure 10. Waypoint navigation executed in SILS environment.

Figure 11. Software in the loop simulation runs in X-plane environment.

reach the second waypoint. The process is repeated until all uploaded waypoints are reached by unmanned aerial vehicle. Finally the onboard algorithm in Matlab/Simulink executes the Return To Home (RTH) command which makes the unmanned aerial vehicle to reach towards the original takeoff position.

8. CONCLUSION

Software in loop simulation based on waypoint navigation simulation platform is set and its results are executed in Matlab/Simulink and X-plane environment. Development of SILS platform helps to aid students, researchers and autopilot designers to validate their control and navigation algorithm before implementing into the real-time environment. It helps to monitor, tune and validate various response of designed autopilot model and to evaluate control gain for the closed loop system. This developed platform plays a vital role to simulate and verify algorithm in short span of time. It may help to reduce potential dangers such as crash and fly-away during real time experiments. In addition to that, this platform helps to evaluate newly proposed algorithm without any risk. This paper focus only on the SILS based simulation environment, In future, the gain derived from the proposed platform will be tested with embedded flight controller and the results may be compared with SILS environment. This will make the proposed waypoint based SILS platform more realistic.

REFERENCES

1. Hajiyev, C. & Vural, S.Y. LQR Controller with Kalman Estimator Applied to UAV Longitudinal Dynamics. Positioning., 2013, **4**, 36-41.
   doi: 10.4236/pos.2013.41005
2. Chao, H.; Cao, Y. & Chen, Y. Autopilots for small unmanned aerial vehicles: A survey. Int. J. Control Autom. Syst., 2010, **8**(1), 36–44.
   doi: 10.1007/s12555-010-0105-z
3. Ganesan, R.; Raajini, X.M.; Nayyar, A.; Sanjeevikumar, P.; Hossain, E.; Ertas, A.H. BOLD: Bio-Inspired Optimized Leader Election for Multiple Drones. Sensors., 2020, **20**, 3134
   doi: 10.3390/s20113134
4. Bittar,A.; Oliveira, N.F.; & Figuereido, H.V. Hardware-In-the-Loop Simulation with X-Plane of Attitude Control of a SUAV Exploring Atmospheric Condition. J.Intell Robot Syst., 2014, **73**, 271–287.
   doi: 10.1007/s10846-013-9905-8
5. Santana, L.V.; Brandão, A.S.; & M. Filho,S. Outdoor waypoint navigation with the AR.Drone quadrotor. In Proc, International Conference on Unmanned Aircraft Systems, Denver, CO, 2015, pp. 303-311.
   doi: 10.1109/ICUAS.2015.7152304.
6. Souza, J.R.; Pessin,G.; Shinzato,P.Y.; Osorio,F.S.; & Wolf, D.F. Vision-based Waypoint Following using Templetes and Artificial Neural Networks. Neurocomputing., 2013, **107**, 77-86.
7. Indrawati,V.; Prayitn, A.; & Kusuma T.A. Waypoint Navigation of AR.Drone Quadrotor Using Fuzzy Logic Controller. Telkomnika 2015, **13**, 930-939.
   doi: 10.12928/TELKOMNIKA.v13i3.1862
8. Wang,Y.; Mulvaney, D.; Sillitoe, I.; & Erick, S. Robot Navigation by Waypoints. J Intell Robot Syst. 2008, **52**, 175–207.
   doi: 10.1007/s10846-008-9209-6
9. Song, L.; Cheng, M.Y.; Yu, L.; & Yu, L. A. Method of

Waypoint Selection in Aerial Images for Vision. Navig Int. J. of Opt., 2014, **2014**, 1-8.
doi: 10.1155/2014/161026

10. Ariff, O.K & Go, T.H. Waypoint Navigation of Small-Scale UAV incorporating Dynamic Soaring. J.navig., 2011 **64**. 29 - 44.
doi: 10.1017/S0373463310000378.
11. Islam, M.D & Kim J.M. An effective approach to improving low-Cost GPS positioning accuracy in real-Time Navigation. *The Scientific World J.,* 2014, **2014**, 1-8.
doi: 10.1155/2014/671494
12. Smith,D.W & Sanfelice R.G. A Hybrid Feedback control startgey for autonomous waypoint transiston and loitering of unmanned Aerial Vehicles. Nonlinear Analysis: Hybrid systems.,2017, **26**,115-136.
13. Hayajneh, M.; Melega, M.; & Marconi, L. Design of autonomous smart phone based quadrotor and implementation of navigation and guidance system. Mechatronics., 2018, **49**,199-133.
doi: 10.1016/j.mechatronics.2017.11.012
14. Krishnan, P.S & Manimala, K. Implementation of optimized dynamic trajectory modification algorithm to avoid obstacles for secure navigation of UAV. appl.soft Computing, 2020, **90**, 1-8.
doi: 10.1016/j.asoc.2020.106168
15. Kwak, J & Sung, Y. Autonomous UAV Flight Control for GPS-Based Navigation, *IEEE Access.*, 2018, **6**, 37947-37955.
doi: 10.1109/ACCESS.2018.2854712.
16. Chen, C.L.; Hsu, T.P.; & Chang J.R. A Novel Approach to Great Circle Sailings: The Great Circle Equation. J. Navig., 2004, **57**, 311-325
doi: 10.1017/S0373463304002644
17. Nguyen, K.D. & Nguyen,T. Vision-Based Software-in-the-Loop-Simulation for Unmanned Aerial Vehicles Using Gazebo and PX4 Open Source. In Proc, 2019 International Conference on System Science and Engineering, Vietnam, 2019, 429-432.
doi: 10.1109/ICSSE.2019.8823322.
18. Sun, J.; Li, B.; Wen, C. & Chen, C. Design and implementation of a real-time hardware-in-the-loop testing platform for a dual-rotor tail-sitter unmanned aerial vehicle, Mechatronics, 2018, **56**, 1-15.
doi:10.1016/j.mechatronics.2018.10.001
19. Bittar,A.; Figuereido,H.V.; Guimaraes,P.A & Mendes,C.A. Guidance Software-In-the-Loop simulation using X-Plane and Simulink for UAVs. *In* Proc, 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, 2014, 993-1002.
doi: 10.1109/ICUAS.2014.6842350
20. Figueiredo,H.V. & Saotome,O. Simulation Platform for Quadricopter: Using Matlab/Simulink and X-Plane. InProc. of Brazilian Robotics Symposium and Latin American Robotics Symposium 2012: Fortaleza, 2012, 51-55.
doi: 10.1109/SBR-LARS.2012.15
21. Garcia, R., Barnes, L. Multi-UAV Simulator Utilizing X-Plane.J Intell Robot Syst 2010, **57**, 393-406.
doi:10.1007/s10846-009-9372-4
22. LAMINAR RESEARCH, "X-Plane Description", Plane Maker Manual, 2013.
23. LAMINAR RESEARCH, "X-Plane Description", X-Plane Manual, 2013.
24. https://www.x-plane.com
25. http://www.mathworks.com
26. https://www.movable-type.co.uk/scripts/latlong.html

ACKNOWLEDGEMENTS

The author express his sincere thank to all faculty and students of Department of Aerospace Engineering, MIT Chennai and Centre for Aerospace Research, Anna University for their support and encouragement.

CONTRIBUTORS

Dr A. Kaviyarasu received his doctorate from Anna University Chennai. He is currently working as an assistant professor in the Division of Avionics, Department of Aerospace Engineering, Anna University, Chennai. His areas of interest are Design, modelling and simulation of Unmanned Aerial Vehicle. He has been awarded for Best Innovative project from Anna University, Chennai.
Contribution in current study, he guided the entire project, interpretation of results and prepared the manuscript.

Dr A. Saravanakumar received his doctorate from Anna University Chennai. He is currently working as an Assistant professor in the Division of Avionics, Department of Aerospace Engineering, Anna University, MIT Campus, Chennai. His areas of interest are fault tolerant control and UAV Acoustic. He has been awarded as Best mentor from Centre for Technology Development and Transfer, Anna University.
Contribution in Current study, he has carried out implementation of waypoint navigation algorithm and it interface.

Mr M. Logavenkatesh received his bachelor degree from Anna University Chennai. He is post graduate (ME Avionics) student from Madras Institute of Technology, Anna University, Chennai. His areas of interest are simulation of unmanned aerial vehicle and navigation.
Contribution in current study, he has carried out implementation of algorithm and simulation in MATLAB.