Proceedings International Seminar of Aerospace Science and Technology 17th SIPTEKGAN -2013, 08: 55-60

CONTROLLING X-PLANE FLIGHT SIMULATION ENVIRONMENTS FROM MATLAB FOR RKX-EDF LAUNCH SCENARIO*

Hakiki, Edi Sofyan
Rocket Technology Center, National Institute of Aeronautics and Space - LAPAN
thehaqq@yahoo.com

Abstract

This paperdescribes the establishment of the software system for RKX200-EDF launch simulation. The flight RKX200-EDF platform is modelled in X-Plane, and several of its parameter in X-Plane are to be controlled by Matlab. The communication between Matlab and X-Plane is realised through UDP port. The result of the RKX200-EDF simulation shows the trajectory, the speed history and the flight position and attitude which can then be studied further for flight assessment of the flight dynamic and performances. Once the autopilot and guidance system is available, then a more thorough analysis of the vehicle performance scenario can be studied. The developed system is therefore very significant in the development of the flying vehicle system whereby the dynamics, visualization and the analysis for many computation algorithms are to be designed.

Key Words: *System Tool Kit (STK), Inertial Measurement Unit, Flight Test, UAV, Rocket*

1. Introduction

There has been several flying vehicles being developed in the National Institute of Aeronautics and Space. Most of these vehicles are designed and tested in the Guidance and Telemetry System Division. These vehicles are mostly used as flying test-bed in research and development of control, guidance system, sensors, telemetry system and so on. Several types of these vehicles are shown in Figure 1. The different in shapes reflects the different mission these vehicles are to carry. Some carries turbojet engines, autopilot systems, telemetry systems, and some vehicles are built to serve for pilot training only.

In 2013, the guidance division design an electric ducted-fan (EDF) flying vehicle with diameter approximately 200mm and hence the name RKX200-EDF, as can be seen in Figure 2 below. The system is built to test several autopilot systems, waypoints, payload systems for medium speed application such as those reaching at least 200 km/hour cruise speed.

During the RKX200-EDF development, a matlab-X-Plane integration is being studied and seen the possiblefurther benefit in integrating both software together to solve some of engineering problems encountered in the field. For the first project, we are studying the RKX200-EDF launch scenario in the first few second while the motor rocket booster is still firing. The dynamics and trajectory of these scenario is to be determined, so at to assess the effectiveness of the launcher and vehicle design.

Figure 1: Flying vehicles to be used as test platform in guidance division - LAPAN

Figure 2: RKX-EDF 1st prototype to be flight tested on end year 2013

The aims of this project is therefore:
-to simulate RKX-EDF launch scenario in X-Plane.
- to controlX-Plane environment during launching RKX-EDF from within Matlab.
- to post-process the result of the simulation further inMatlab.

2. System Development

The system to be developed mainly consists of three components as shown below. The flight system simulation software which in this case the X-Plane software, the Matlab software which controls the X-Plane environment, and the communication unit between the two which uses a TCP/IP UDP protocols.

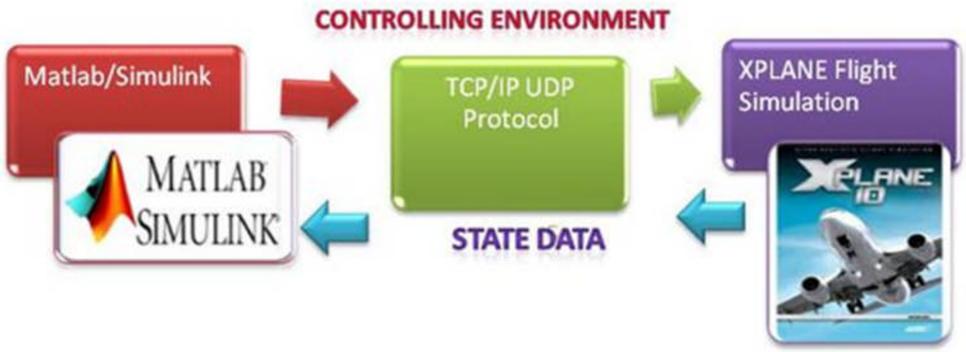

Figure 3: RKX-EDF launch scenario in X-Plane and mat lab environment

Main reasons for connecting MATLAB to X-Plane;
• MATLAB is a graphical programming environment that has very powerful numeric array and string handling capabilities, and easy numeric to string conversion. Matlab also has Aerospace Toolbox and Blockset which facilitates for advanced analysis for flying vehicles.
• Building user interface in MATLAB is relatively simple, which can reduced significantly the development time for the software system.
• MATLAB has basic TCP functions built in – Open, Write, Read,Close Connection, which can then be used to facilitate communication with other software and hardware.
• Additional functions can be easily added to create Connectcommands from raw telemetry data to further analyse and visualise the collected information.
• X-Plane can perform flight dynamic analysis. Given the flying vehicle model which can be prepared in Plane-Maker or some other CAD software.
• X-Plane can perform graphical visualization of position uncertainty at any specified times.

3. System Components

3.1. X-Plane version 10.

To enable testing of controllers developed in MATLAB and ensure properconversion from MATLAB to C code, an aircraft simulation environment was utilized.Similar to Microsoft's Flight Simulator, X-Plane provides extremely accurate flight models—accurate enough to be used to train pilots [1] and also allows externalcommunication as well as airfoil design. Unlike options such as Microsoft Flight Simulator and Flight Gear, however, X-Plane allows input and output from an external source, which is the main reason we use this software. X-Plane also provides future capabilities that unmanned objects will need including navigation markers, changing weather conditions, and air traffic control communication.

3.2. X-Plane UDP Communication

X-Plane uses UDP communication to send and receive data packets which allowschanges to various values within X-Plane, but UDP protocol has its strengths and weaknesses. Over a distant network connection, UDP may be unreliable because no error detection exists in the packets; however, UDP is extremely fast. X-Plane is able to dump up 50 frames per second across a local network—this has an important impact on the operation and simulation of many controllers because they require sufficient update17speed to operate correctly. X-Plane offers a large variety of values that can be changed including control of the aircraft as well as causing in-flight failures. To select which data items to export to SIMULINK, X-Plane provides an easy to use check box interface.

3.3. X-Plane Exported UDP Data

X-Plane has the ability to send and receive a large number of parameters allowing controllers to be designed using a variety of sensor readings. When the Inet 2 tab isopened, the IP address of the computer containing the controllers is entered as well as the proper ports. Afterwards, the Data Set tab is chosen and the values requiredby the controllers are selected. X-Plane now begins sending the data to the destination IPand port. In addition, X-Plane will wait to receive packets on the other IP and port specified. In the Data Tab, the speed at which the data will exported is controlled by increasing or decreasing the number of frames per second—a range from 0 to 50. Tosend the data, a UDP packet is formed consisting of the string of characters “DATA”followed by an integer, and then the data items selected in the screen are attached in increasing order of the data item numbers to the packet. For ease of use, each data item consists of one integer (the number specified in the output screen) and 8 float values.The proper index value for a particular data item selected may be determined bydisplaying the selected values to the screen—selecting one of the four checkboxes in thedata output dialog box. The index value for each data item begins with 0. For example,X-Plane data item number 0 is the frame rate, and the frame rate data consists of 318different values which are indexed by 0, 1, and 2. Once, all the data value selections have been assembled into the packet, it is then sent tothe destination IP and port in network byte order. After the header information, each item selected to be output from X-Plane contains 36 bytes.

X-Plane was chosen because it offers a vast amount of model information and flexibility for future development. For flight failures, X-Plane offers the ability to fail GPS, control surfaces such as left roll, right roll, pitch up, pitch down, yaw left, yaw right, roll trim, pitch trim, yaw trim, control throttle jam minimum, maximum, and current, engine failure, engine fire, and engine mixture. Because X-plane has the ability to communicate with multiple aircraft, swarms can be simulated to see how the vehicles will interact with each other. A host of other features are available and can be viewed in UDP documentation.

3.4. Matlab/simulink interfacing with x-plane

Although there exist several simulators like Microsoft’s Flight Simulator, and Flight Gear, X-Plane provides extremely accurate flight models and allows for external communication as well as airfoil design. It is accurate enough to be used to train pilots. Unlike other Flight Simulators, X-Plane also allows for input and output from external sources. X-Plane provides future capabilities that unmanned aerial vehicles will need, including navigation markers, changing weather conditions, and air traffic control communication. The main reason for the selection of X-plane in this part of work is because of its property to easily interface with external sources [3] and [4]. Below shows the implemented simulink diagram used for this RKX200-EDF launch simulation. The simulink received all flight information from X-Plane and send the time and throttle parameter to X-Place to control the launch scenario.

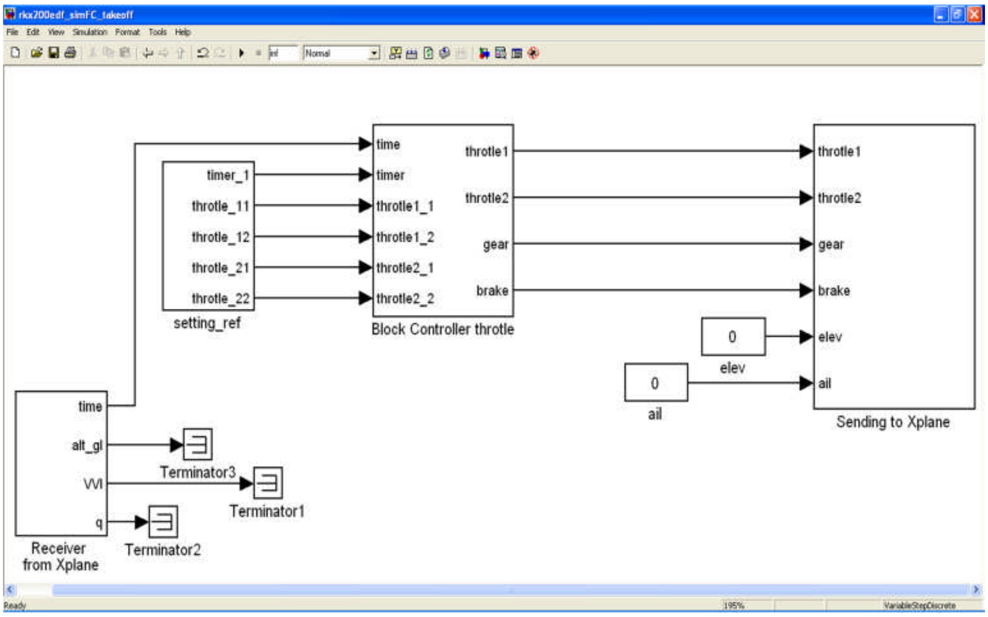

Figure 4: The simulink block used in the launch simulation

3.5. Simulation Results

The simulation was conducted to see how the launch of the flying vehicle RKX200-EDF to be performed. A plane maker was used to add the imported CAD drawing several other model and launcher characteristics.Since no launcher model is available in Plane Maker, therefore an undercarriage is used to model the launcher of the RKX200-EDF as shown in the Figure 5 below. The under carriage will retracts as the RKX200-EDF lift offs the runway and the therefore simulate the launcher. The elevation of the launch is simulated by the different length of the two under carriages. Here the other model parameter such as eight, cg, fuel, engine, etc are inputted in Plane Maker.

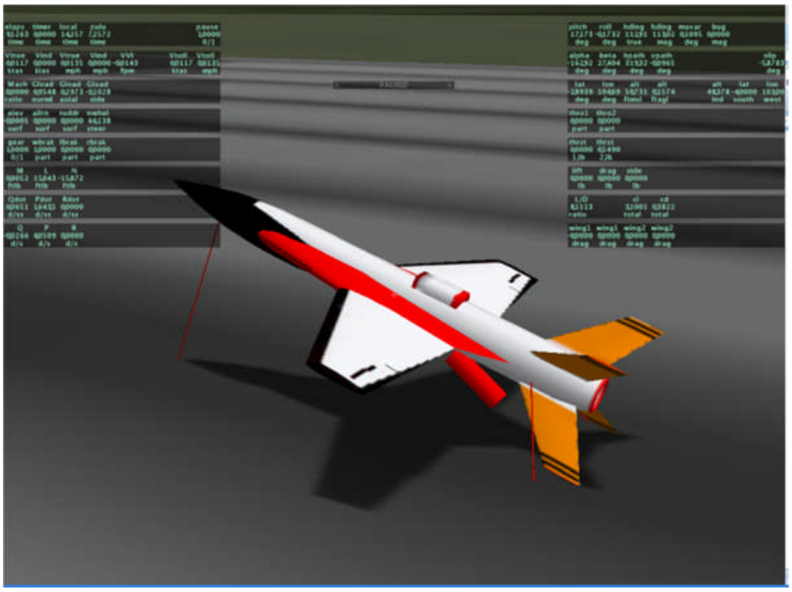

Figure 5. The launch simulation of RKX200-EDF, sitting on a launcher at an elevation angle

As shown in the simulink diagram in Figure 4, the data inputted to the X-Plane is the thrust parameter an timer. whereas the time, velocity, and position (longitude, latitude) and attitude are outputted from the X-Plane, using the built in dynamics already implanted in the X-Plane software [2]. Figure below shows the setting of the input and output data used in X-Plane.

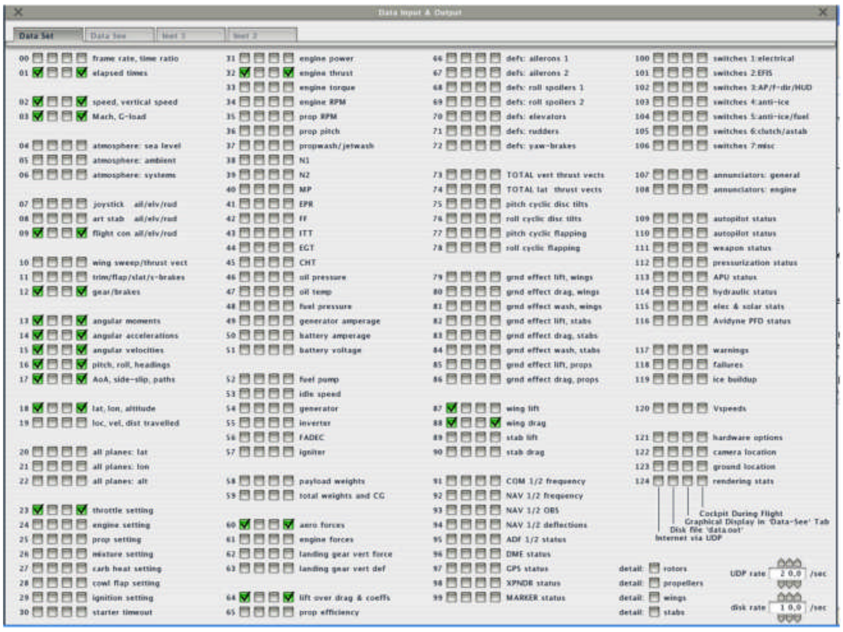

Figure 6. The setting of the X-Plane interface during RKX200-EDF launch simulation

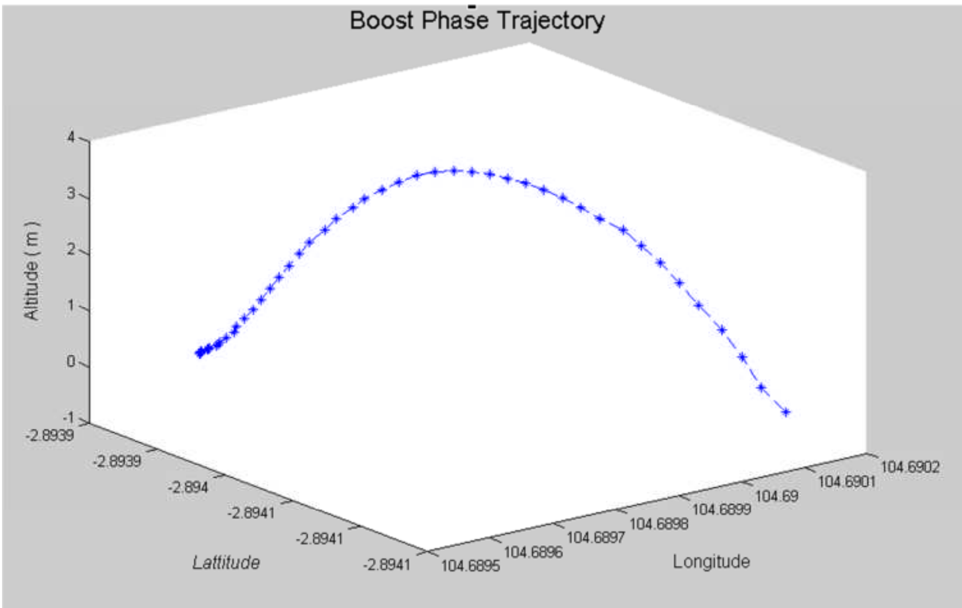

Figure 7. The 3D trajectory of the RKX200-EDF launch simulation

Figure 7 above shows the trajectory results of the RKX200-EDF launch scenario. It can be seen that the trajectory follows the parabolic shape from launch to touching the ground. The Figure 8 shows the 2D trajectory which shows clearly the attained altitude during the launch. This information is very important to make sure that the vehicle has a safe clearance from the ground, and to enable assessment for the performance of the stabilization system.

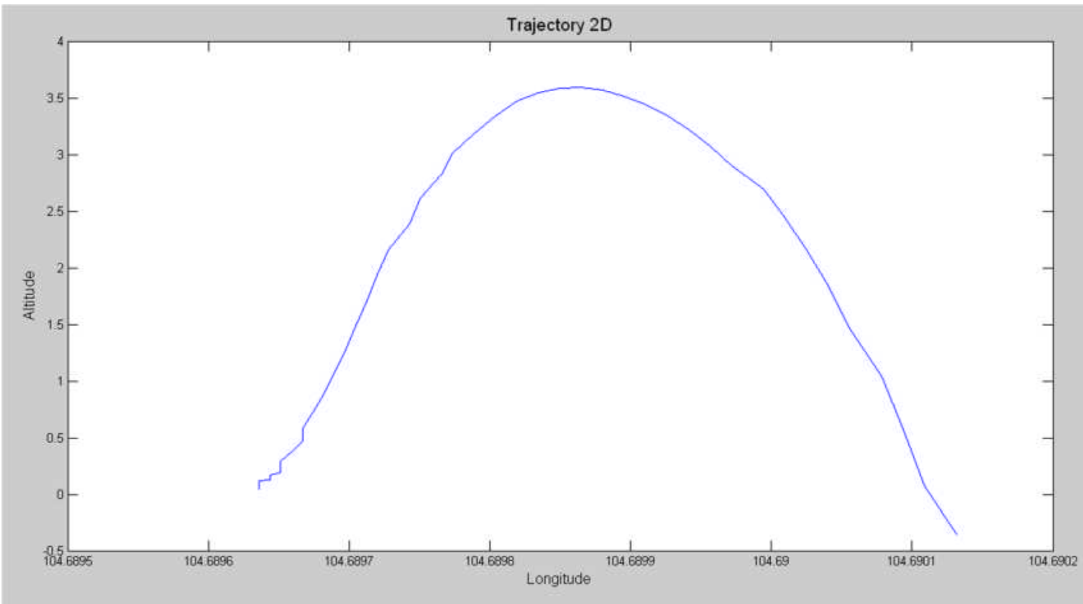

Figure 8. The 2D trajectory of the RKX200-EDF launch simulation

Figure 9 the shows the speed history of the vehicle during the launch. It can be seen that during the launch, the vehicle is accelerated using booster engine to a speed of 35m/sec. This is just the right value for the needed speed that the vehicle can generate sufficient lift force for the sustained flight. In other scenario where a higher attained speed is needed, than the different thrust engine data is injected and the simulation is run again with different setting. The resulting trajectories can then be studied whether or not the vehicle can fly according to the desired flight trajectories. In addition, the speed trajectories can then be seen whether the vehicle achieves the desired max speed needed for the flight.

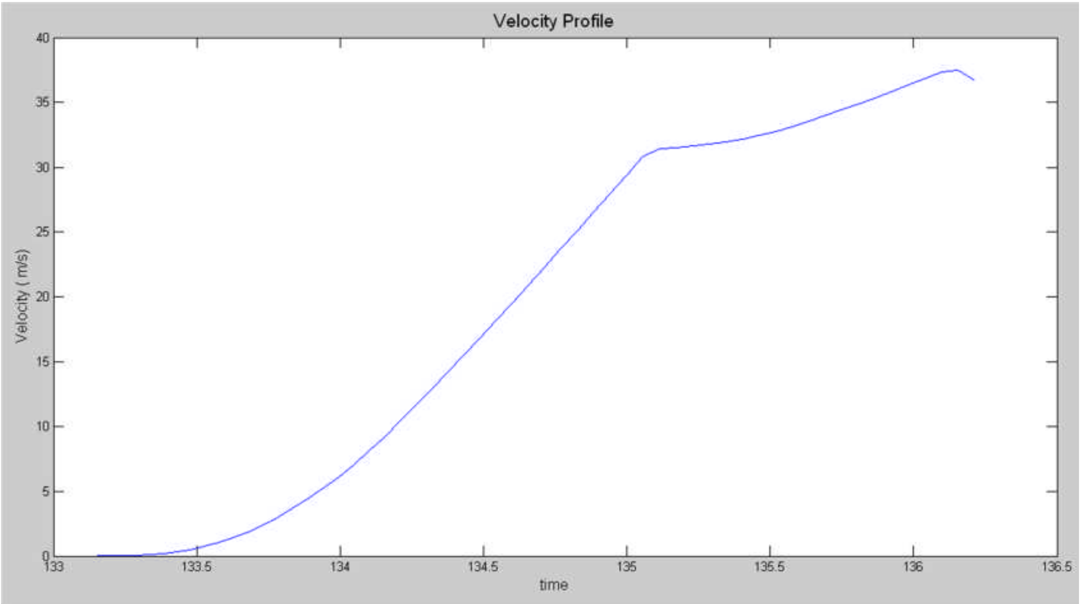

Figure 9.The speed history of the RKX200-EDF during launch

Further work is to be done in this system. An autopilot system can then be added to assess the effectiveness of the added autopilot. The gain settings of the autopilot can also be fine tuned for better performance. In cases where guidance system block simulink in Matlab is available, a more thorough simulation and understanding of the vehicle guidance and autopilot system can be studied, utilising both the capability of X-Plane dynamic and visualisation features, and capability of Matlab and its Simulink for advanced computation.

4. Conclusions

A system for controlling X-Plane simulation from Matlab has been developedsuccessfully. The communication between the two uses an UDP communication. The system is very useful in simulating launch scenario where flight dynamic can be simulated inX-Plane and the control scenario is programmed in Matlab.

The system is then used for simulating the launch of the RKX200-EDF flight platform. The trajectories and other parameters are output from the software and can be analysed further.

More Plug-in and functions can be added to enhance analysis in both Matlab and X-Plane environment.

Matlab and X-Plane are a good combination for simulating flight vehicle dynamics and performance.

Acknowledgments

The authors would like to thank the Guidance Division at LAPAN who has provided the resources and facilities during the development of the X-Plane simulation environment.

References

1) AGI:STK 10 User Manual, 2013
2) Meyer, A. : X-Plane UDP Reference Manual, 2004
3) The Mathworks : MATLAB, Software package, Ver. 7.14, 2012
4) The Mathworks: Simulink, Software package, Ver. 7.9, 2012.