ROCKET SIMULATION BASED ON SOFTWARE-IN-THE-LOOP

Oleh:
Singgih Satrio Wibowo*
Ilham Akbar**
Idris Eko Putro**

Abstract

Software-In-The-Loop system has been developed for research on LAPAN's ballistic rocket. This system can be used to evaluate the control algorithm of rocket. Flight simulation of rocket on this software-in-the-loop system is very useful for predicting rocket's dynamics. Risk and cost reduction can be performed by using this simulation before doing the flig[1]ht test.

This paper will focus in rocket simulation which the rocket model is presented through Matlab/Simuink environment and visualized by X-Plane in 3 dimension. Matlab/Simulink send the output data to the XPlane through cache memory of XPlane plugin. This XPlane plugin has been accessed by communication interface which was developed by Delphi program. Interactive communication between Matlab/Simulink and XPlane become software in the loop for simulation of ballistic rocket.

Keyword : X-Plane, Simulink, Software-in-the-loop

Abstrak

Sebuah sistem software-in-the-loop telah dibuat dalam rangka riset roket balistik LAPAN. Sistem ini dapat digunakan untuk menguji algoritma kendali roket. Simulasi terbang roket pada sistem software in the loop ini sangat berguna untuk prediksi awal dinamik gerak roket. Sistem ini berguna untuk mengurangi resiko dan mengurangi biaya sebelum uji terbang dilakukan.

Tulisan ini akan memfokuskan pembahasan pada simulasi roket, dimana model roket di bangun menggunakan piranti lunak Matlab/Simulink dan di visualisasikan secara 3 dimensi dalam piranti lunak X-Plane. Keluaran data dari model roket pada Matlab/Simulink selanjutnya akan dibaca oleh XPLane melalui cache memory pada pluginya. Plugin dari Xplane ini diakses melalui komunikasi antar muka yang diprogram di dalam program Delphi. Komunikasi yang interaktif antara Matlab/Simulink dengan XPlane ini merupakan simulasi software in the loop dari roket balistik

Kata kunci : X-Plane, Simulink, Software-in-the-loop

1. INTRODUCTION

The development in spacecraft technology has been grown rapidly. It should be supported by creating applicable testing system. So, we tried to build a simulation system known as Rocket Flight Simulator system that can accommodate the future challenges of rocket technology. Flight simulation is one of the most important things that should be before doing the flight test. From flight simulation, we can see the characteristics of an aircraft to find its weakness and strengths.

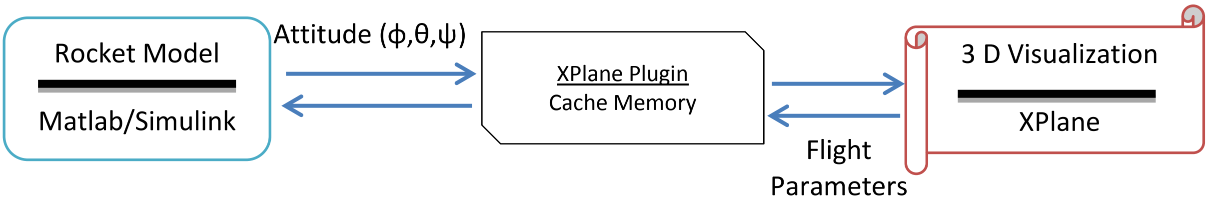
This study is one of activities that carried out in Control and Telemetry Division of Pustekroket (The Center of Rocket Technology) LAPAN. It's also one part of the series of major studies on the project named Sijidalang (The System of Flight Control Test).

X-Plane is flight simulator software that emulates the real-world performance of an aircraft by using lookup tables to find known aerodynamics forces such as lift or drag, which vary with flight condition. X-Plane does a good job of simulating the flight characteristics of the aircraft they were designed to simulate (aerodynamics data). In this case, the quaternion formulation was defined for six degree-of-freedom flight simulation to have a precise rocket model. Rocket model was made also in SIMULINK to provide the desired output data. SIMULINK model rocket and X-Plane model rocket will exchange data each other. Thus, the constant K will be obtained as a matrix that will be used by the microcontroller in Hardware-In-the-Loop.

* Lecturer at Bandung State Polytechnic – Mechanical Engineering, Aeronautics Engineering Program
**Researcher of Pustekroket LAPAN – Control and Telemetry Division

2. METHOD

Interactive communication between two software will be the most important subject to be done. This method will guarantee the data flow from Rocket model to the visualization of rocket flight dynamics in 3 dimension. The Interactive communication can be seen from the chart below:



Firstly, The data will be sent by UDP from X-Plane to another computer. This computer received a typical data from X-Plane. The data contains special header. This method has been done previously by Jeff Lewis[1]. Every version of X-Plane has different type of data packet.

A typical DATA packet are being sent out from X-Plane following the protocol as shown below:

68 65 84 65 38 0 0 0 34 68 151 111 166

The first five bytes are what X-Plane uses for its header. Each of these bytes is actually ASCII codes, so we convert each of them into a symbol. The first 4 bytes of the header tell us what type of packet it is. In this example, they're 68,65,84,65, which correspond to D,A,T,A, respectively, so we know this is a DATA packet. The fifth byte in the header, 38 in this example, is an index used internally by X-Plane. These data 0 0 0 34 68 151 111 166 are the contained data.

3. THEORY

3.1 Rocket Dynamics

Equation of motion for rocket dynamics is generated from Rigid body equation. The motion of this model in three-dimensional space is represented by the position of the center of mass and the Euler angles of vehicle rotation. The Euler-Newton equations are derived from the law of conservation of linear angular momentum.

$$m\left.\frac{d\vec{V}}{dt}\right|_{I} = \vec{F} \tag{3-1}$$

$$\vec{I}\left.\frac{d\vec{\omega}}{dt}\right|_{I} = \vec{M} \tag{3-2}$$

For the rocket moving in six degrees of freedom, the above equations produce six differential equations describing the rocket’s translational motion and angular motion in three references axes.

$$\dot{u} = \frac{F_x - \dot{m}u}{m} - qw + rv \tag{3-3}$$

$$\dot{v} = \frac{F_y - \dot{m}v}{m} - ru + pw \tag{3-4}$$

$$\dot{w} = \frac{F_z - \dot{m}w}{m} - pv + qu \tag{3-5}$$

$$\dot{p} = \frac{M_x I_{xx} + M_z I_{xz}}{\Delta} + \frac{I_{xz}\left(I_{xx} - I_{yy} + I_{zz}\right)pq}{\Delta} - \frac{\left[I_{zz}\left(I_{zz} - I_{yy}\right) + I_{xz}^2\right]qr}{\Delta} \tag{3-6}$$

$$\dot{q} = \frac{M_y}{I_{yy}} + \frac{(I_{zz} - I_{xx})pr}{I_{yy}} - \frac{I_{xz}(p^2 - r^2)}{I_{yy}} \tag{3-7}$$

$$\dot{r} = \frac{M_x I_{xz} + M_z I_{xx}}{\Delta} + \frac{\left[I_{xx}\left(I_{xx} - I_{yy}\right) + I_{xz}^2\right]pq}{\Delta} - \frac{I_{xz}\left(I_{xx} - I_{yy} + I_{zz}\right)qr}{\Delta} \tag{3-8}$$

Where

u	:	Rocket velocity in -x body axis
v	:	Rocket velocity in -y body axis
w	:	Rocket velocity in -z body axis
p	:	Roll rate
q	:	Pitch rate
r	:	Yaw rate
F_x	:	Aerodynamics force in -x body axis
F_y	:	Aerodynamics force in -y body axis
F_z	:	Aerodynamics force in -z body axis
m	:	Rocket mass

3.2 Quaternion

Quaternion defined as:

$$\mathbf{q} = 1 \cdot q_0 + \mathbf{i} \cdot q_1 + \mathbf{j} \cdot q_2 + \mathbf{k} \cdot q_3 = [q_0 \quad q_1 \quad q_2 \quad q_3]^T \tag{3-7}$$

As q_0, q_1, q_2 and q_3 are real number, 1 is the identity of multiplication, and **i**, **j**, **k** are the symbols whose properties:

$$\begin{gathered} \mathbf{i}^2 = -1, \mathbf{j}^2 = -1, \mathbf{k}^2 = -1 \\ \mathbf{ij} = \mathbf{k}\,, \mathbf{ji} = -\mathbf{k} \\ \mathbf{jk} = \mathbf{i}\,, \mathbf{kj} = -\mathbf{i} \\ \mathbf{ki} = \mathbf{j}\,, \mathbf{ik} = -\mathbf{j} \end{gathered} \tag{3-8}$$

Euler angles can be determined from quaternion as

$$\begin{gathered} \varphi = \arctan\left[\frac{2(q_0 q_1 + q_2 q_3)}{q_0^2 - q_1^2 - q_2^2 + q_3^2}\right] \\ \theta = \arcsin[2(q_0 q_2 - q_1 q_3)] \\ \psi = \arctan\left[\frac{2(q_0 q_3 + q_1 q_2)}{q_0^2 + q_1^2 - q_2^2 - q_3^2}\right] \end{gathered} \tag{3-9}$$

Otherwise, quaternion also can be determined as function of Euler angles that formulated as:

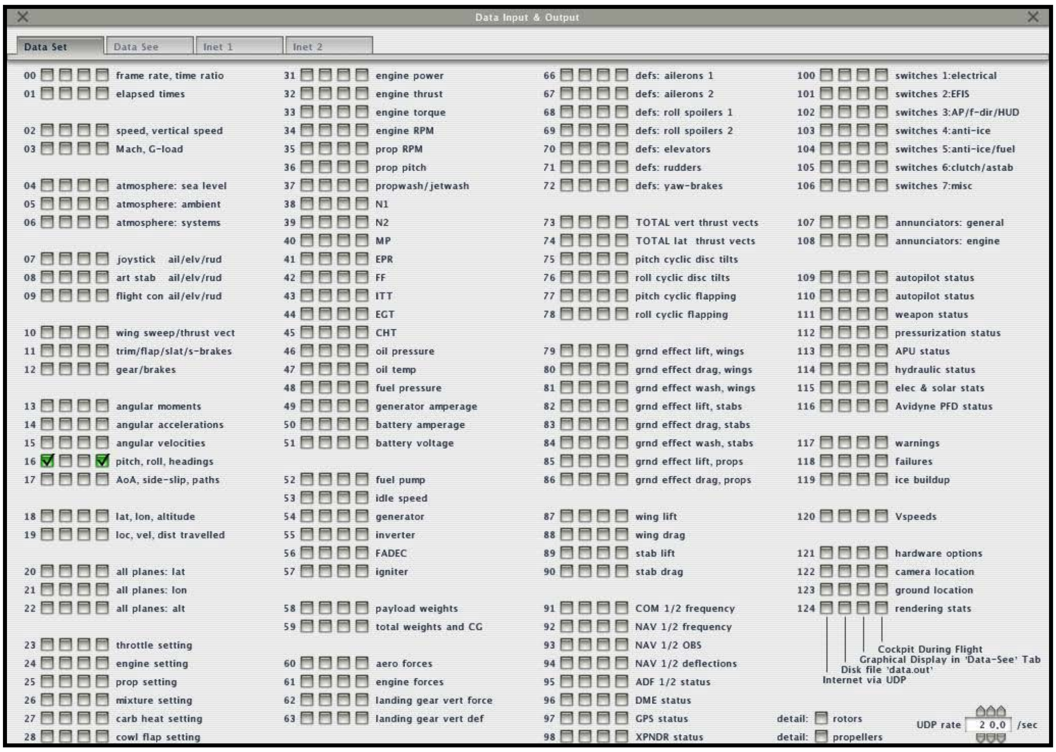
$$
\begin{aligned}
q_0 &= \cos\frac{\varphi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\varphi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} \\
q_1 &= \sin\frac{\varphi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} - \cos\frac{\varphi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} \\
q_2 &= \cos\frac{\varphi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\varphi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2} \\
q_3 &= \cos\frac{\varphi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2} - \sin\frac{\varphi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2}
\end{aligned} \tag{3-10}
$$

3.3 UDP COMMUNICATION

UDP stands for Universal Data Packet. It is a method of sending information over a network. Unlike other protocols, like TCP, UDP does not do any error checking. If we send a packet, and the other computer gets the wrong data, or doesn't even get any data at all, we have no way of knowing, unless we program our own error checking into the program. There were two computers needed to build this kind of communication. X-Plane will be run in the first computer. SIMULINK model of rocket will be run in the other computer. Set these two computers' IP.

For example:

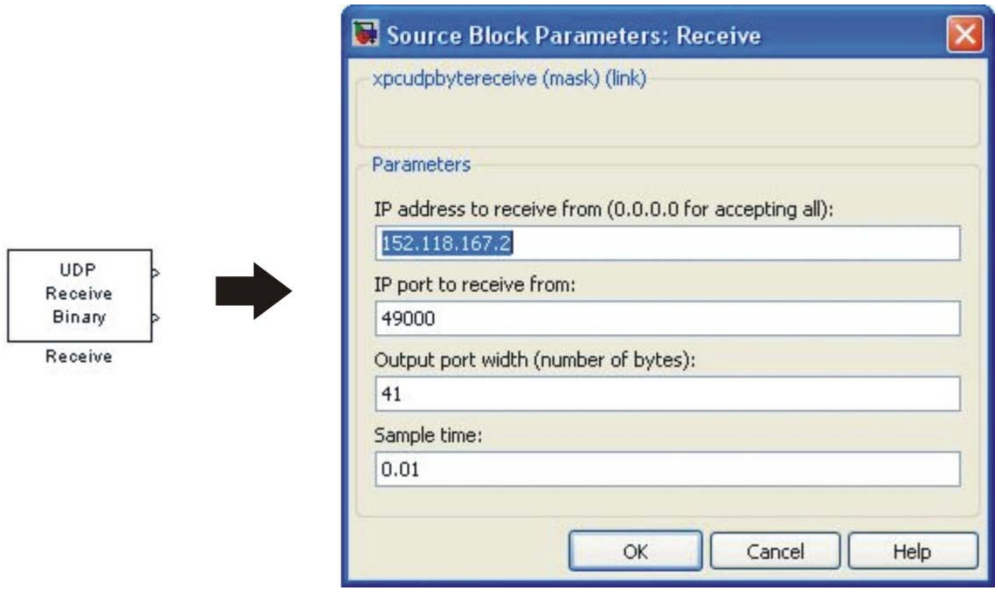
Computer 1 | IP:152.118.167.2 | SIMULINK (MATLAB)
Computer 2 | IP:152.118.167.1 | X-PLANE



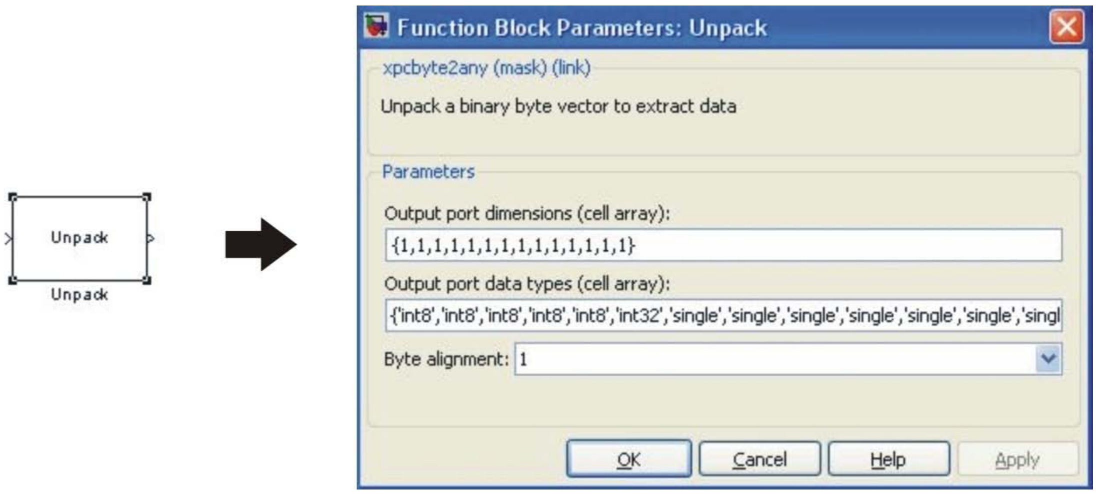
Picture 2-1 Example Setting of Data Input & Output X-Plane

In the left side column of every single data, checkbox is set up to make it communicable via UDP. The right side column is set up to show data on the cockpit during flight.

In SIMULIK, drag 'UDP Receive Binary'. This block receives data packed from X-Plane via UDP. Then these data have to be unpacked by using "Unpack" block.

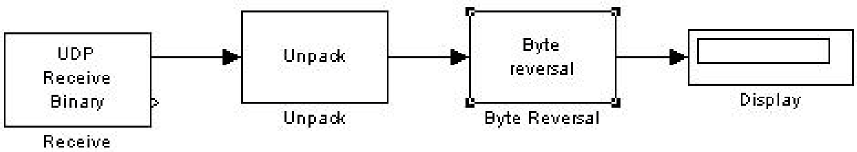


Picture 2-2 Setting example of UDP Receive Binary & its Parameter



Picture 2-3 Setting example of Unpack block and its Parameters

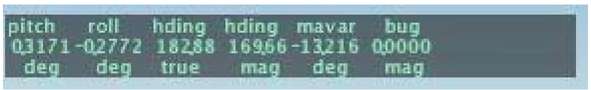
Picture 2-2 and 2-3 are block parameter settings. We have to make sure that we fulfill the parameters with the right values.



Picture 2-4 Example Byte Reversal block uses for data reversing

Unpack block will unpack a binary byte vector to extract data from X-Plane. We need one more Simulink's UDP block to reverse this data. It's Byte Reversal block. The block will reverse the order of byte. We need this because X-Plane was made for uses in Mac OS. Byte order in Mac OS is viseversa with byte order in Windows. Thus, we need this block.

Otherwise, this UDP communication would not work properly.



Picture 2-4 Data packed that sent from X-PLANE

These number of data in table:

Table 2-1 Packet Data Table

Data1	Data2	Data3	Data4	Data5	Data6	Data7	Data8
Pitch (deg)	Roll (deg)	Heading (true)	Heading (mag)	Mavar (deg)	Bug (mag)	0	0

Received data in **Simulink**

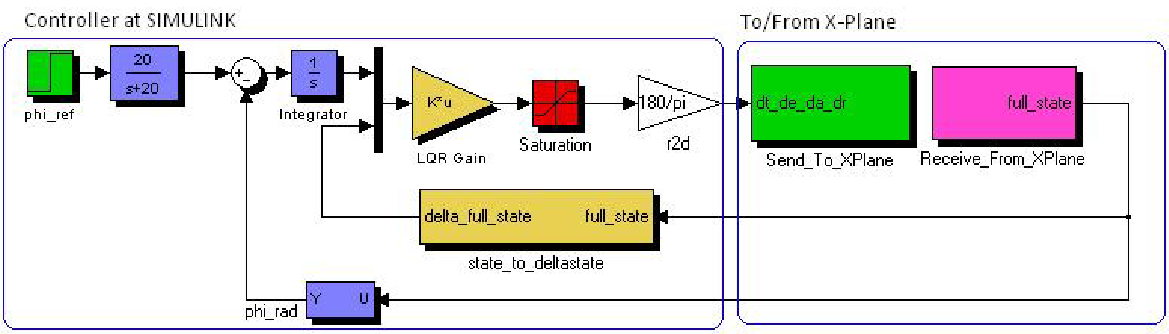
Table 2-2 Received Data Table

	D	A	T	A	N	i	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Data8
Byte	1	1	1	1	1	4	4	4	4	4	4	4	4	4

Picture 2-4 is a data packed of X-Plane. It has 8 data. It described in table 2-1 and detailed in Table 2-2. Data 7 and data 8 gives nothing, but they still had to be counted up to have the number of bytes. These bytes total is 41. Thus, 'output port width' in UDP Receive Block's Parameters is set to 41. And yet, we have to make sure that we put the right IP Port of X-Plane. So, we can put it UDP block's parameters.

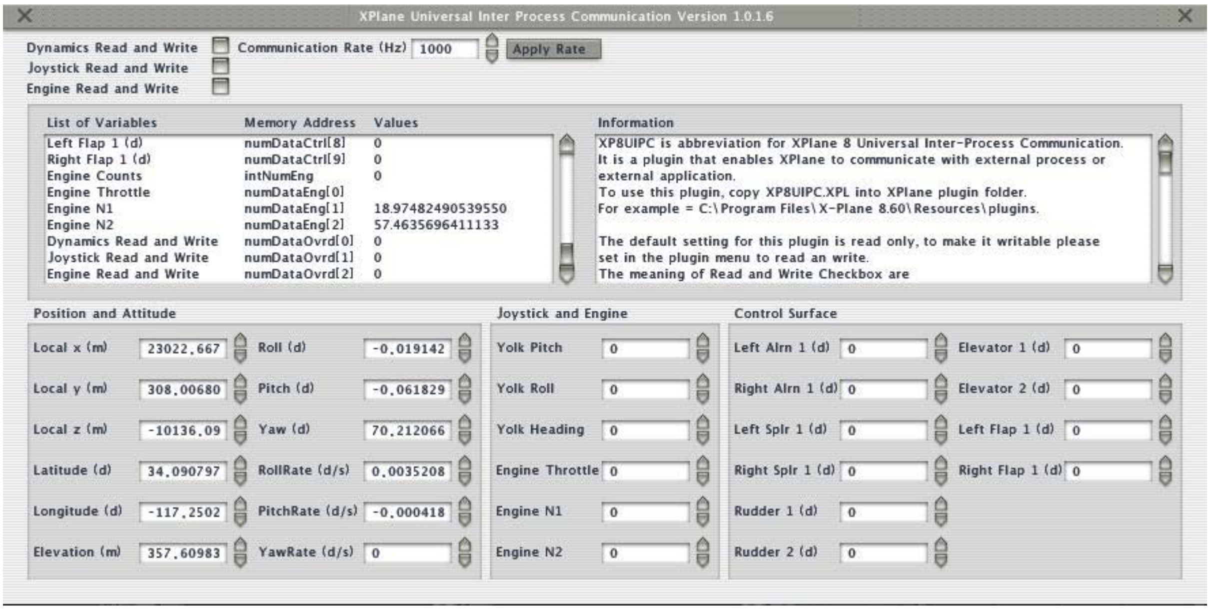
This UDP Communication was built in X-Plane version 8.4. Newer versions of X-Plane have different network communication requirements. They have different data format that are not suitable for this UDP Communication setting.

4. RESULTS AND ANALYSIS



Picture 3-1 Simulink model

This model shows the output data directly sent to the X-Plane through UDP. At the same time, the X-Plane also issued attitude and achievement data via UDP to SIMULINK model. Furthermore, these data will be passed back to the LQR-Gain to gain matrix K. This matrix K is very useful when we need to replace any SIMULINK model with a microcontroller.



Picture 3-2 X-Plane Universal Inter Process Communication

We need to build a plugins to make this system works. Plugins can run inside X-Plane. It can handle the flight simulator's capabilities. They can accomplish things that a standalone program might not be able to. Picture 3-2 is the plugin that can handle a few essentials property of flight simulation. This plugin can read data from simulator. For example, it can continually read the values of the flight instruments and send them over the network. It also can write data to the simulator, changing its actions or behavior. Various subsystems of the simulator can be controlled by this plugin.

5. CONCLUSION

The sofware in the loop for rocket simulation has been developed. This simulation system can be used to simulate and predict the flight dynamics of rocket. The simulation was generated by interactive communication between rocket model and rocket visualization in 3 dimension. The Rocket model was presented through Matlab/Simulink environment while its dynamics was visualized through XPlane. The communication between them was bridging by XPlane plugin protocol which was accessed from cache memory by Delphi program. This system can also be equipped by controller to evaluate the control algorithm and applied to the rocket flight. Rocket simulation based on software in the loop bring the advantage to reduce the high risk and high cost which might be happened during the flight test.

REFERENCES

- Jeff Lewis, ***X-Plane, UDP, and Visual Basic, for X-Plane version 6, 8 and 9***. Online Paper, 2011. (available online at http://www.jefflewis.net/XPlaneUDP_6.html, http://www.jefflewis.net/XPlaneUDP_8.html, http://www.jefflewis.net/XPlaneUDP_9.html)
- C. Rodney Hanke, "***The Simulation of a Jumbo Jet Transport Aircraft, Volume I: Mathematical Model***", NASA CR-1756, 1971.
- Nugroho, Larasmoyo, " ***SIJIDALANG – Sistem Uji Kendali Terbang***", Siptekgan 2008, Lembaga Penerbangan dan Antariksa Nasional, Jakarta, 2008

- Akbar, Ilham. Larasmoyo nugroho, " ***RANCANG BANGUN SISTEM SIMULASI UJI DARAT WAHANA TERBANG***", Siptekgan 2010, Lembaga Penerbangan dan Antariksa Nasional, Jakarta, 2010
- Roskam, Jan, " ***AIRPLANE AERODYNAMICS AND PERFORMANCE***", Darcorporation, Kansas USA, 1997
- Allerton, David, " ***PRINCIPLES OF FLIGHT SIMULATION***", Wiley, West Sussex UK, 2009
- Lucio R. Ribeiro and Neusa Maria F. Oliveira, ***UAV Autopilot Controllers Test Platform Using Matlab/Simulink and X-Plane***, 40th ASEE/IEEE Frontiers in Education Conference, Washington, DC, October 27 - 30, 2010