



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

# IJIRSET

International Journal of Innovative Research in  
**SCIENCE | ENGINEERING | TECHNOLOGY**



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Issue 1, January 2023

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA

**Impact Factor: 8.118**

# Design and Development of An Open Source Flight Simulator

Dhairya Dosi<sup>1</sup>, Aman Kumar<sup>2</sup>

P.G. Student, Dept. of Aerospace engineering, Indian Institute of Technology, Chennai, India<sup>1</sup>

U.G. Student Dept. of Computer Science engineering, Chandigarh University, Gharuan, India<sup>2</sup>

**ABSTRACT:** The flight simulation is the process in which a pilot is trained to fly the aircraft without being airborne with the help of computers. It simulates the flight conditions using the equations of motion. i.e. it basically mimics the aircraft motion and tries to make the pilot feel that he is flying without actually risking his dear life and millions of dollars.

The problem with current flight simulators is that it's expensive and can not be customized without additionally paying a hefty amount of money to the development company. The solution to the given problem is an open source flight simulator. In which you can define your own flight model based on data fed into the system.

It can be used to compliment flight testing as it works on the parameters defined during the aircraft design. It means you can also test the validity of your aircraft design before actually building it. So it can be added as an optional process in aircraft design.

The system enables you to make an artificial cockpit with control-stick, throttle-lever, foot-paddles and all other necessary switches and knobs. It supports 4DOF seat motion to generate the so called feel of the flight. as aircraft yaws pitches and rolls the pilot seat moves too.

The core of the simulator is the programme that solves 6DOF coupled differential equations of motion 100 times a second and the solution gives linear and angular positions that are rendered in 3D gamespace.

so the tool can help student pilots, enthusiasts, and gamers feel the dynamic behaviour of their hypothetically designed aircrafts.

**KEYWORDS:** flight simulation, flight dynamics, flight testing, aircraft design

## I. INTRODUCTION

Flight simulation is the field of aviation that has been running parallel to the development of aircraft and spacecraft itself. The reason being helpful for the development, testing and training for these cutting edge machines. For development flight simulators are used to test control, stability and performance parameters. It is also a useful tool to check the feasibility of the aircraft design in its preliminary stages. But mainly they are used to give pilots the so-called "intuition" of the aircraft handling including understanding flight characteristics of the given aircraft. This results in enhancement of flight safety. not to mention it makes flight training safe and cheaper.

The flight simulator in general consists of 4 major parts

1. The flight computer
2. Visual system
3. Virtual cockpit
4. Motion platform

There are several other parts (Eg. Avionics) that falls under sub-category of these four.

Market is full of flight simulators that have these crucial components working in it these days. But they are too expensive to use. Also another disadvantage they offer is the limited number of aircraft to fly which offer no room for small changes in characteristics due to say slight perturbation in C.G. position arrived due to extra payload which changes a lot of things about flight handling.

The goal is to develop an open source flight simulator that allows users to add and edit each parameter of the aircraft including the aircraft itself.

The possible applications are

1. Understand the flight behaviour in design phase only
2. Validate the feasibility of the aircraft design

## II. FORMULATION

To simulate flight the computer must have the position vector of the aircraft in 3D cartesian space. The left hand 3D cartesian space is taken for use in the upcoming calculations with the origin at the center of the gamespace

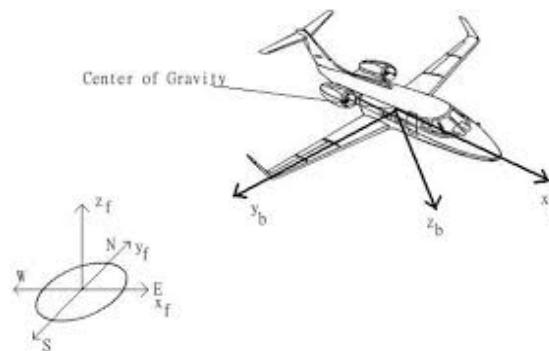


Figure 1 - coordinate system of the aircraft

To find out position vector (3 liner 3 angular) the equation of motion in each direction is used independently

Theoretically the forces and moments in/about those directions generate linear/angular acceleration in its respective direction. By integrating numerically twice with initial conditions given the position vector could be determined.

Primary objective of the following text is to define expressions for force and moment vectors in/about those directions.

### A. Forces

As a result of newton's second law the equation of motion can be derived using forces in 3 coordinates which says

$$\text{Force} = v \cdot (dm/dt) + m \cdot (dv/dt)$$

Integrating it for a rigid body assuming no change in mass and you get

$$\text{force} = \text{mass} \cdot \text{acceleration}$$

It can be written in matrix notation for 3 dimensions as

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & M \end{bmatrix} \cdot \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix}$$

eqn 1

Using aerodynamics the forces in 3 dimensions are as

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} T - QS \cdot (\partial C_d / \partial \alpha) \cdot \alpha - W \sin \theta \\ (QS \cdot (\partial C_l / \partial \alpha) \cdot \alpha) \cdot \cos \beta \\ (QS \cdot (\partial C_l / \partial \alpha) \cdot \alpha) \cdot \cos \beta - W \cos \theta \end{bmatrix}$$

eqn 2

Putting equation 2 in the force vector of equation 1 we get an acceleration vector.

**B. Maintaining the Integrity of the Specifications**

Newton's second law can also be applied on the context of angular motion just with different notations

$$\text{Moment} = \partial L / \partial t$$

By simplifying we get

$$\text{Moment} = I \cdot \partial^2 \Theta / \partial t^2$$

It can be written in matrix notation for 3 dimensions as

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{bmatrix} \begin{bmatrix} \alpha'' \\ \beta'' \\ \phi'' \end{bmatrix}$$

eqn 3

Using aerodynamics the moments in 3 dimensions can be defined as

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} QSC \left( \partial C_n / \partial \beta \cdot \beta + \partial C_n / \partial \beta' \cdot \beta' + \partial C_n / \partial \delta \cdot \delta + \partial C_n / \partial \delta' \cdot \delta' + C_{no} \right) \\ QSC \left( \partial C_m / \partial \alpha \cdot \alpha + \partial C_m / \partial \alpha' \cdot \alpha' + \partial C_m / \partial \delta \cdot \delta + \partial C_m / \partial \delta' \cdot \delta' + C_{mo} \right) \\ QSC \left( \partial C_l / \partial \phi \cdot \phi + \partial C_l / \partial \phi' \cdot \phi' + \partial C_l / \partial \delta \cdot \delta + \partial C_l / \partial \delta' \cdot \delta' + C_{lo} \right) \end{bmatrix}$$

eqn 4

And similar to force equation 2 has to be put in equation 1 to get an angular acceleration vector .

**C. Euler angles**

The linear accelerations we get are all in the body fixed axis system of the aircraft . but it has to transfer those accelerations into the local horizon axis. To do that rotation matrices are applied about all three axes to find projections of given accelerations on the local horizon which is parallel to the earth fixed inertial coordinate system.

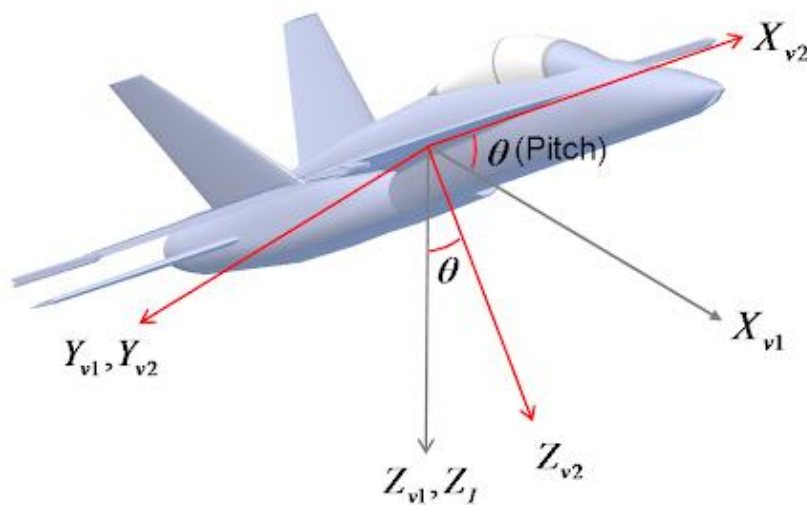


Figure 2 - euler angles on aircraft

$$\begin{bmatrix} \bar{F}_x \\ \bar{F}_y \\ \bar{F}_z \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \cdot \begin{bmatrix} \cos \beta & -\sin \beta & 0 \\ \sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha \\ 0 & 1 & 0 \\ \sin \alpha & 0 & \cos \alpha \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}$$

Equation 1 - Transformed forces

#### D. Unification of force and moment equations

To simplify the calculation for the computer the force and moment matrices are combined

$$\begin{bmatrix} \bar{F}_x \\ \bar{F}_y \\ \bar{F}_z \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} M & 0 & 0 & 0 & 0 & 0 \\ 0 & M & 0 & 0 & 0 & 0 \\ 0 & 0 & M & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x & -I_{xy} & -I_{xz} \\ 0 & 0 & 0 & -I_{yx} & I_y & -I_{yz} \\ 0 & 0 & 0 & -I_{zx} & -I_{zy} & I_z \end{bmatrix} \cdot \begin{bmatrix} A_x \\ A_y \\ A_z \\ \beta'' \\ \alpha'' \\ \phi'' \end{bmatrix}$$

Equation 1 - unified inertia matrix

The above equation can be generalized as

$$[F] = [I] \cdot [A]$$

With

$$|F| = (|\dot{x}|[C] + |x|[S] + |\delta|[C.D.] + |\delta|[C.R.D.]$$

Equation 2 - the unified forces and moment matrix

Where

[C] is damping coefficient matrix [S] is the stability derivative matrix [C.D.] is the control derivative matrix [C.R.D.] is control rate derivative matrix

#### E. Formation of a coupled linear differential equation

The final stage is to add all these together to find an expression for the acceleration vector

Using equation 1 & 2 from section D

$$|F| = |\ddot{x}|[I]$$

$$|\ddot{x}| = (|\dot{x}|[C] + |x|[S] + |\delta|[C.D.] + |\delta|[C.R.D.] ) * [I]^{-1}$$

The above equation can also be written as

$$|\dot{x}|[C] + |x|[S] + |\delta|[C.D.] + |\delta|[C.R.D.] + |\ddot{x}|[-I] = 0$$

Above equation is the expression to be explained to the computer to find out acceleration vector.

#### F. Computation scheme for the computer to find out position vector

The acceleration vector has to be numerically integrated with the help of UNITY game engine as the initial condition taken as everything to be zero with the frequency of 1000 per second (it would refresh the position of the aircraft every millisecond) which ensures high accuracy of position vector in flight simulator.

### III. MODELING

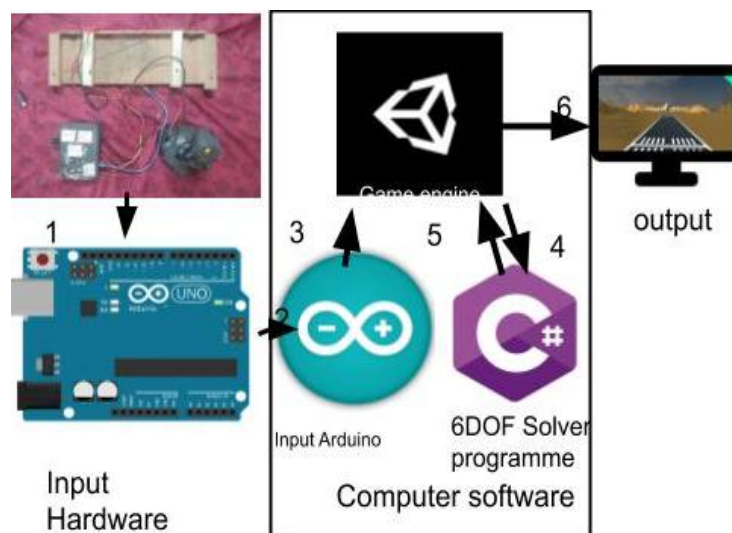
For implementing this data in a flight simulator the unity game engine was used. UNITY is a game engine by unity technologies. In which the script is written in C# programming language

For taking inputs arduino development platform is used because it's an open source platform along with thrustmaster joystick for control

The programme that runs the engine is written implementing relations and equations that are developed in the previous section of this article.

#### A. System architecture

the system architecture looks like this so the work was done accordingly.



#### B. Workflow

- 1. Physical switches and levers are connected to switches and potentiometers respectively and processes like analog to digital conversion or operational amplification (for buttons) are done in arduino and encoded in such a way that the computer understands the command.
- 2. The unity game engine receives the signal and processes into values that could be processed by 6DOF motion equations explained in section 2 . the programme gives the 6 coordinates back to the game engine (3 linear,3 angular) back to the game engine.
- 3. Game engine renders the position of these 6 coordinates and outputs a video to the screen.

#### C. Testing

One of the aircraft designs used for testing the flight simulator titled “pushpak viman” is a short haul air taxi with pertinent flight data mentioned below.



Figure 3 - pushpak viman

pushpak viman data			
wing properties		Horizontal Tail Properties	
Wingspan	10.3 M	Horizontal Tail Wingspan	4.255 M
Wing Planform Area	16.9 M <sup>2</sup>	Horizontal Tail Tip Chord Length	0.670 M
Mean Aerodynamic Chord (MAC)	1.667 M	Horizontal Tail Root Chord Length	1.40 M
Tip Chord Length	1.37 M	Leading Edge Horizontal Tail Sweep	12.5 DEG
Root Chord Length	1.9 M		
sweep_LE_deg	1.5 DEG		
Cruise Flight Conditions		Cruise Mass and Inertial Data	
Accelration due to gravity	-9.81 M/S <sup>2</sup>	Aircraft Weight	2884.8 Kg
Geometric Altitude	3000 M	Moment of Inertia (x-axis)	I <sub>xxB</sub> = 7985
True Airspeed	139 M/S	Moment of Inertia (y-axis)	I <sub>yyB</sub> = 3326
Dynamic Pressure	4438.5 Pa	Moment of Inertia (z-axis)	I <sub>zzB</sub> = 3326
		Product of Inertia (xz-plane)	I <sub>xzB</sub> = 0

Longitudinal Aerodynamic Coefficients		
Longitudinal Steady State Coefficients	Longitudinal Stability Derivatives	Longitudinal Control Derivatives
CL1 = 0.378;	CD0 = 0.02;	CD_deltaE = 0;
CD1 = 0.03;	CDu = 0;	CL_deltaE = 0.5;

CM1 = 0;	CDalpha = 0.25;	CM_deltaE = -1.12;
CTx1 = 0.03;	CTxu = -0.07;	
CMT1 = 0;	CL0 = 0.2;	
	CLu = 0;	
	CLalpha = 5.15;	
	CL_alphaDot = 2.0;	
	CLq = 4.1;	
	CM0 = 0.025;	
	CMu = 0;	
	CMalpha = -0.7;	
	CM_alphaDot = -6.95;	
	CMq = -14.9;	
	CMTu = 0;	
	CMTalpha = 0;	

Lateral Directional Aerodynamic Coefficients		
Lateral Directional Stability Derivatives	Lateral Directional Stability Derivatives	Lateral Directional Control Derivatives
Clbeta = -0.0944;	Clbeta = -0.0944;	Cl_deltaA = 0.181;
Clp = -0.442;	Clp = -0.442;	Cl_deltaR = 0.015;
Clr = 0.0926;	Clr = 0.0926;	Cy_deltaA = 0;
	CYbeta = -0.346;	Cy_deltaR = 0.2;
	CYp = -0.0827;	Cn_deltaA = -0.0254;
	CYr = 0.3;	Cn_deltaR = -0.0365;
	Cnbeta = 0.1106;	
	CnTbeta = 0;	
	Cnp = -0.0243;	
	Cnr = -0.139;	

The data given above is used to test the aircraft flight properties in flight simulator.

#### IV. RESULT AND CONCLUSION

The aircraft runs perfectly fine and flight properties stay within the calculation. A couple of screenshots of the programme given below



Figure 4 - pushpak viman model after takeoff



Figure 5 - game view

#### A. Results

*The project is a developmental project where the coupled linear mathematical equations are being solved. the problem is we do not have the actual figures to compare with. basically the whole flight simulation is all about solving a huge numerical problem 100 times a second. As it is over I can conclude that those differential equations are quite effective in getting approximately correct results. those are correct is the only reason i am finding those equations in the textbook.*

#### B. Conclusion

we have a possibility now to conclude that a designed aircraft will fly or not if yes how would it handle it. Will it cause access fatigue to the pilot? Is it too stiff or too unstable? We are able to find that out without paying any money, without making a real aircraft risking your dear life. open source projects in general have a lot of possibilities.

#### ACKNOWLEDGMENT

This project, though done by us would not have been possible, without the support of various people, who by their cooperation have helped us in bringing out this project successfully.

#### REFERENCES

- [1] Arthur Bryson & Yu-Chi Ho, Applied Optimal Control: Optimization, Estimation, and Control, Hemisphere, 1975.
- [2] Bernard Etkin & Lloyd Duff Reid, Dynamics of Flight, Stability and Control, McGraw- Hill, Third Edition, 1996.
- [3] Frances B. Hildebrand, Advanced Calculus for Applications, Prentice-Hall, 1962.
- [4] Robert C. Nelson, Aircraft Stability and Automatic Control, McGraw-Hill, Second edition, 1998.
- [5] Louis V. Schmidt, Introduction to Aircraft Flight Dynamics, AIAA Education Series, 1998.
- [6] Jaime A. Ramirez Durate, Feasibility study of application of open source autopilot simulator, October 2018
- [7] Emad eveid, Martin skriver, Kristin husum, ulrik pagh shultz, A survey of open source flight controller and flight simulators, May 2018
- [8] Mansoor Ahsan, Farrukh mahzar, Development of a generic flight simulator for a fixed wing aircraft, November 2013



**INNO SPACE**  
SJIF Scientific Journal Impact Factor  
**Impact Factor: 8.118**



**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
**INDIA**



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



**www.ijirset.com**

Scan to save the contact details