

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/236174210>

# Automatic control education using FlightGear and MATLAB based virtual lab

**Conference Paper** in The International Conference on Electrical Engineering · May 2012

DOI: 10.21608/iceeng.2012.31372

CITATIONS

4

READS

5,735

5 authors, including:



**Mohammed Moness**

Minia University

88 PUBLICATIONS 519 CITATIONS

SEE PROFILE



**Ahmed Mahmoud Moustafa**

Minia University

26 PUBLICATIONS 333 CITATIONS

SEE PROFILE



**Mohammed Sayed**

Cincinnati Children's Hospital Medical Center

17 PUBLICATIONS 1,131 CITATIONS

SEE PROFILE

**Military Technical College  
Kobry El-Kobbah,  
Cairo, Egypt**



**8<sup>th</sup> International Conference  
on Electrical Engineering  
ICEENG 2012**

## **Automatic control education using FlightGear and MATLAB based virtual lab**

*By*

M. Moness\*   A. M. Mostafa\*   M. A. Abdel-Fadeel\*   A. I. Aly\*   A. Al-Shamandy\*

### **Abstract:**

In this paper, a virtual lab based on MATLAB and FlightGear flight simulator is discussed and developed. The virtual lab consists of number of experiments that enhances the understanding of fundamental concepts of classical control for undergraduate students. The objective of the virtual lab is to design a stability augmentation system for C310 aircraft model according to known flying qualities then implement a hold autopilot for the aircraft. This objective is achieved through a series of experiments, animation and simulation to illustrate main concepts of classical control like modeling, time domain analysis, frequency domain Analysis, non-minimum phase systems, root locus, feedback design and PID compensation. These concepts are demonstrated in four experiments through the analysis and design of a flight control system for Cessna 310 aircraft on FlightGear flight simulator. The developed virtual lab can be used easily by students to study and visualize classical control principles using FlightGear and MATLAB GUI with an attractive case study of a flight control system.

### **Keywords:**

Virtual lab, automatic control education, flight simulator and FlightGear

\* Faculty of Engineering, Minia University, Minia, Egypt

## **1. Introduction:**

A flight simulator represents a field of virtual engineering that tries to copy or to simulate the experience of flying an aircraft as realistic as possible. Modeling and simulation are important tools of virtual engineering. In this paper, we introduce a new technique for teaching control engineering with flight simulators. We build a virtual lab using MATLAB and the FlightGear flight simulator which is an open-source, multi-platform, cooperative flight simulator development project. The tasks of virtual engineering in flight systems – as stated by Zipfel [8]– are:

1. Developing performance requirements.
2. Guiding and validating designs.
3. Test support.
4. Reducing test cost.
5. Investigating inaccessible environments.
6. Pilot and operator training.
7. Practicing dangerous procedures.
8. Gaining insight into flight dynamics.
9. Integrating components.
10. Entertainment.

We introduce a new task to the previous tasks:

11. Control engineering education.

The FlightGear flight simulator implements number of flight dynamics models (FDM) libraries like: LaRCsim, YASim, UUIC, and JSBSim [6]. This paper aims to develop a virtual lab based on a flight control system for a JSBSim aircraft model that is compatible with FlightGear. JSBSim is an open source flight dynamics model software library that models the flight dynamics of a 6-DoF aerospace vehicle. The designed virtual lab will use the flight control system to serve as a fun, rich and interesting case study for teaching control engineering. This case study will be organized in a lab course which includes fundamental concepts for introductory automatic control courses and advanced concepts for higher students. These concepts will be visualized in a virtual reality environment based on a flight simulator.

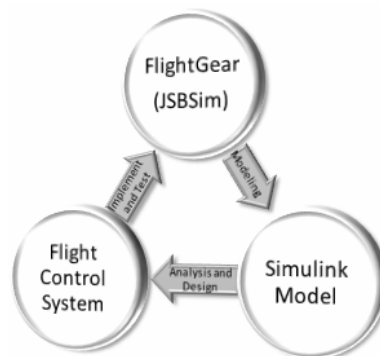
There are previous implementations of flight simulators for education. Bossert and Yechout [2] used MATLAB with Genesis 3000 flight simulator for flight mechanics education. Munz et al. [10] developed a spacecraft simulator for advanced automatic control education. Pavel and Martin [9] proposed a course implemented over an embedded graphics system simulator for teaching Board Information Systems (BIS) course.

## **2. Virtual Lab Architecture:**

The virtual lab will be based on three components: a flight simulator virtual cockpit for visual flight experiments, a powerful control system analysis and design software and a lab manual that contains control experiments. This virtual lab is aimed to serve for a wide range of students studying automatic control course in Computer, Electrical and Mechanical Engineering.

We will use the FlightGear flight simulator based on the JSBSim FDM library and a control systems analysis tool like MATLAB and Simulink. In the following points, we will introduce the role of each component to the education process (Figure 1):

1. *FlightGear Flight Simulator*: Represents the virtual reality environment where the three dimensions enhance the student understanding by visualizing flight designed experiments.
2. *JSBSim FDM*: This flight dynamics library is developed by a number of aerospace engineers, software engineers and mathematicians [5]. It plays the role of the plant that needs to be analyzed, modeled and controlled.
3. *MATLAB and Simulink*: Engineering software where analysis and control design is performed.



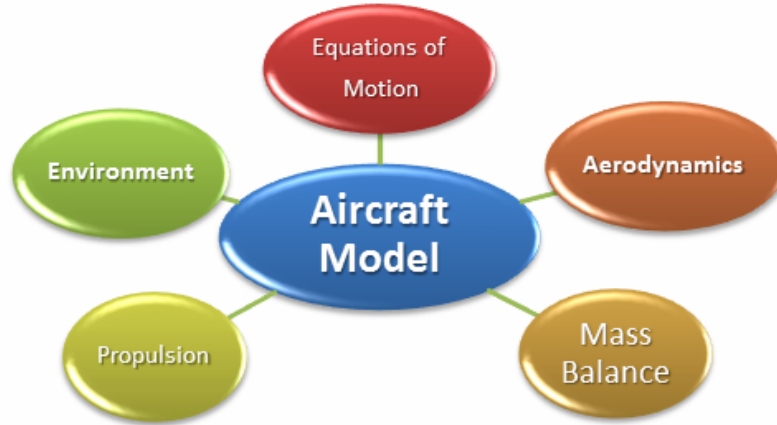
***Figure (1): Control Education Process and Components***

This virtual lab links between a ready-made control tool and a fancy graphical representation as suggested by Munz et al. [10] for choosing a platform for education control games.

## **3. The Cessna 310 Aircraft Model:**

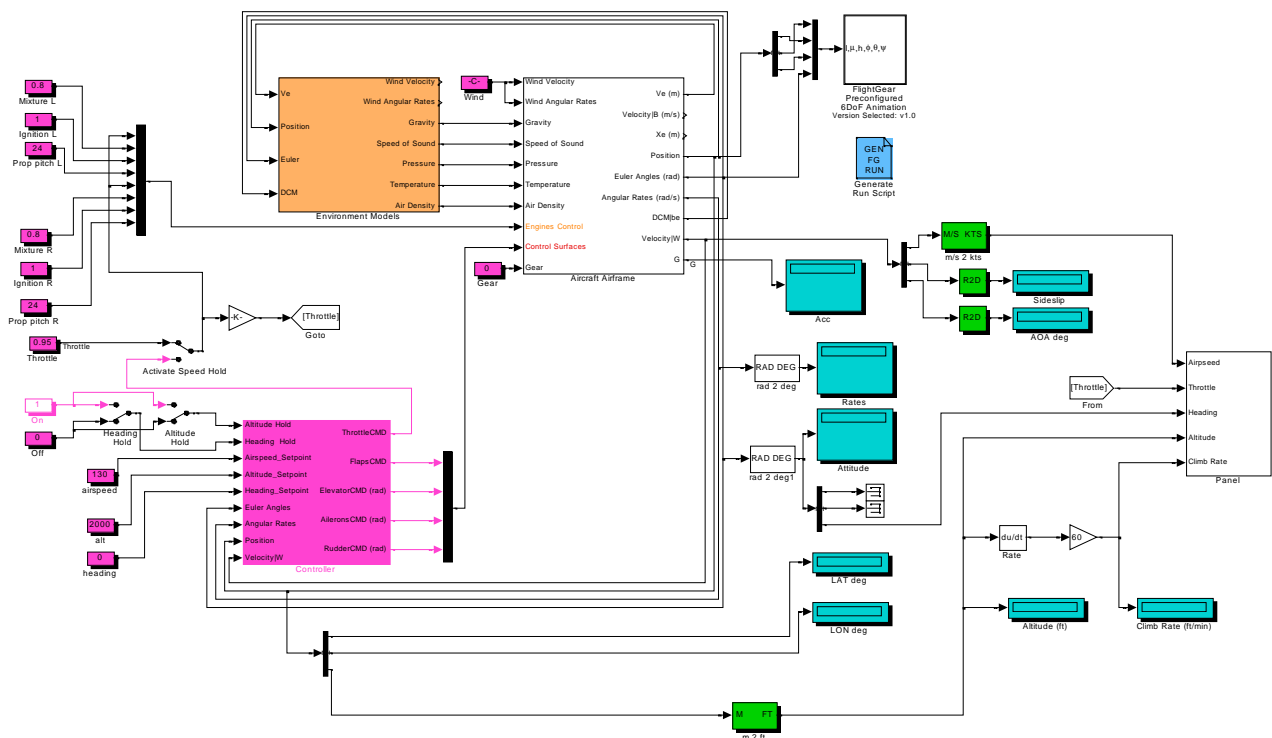
The aircraft model consists of number of subsystems. Each subsystem is responsible for generating the forces, moments and other physical variables that affects the aircraft

dynamics. Our aircraft model has five main subsystems as illustrated in Figure 2.



**Figure (2): Aircraft Model Subsystems**

The aircraft model is implemented in Simulink with the help of Simulink Aerospace Blockset and Aerosim Blockset by Unmanned Dynamics [11]. (Figure 3)



**Figure (3): C310 Aircraft Simulink Model**

#### 4. Wings Level Operating Point of the Aircraft Model:

We need to put the aircraft to equilibrium or a steady state. Then, we can analyze the dynamics of the aircraft by perturbing certain parameters. The aircraft model is initialized with the conditions values and constraints of the steady state wings level. We use the gradient descent with elimination algorithm to perform an optimization to get an operating point by using *Simulink Control Design* and *MATLAB Control System Toolbox*. The operating point is computed using Tables 1 and 2.

**Table (1):** Internal state values at wings level operating point

State	Value	dx
Fuel Tank 1 Mass	106.8395	-0.00363
Fuel Tank 2 Mass	112.962	-0.00363
Fuel Tank 3 Mass	50.8198	-0.00363
Fuel Tank 4 Mass	52.0338	-0.00363
Engine 1 Speed	197.166	-1.6975e-10
Engine 2 Speed	197.166	-1.7483e-10
$\phi$	0	0
$\theta$	0.047634	0
$\psi$	1.1593e-13	0
P	0	8.3897e-15
Q	0	-3.948e-13
R	0	3.7523e-13
U	63.7578	5.5257e-12
V	0	2.6964e-22
W	3.0394	4.5946e-14
$x_e$	-3.2298e-12	63.8302
$y_e$	5.1201e-14	7.3996e-12
$z_e$	-2251.3711	8.8818e-15

**Table (2):** Inputs values at wings level operating point

Inputs	Initial Value
$\delta_{th}$	0.82315
$\delta_e$	-2.2408e-20
$\delta_a$	0.2913
$\delta_r$	3.4323e-21

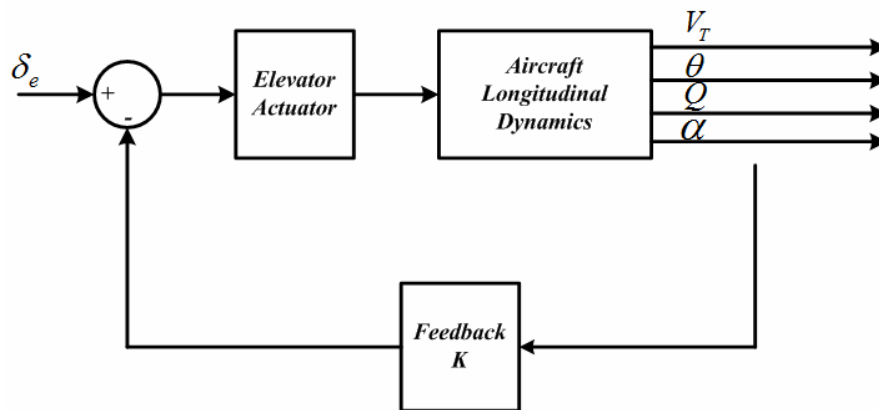
At this operating point, the model can be linearized to get both the aircraft longitudinal and lateral dynamics model.

### 5. Stability Augmentation System (SAS):

The feedback loops that force aircraft oscillation modes to the flying qualities are called stability augmentation system (SAS). SAS is essential to achieve the aircraft dynamic stability. SAS is separately designed for the longitudinal and the lateral dynamics depending on the decoupling of the aircraft dynamics as illustrated by Stevens and Lewis [1], Blakelock [3], and Roskam [4].

#### 5.1. Longitudinal Stability Augmentation System:

The main purpose of the longitudinal SAS is to give satisfactory natural frequency and damping ratio for short period mode and to pull the short period poles from the over damped region to the under damped region with frequency of 0.4 to 0.6 rad/sec suggested by the flying qualities described in [1] and [7]. Possible feedbacks for longitudinal SAS are illustrated in Figure 4.



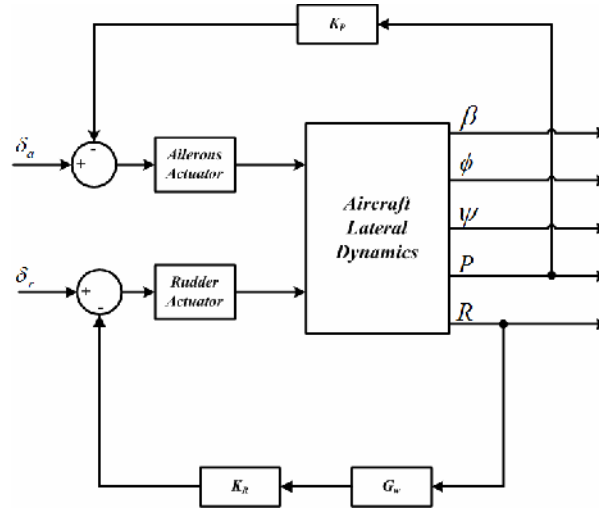
**Figure (4): Longitudinal SAS Possible Feedbacks**

#### 5.2. Lateral Stability Augmentation System:

The lateral SAS is intended to perform the following:

1. The roll subsidence mode and spiral mode time constants meet the flying qualities.
2. A roll damper is needed to eliminate the Dutch roll mode from the ailerons to roll rate transfer function and speed up roll rate.
3. A yaw damper is needed to generate an opposite yawing moment to eliminate any yaw that builds up from the Dutch roll mode.

Figure 5 shows the block diagram of lateral directional stability feedback system:



**Figure (5): Lateral SAS Block Diagram**

## 6. Hold Autopilot Design:

The autopilot is usually designed to meet specifications of steady state error and disturbance rejection with less concentration on dynamic response. The hold autopilot functions can be divided for the aircraft with respect to its dynamics into longitudinal hold autopilots and lateral hold autopilots. (**Error! Reference source not found.**)

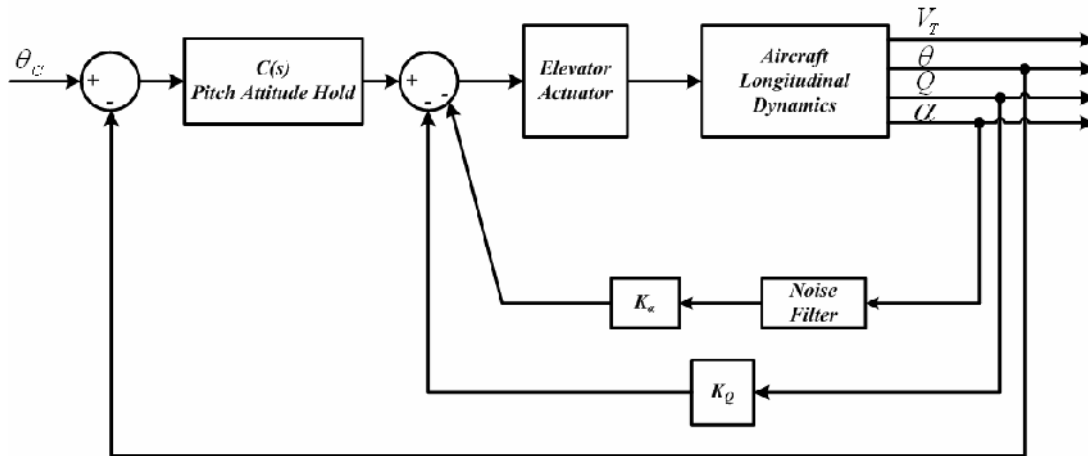
**Table (3): Hold Autopilots for C310**

Longitudinal Hold	Lateral Hold
1. Pitch Attitude Hold	1. Bank Angle Hold
2. Altitude Hold	2. Heading Angle Hold
3. Airspeed Hold	

### 6.1. Pitch Attitude Hold:

The purpose of pitch attitude hold autopilot is to make the pitch attitude angle constant at certain value determined by the pilot. It is rarely used alone. The pitch attitude hold autopilot is an internal autopilot mode where other autopilots can be closed around like the airspeed hold autopilot and the altitude hold autopilot. The pitch attitude hold autopilot compensator is closed around the pitch axis stability augmentation systems. The final block diagram of the pitch attitude hold autopilot is shown in Figure 6:

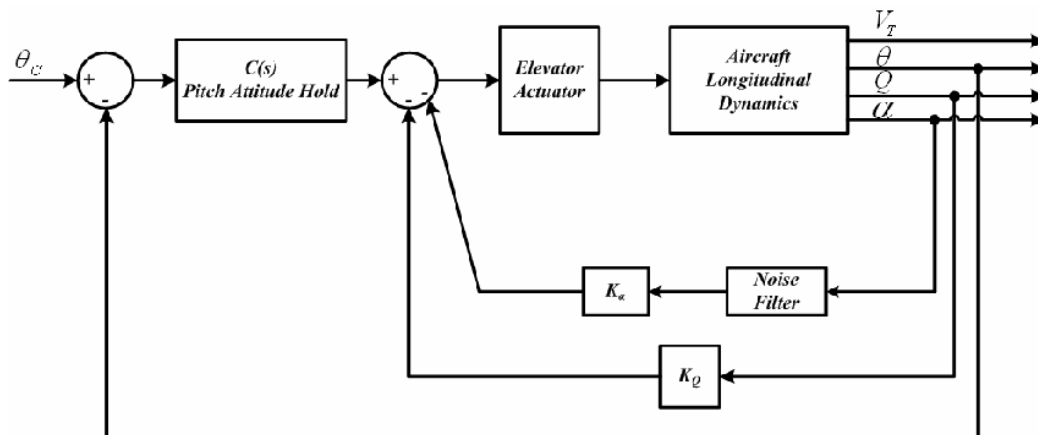




**Figure (6):** Pitch Attitude Hold Autopilot Block Diagram

## 6.2. Altitude Hold:

The main objective of the altitude hold autopilot is to keep the aircraft flying at an altitude set point defined by the pilot. It also needs to consider the aircraft maximum climb rate and maximum descend rate of the aircraft when reaching a set point. The altitude hold can be built around either the pitch attitude hold or the airspeed hold. In our design, we will build the altitude hold autopilot around the pitch hold. The final block diagram of the altitude hold autopilot is shown in Figure 7.



**Figure (7):** Altitude Hold Autopilot Block Diagram

## 6.3. Airspeed Hold:

The main objective of the airspeed hold autopilot is to keep the aircraft flying at a constant airspeed defined by the pilot. The airspeed hold can be built in two ways. It can

be either around the pitch attitude hold autopilot or by closing the loop from the airspeed to throttle. In our design, we chose the second scheme. The final airspeed autopilot block diagram is shown in Figure 8.

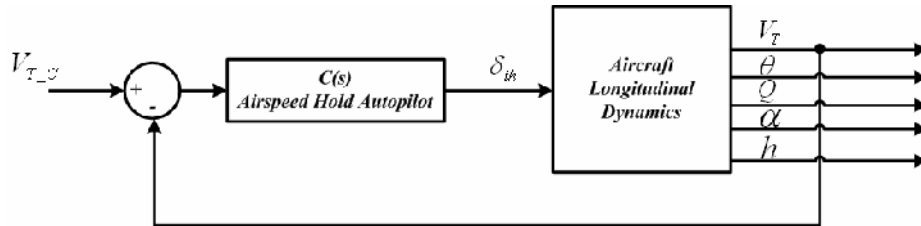


Figure (8): Airspeed Hold Autopilot Block Diagram

#### 6.4. Bank Angle:

The purpose of bank angle hold autopilot is to make the bank angle constant at certain value determined by the pilot. It is rarely used alone except for wings level hold. The bank angle hold autopilot is an internal autopilot mode where we will close the heading hold autopilot around. The bank angle will be closed around the lateral directional stability augmentation system. We will use the ailerons input and set the rudder input to zero. These concepts are illustrated by Figure 9.

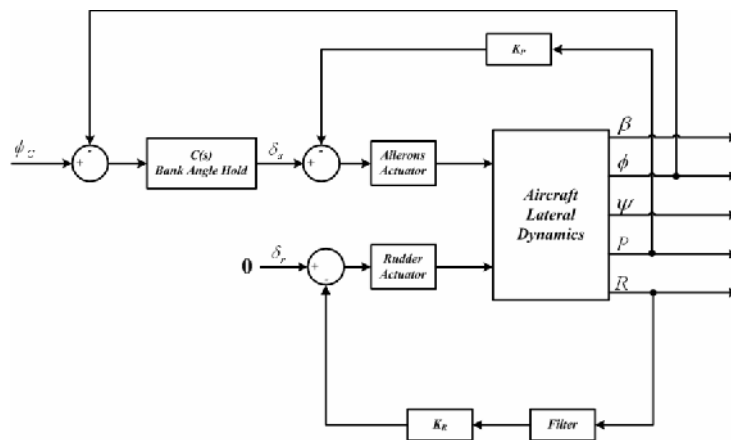
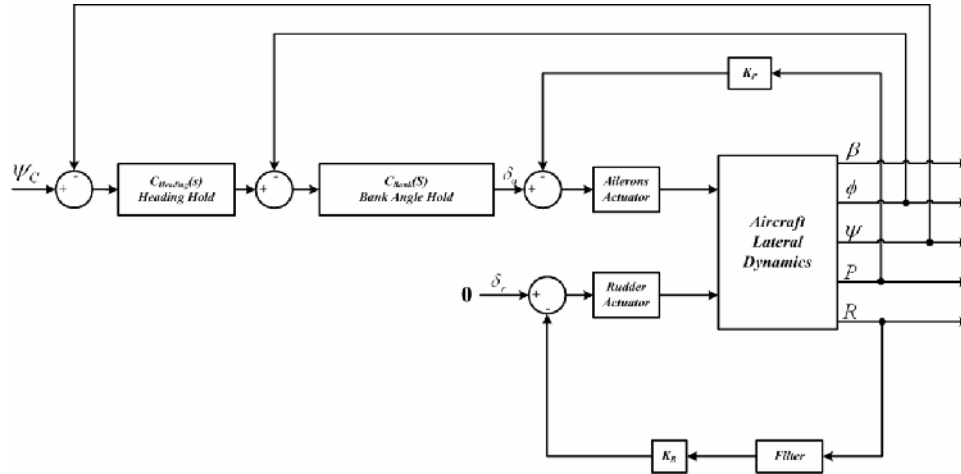


Figure (9): Bank Angle Hold Autopilot Block Diagram

#### 6.5. Heading Hold:

The main objective of the heading hold autopilot is to keep the aircraft at a heading set point defined by the pilot. The heading hold autopilot will be closed around the bank angle hold autopilot with input of reference heading angle ( $\psi_c$ ) and output yaw angle.

Figure 10 shows the block diagram of the heading hold autopilot.



**Figure (10):**Heading Hold Autopilot Block Diagram

## 7. Design of Flight Control System Virtual Lab:

The proposed design of the virtual lab consists of 4 experiments: Linearization of the nonlinear aircraft model, analysis of the dynamic modes of the aircraft and the flying qualities, feedback design of the stability augmentation system and PID controller design for the hold autopilots.

### 7.1. Linearization Experiment:

The first experiment to be illustrated is linearization. A nonlinear system such as an aircraft has multiple inputs and outputs. A nonlinear system needs to be linearized before it can be handled using classical control analysis and design. The GUI window of this experiment is shown in Figure 11.

As shown in **Error! Reference source not found.** the nonlinear and linearized models are close with small error at the same operating point using wings level operating point, but after changing the operating point into steady turning the output of the linearized model has a higher error.

### 7.2. Analysis of the Aircraft Dynamics Experiment:

The second experiment in the virtual lab is the dynamics analysis (Figure 12). This experiment aims to illustrate the transfer functions response in time domain and frequency domain and also demonstrates the dynamic modes of the aircraft in both longitudinal and lateral modes. The last part in the experiment is about the flying qualities and different modes, levels and flight phase category representing the design targets for the stability augmentation system.

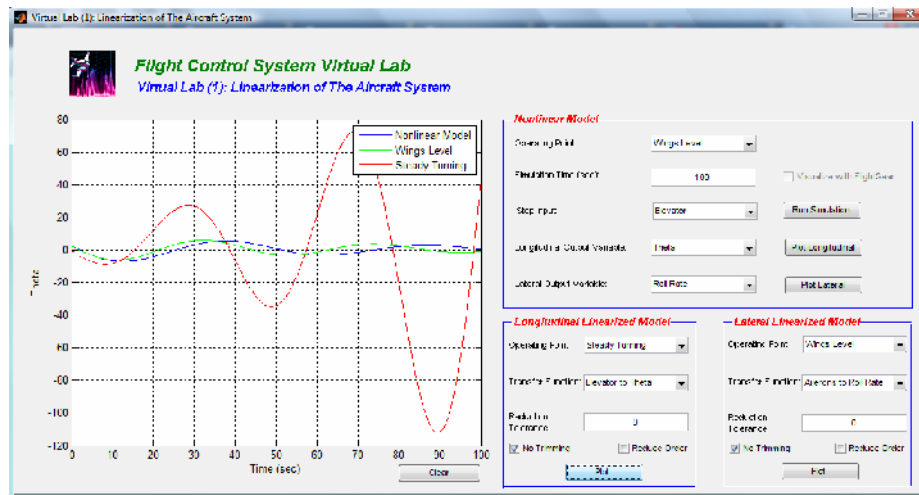


Figure (11): Linearization Experiment GUI

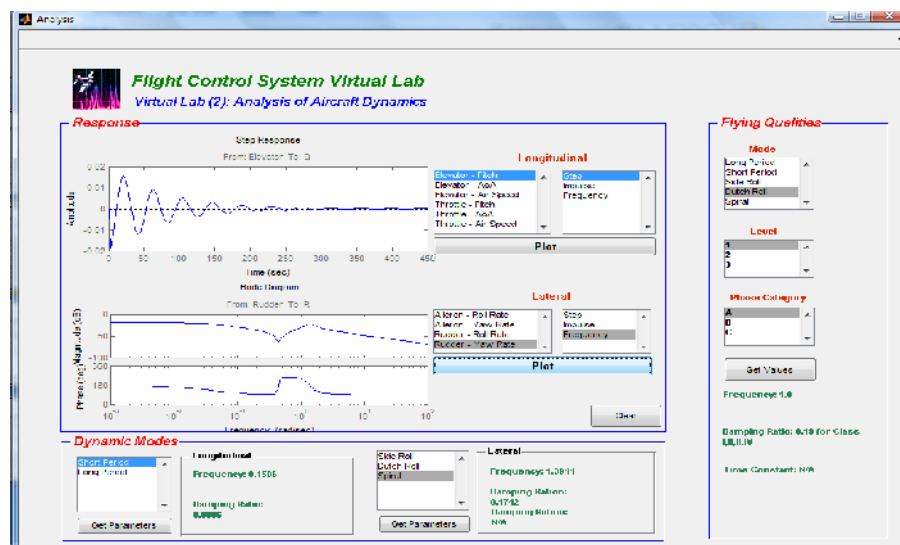


Figure (12): Analysis of Aircraft Dynamics Experiment GUI

### 7.3. Feedback Design for Stability Augmentation System Experiment:

The feedback design for stability augmentation experiment is designed to apply the feedback required and monitor the effect on the step response and root locus design. The feedback is applied to shape the dynamics of both the longitudinal and lateral modes to get the targeted flying qualities of the aircraft. Figure 13 shows the all GUI components of this experiment.

#### 7.4. PID Design for Hold Autopilots Experiment:

After the linearization, analysis and stability augmentation processes, the last phase in is to design the autopilot controller. This experiment (Figure 14) illustrates the PID controller and to maintain hold functions for some aircraft parameters like pitch attitude, altitude, bank angle, heading and airspeed.

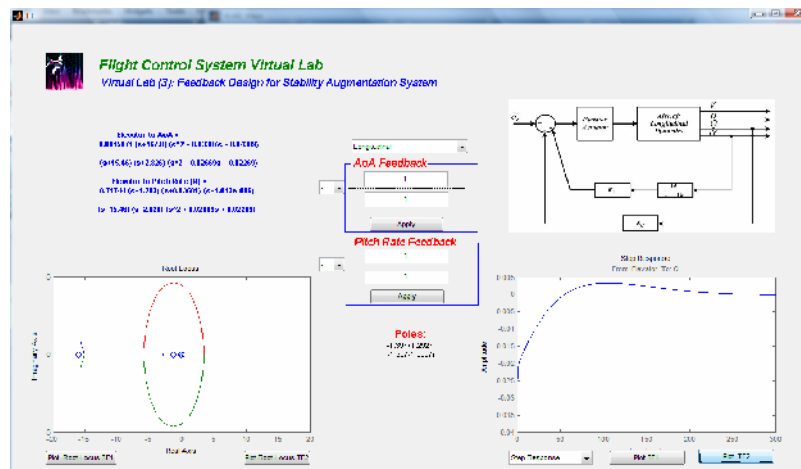
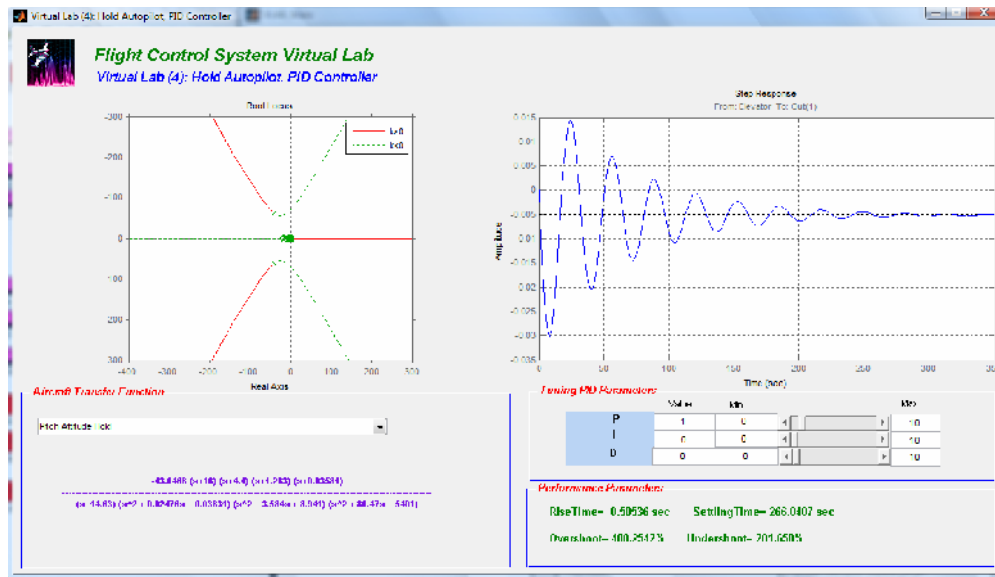


Figure (13): Feedback Design for SAS Experiment GUI



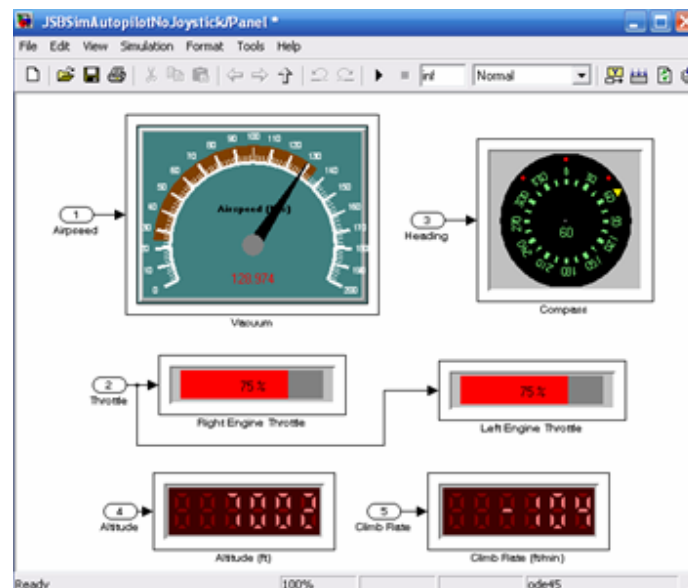
**Figure (14):PID Design for Hold Autopilot Experiment GUI**

## 7.5. Visualization of the Experiments:

The SAS and hold autopilot experiments can be visualized by linking the C310 aircraft *Simulink* model to the FlightGear flight simulator to give an experience of reality as shown in Figure 15 and Figure 16. The FlightGear flight simulator provides a rich and good graphical representation and animation. The graphical representation and experience of reality were problems in SpacecraftRT developed by Munz et al. [10].



**Figure (15):FlightGear Visualization of the C310 Aircraft and Panel**



**Figure (16):** Simplified Flight Panel in Simulink

## **8. Conclusion:**

This paper has implemented four experiments for control engineering education. The four control experiments were built as MATLAB GUIs with link to FlightGear Flight Simulator for visualization. The four experiments discussed: Linearization and operating points, analysis in time domain and frequency domain, feedback design by root locus, and PID Control. Discussion, analysis and design of a flight control system of a FlightGear aircraft show that it can be a great tool with fun and wealthy examples for control problems.

## **References:**

- [1] B. L. Stevens, and F. L. Lewis, *Aircraft Control and Simulation*, John Wiley and Sons, 1992.
- [2] D. E. Bossert and T. R. Yechout, *A MATLAB-Based Flight Simulator for Flight Mechanics Education*, AIAA Atmospheric Flight Mechanics Conference and Exhibit, AIAA 2002-4876, August 2002.
- [3] J. H. Blakelock, *Automatic Control of Aircraft and Missiles*, 2<sup>nd</sup> Edition, John Wiley and Sons, 1991.
- [4] J. Roskam, *Airplane Flight Dynamics and Automatic Flight Control*, DARcorporation, 2001.
- [5] J. S. Berndt and the JSBSim Development Team, *JSBSim: An open-source, Platform-independent, Flight Dynamics Model in C++*, <http://www.JSBSim.org>, 2008.

- [6] M. Basler, M. Spott, S. Buchanan, J. Berndt, B. Buckel, C. Moore, C. Olson, D. Perry, M. Selig, D. Walisser et al, *The FlightGear Manual*, <http://www.flightgear.org/>, 2008.
- [7] M. V. Cook, *Flight Dynamics Principles, A Linear Approach to Aircraft Stability and Control*, 2<sup>nd</sup> Edition, Elsevier, 2007.
- [8] P. H. Zipfel, *Modeling and Simulation of Aerospace Vehicle Dynamics*, AIAA, 2007.
- [9] P. Pavel and S. Martin, *Introducing Students to Aerospace Board Information Systems, Using an Embedded Graphics Systems Simulator*, 10<sup>th</sup> IEEE International Conference on Advanced Learning Technologies (ICALT), pp. 397–399, July 2010.
- [10] U. Munz, C. Bohm, J. Ech, M. Reble, P. Schumm and F. Allgower, *A Matlab-Based Game for Advanced Automatic Control Education*. Stuttgart Research Centre for Simulation Technology (SRC SimTech) Issue No. 2009-40, October 2009.
- [11] Unmanned Dynamics, *Aerosim: Aeronautical Simulation Blockset v1.2 User's Guide*, <http://www.u-dynamics.com/aerosim/>, 2005.

### Nomenclatures:

$\alpha$ ...	Angle of attack ( <i>rad</i> )
$\delta_a$ ...	Ailerons Deflection
$\delta_e$ ...	Elevator Deflection
$\delta_r$ ...	Rudder Deflection
$\delta_{th}$ ...	Throttles Input
$\Theta$ ...	Roll Angle ( <i>rad</i> )
$\Psi$ ...	Pitch Angle ( <i>rad</i> )
$\Phi$ ...	Yaw Angle( <i>rad</i> )
P ...	Roll Rate ( <i>rad/s</i> )
Q ...	Pitch Rate ( <i>rad/s</i> )
R ...	Yaw Rate ( <i>rad/s</i> )
U ...	Axial Velocity ( <i>m/s</i> )
V ...	Lateral Velocity ( <i>m/s</i> )
$V_T$ ...	Airspeed ( <i>knots</i> )
W ...	Normal Velocity ( <i>m/s</i> )
$x_e$ ...	X coordinate in Flat Earth Frame ( <i>m</i> )
$y_e$ ...	Y coordinate in Flat Earth Frame ( <i>m</i> )
$z_e$ ...	Z coordinate in Flat Earth Frame ( <i>m</i> )