Abstract. We modelled an aircraft based on the Airbus A320 and constructed a synthesisable flight control system. The novel feature was the use of C and VHDL, Very High Speed Integrated Circuit Design Language, to allow the flight control system to reside in a Field Programmable Gate Array in a model aircraft or an Uninhabited Aerial Vehicle. The simulator models axial, normal, transverse, pitch, roll and yaw movements. The flight control system has automatic manoeuvre envelope protection and constrained linear and angular accelerations. A graphical front end written in Java makes the flight control system very usable for aircraft engineers.

Introduction

The project objectives were to model an aircraft and construct a synthesisable flight control system. This paper describes the general approach and the novel conception for the simulation language of the flight control system. Similar systems are based on graphical simulation and never on hardware design or testing [1]. The novel feature of this project is the combined use of C and VHDL, Very High Speed Integrated Circuit Design Language [2], for system modelling so bridging the gap between software and hardware development. VHDL was chosen as the modelling language for the flight control system so that it could be synthesised onto a Field Programmable Gate Array (FPGA) and planted directly into a model aircraft or an Uninhabited Aerial Vehicle (UAV). Java provides a graphical front-end for the simulation. The developed simulator should model an aircraft with six degrees of movement: axial, normal, transverse, pitch, roll and yaw. Aircraft characteristics are based on fly-by-wire aircraft, like Airbus A320 [3]. Mechanisms such as servo systems and gearing influence the characteristics of the flight control system along with the autopilot system now installed on all modern passenger aircraft [4]. An important capability is the autothrottle feature which allows the pilot to control the velocity of the aircraft automatically. The system should include advanced features like passenger comfort through the limiting of linear and angular accelerations. Other essential features are environment and sensors subsystems. A graphical display module should provide an enhanced environment to analyse the results produced by the simulation.

System Design

Model Technology ModelSim [7] was used as the VHDL development and simulation environment. This package supports the integration of C and VHDL modules through the use of its Foreign Language Interface. This enabled the subsystems to be programmed in either language.

The overall system consists of five subsystems:

- Pilot subsystem
- Flight control subsystem
- Aircraft subsystem
- Environment subsystem
- Sensors subsystem

The supporting modules include the International Standard Atmosphere module [5] which converts environment variables such as atmospheric density, pressure and temperature into flight parameters such as altitude and axial velocity, and the graphical display module to analyse the results produced by the simulation. Figure 1 shows the flow of signals between the subsystems.

The sensors subsystem reads the environmental parameters that are realistically available to sensors on aircraft. These parameters include readings from accelerometers, gyroscopes, pressure and temperature gauges.

The aircraft subsystem is built upon these assumptions:

- The aircraft motions are not coupled unless stated otherwise.
- The wings have no dihedral or sweep.
- The total lift and drag are applied through the centre of gravity of the aircraft.
- The mass of the aircraft is constant.

The linear motion of the aircraft can now be determined using force equations and the linear equations of motion. Equations (1) and (2) are examples.

The environment subsystem models the local environment of the aircraft. It reads the linear and rotational acceleration, velocity and position parameters from the aircraft subsystem. The International Standard Atmosphere (ISA) model describes the air temperature, pressure and density variations with respect to the altitude. The model is calibrated for temperate latitudes and consequently has inherent error. Nevertheless, the model is used in modern passenger aircraft to calculate approximate velocity, altitude and Mach number.

At the altitude of 11 km the stratosphere starts and the temperature remains constant at 217 K.

[Programming Language]

TISA: International Standard Atmosphere

![Subsystem signal flow diagram](image)

The hydrostatic equation, perfect gas law and the lapse rate equation are:

\[
\frac{dp}{dh} = \rho \cdot g \\
p = \rho \cdot R \cdot T \\
T = T_0 - L \cdot h
\]  

(1)

\[p\] - Pressure (Pa), \(T\) - Temperature (K), \(\rho\) - Density (kg/m\(^3\)), \(p_0\) - The standard sea-level pressure (101325 Pa), \(T_0\) - The standard sea-level temperature (288.15 K), \(\rho_0\) - The standard sea-level density (1.225 kg/m\(^3\)), \(q\) - Dynamic pressure (kg/s\(^2\)), \(h\) - Altitude above sea level (m), \(V\) - Velocity (m/s), \(M\) - Mach number, \(V_s\) - The speed of sound (m/s), \(L\) - The lapse rate (0.0065 K/m), \(g\) - The gravitational acceleration (9.80665 m/s\(^2\)), \(R\) - The gas constant for air (287 m\(^2\)/s/K), \(\gamma\) - The ratio of specific heats (1.4).

The altitude of the aircraft can be calculated given the static pressure.

\[
h = \frac{T_0}{L} \left( 1 - \left( \frac{p}{p_0} \right) \left( \frac{R \cdot L}{g} \right) \right)
\]  

(2)

Figure 2 shows the results of an example calculation relating wing lift to flap deployment.

The aircraft model is based on the Airbus A320 passenger aircraft. Many of this aircraft’s physical characteristics are publicly available [4], some, such as the control surface areas and the distances between the centre of gravity and the centres of torques are impossible to obtain so were estimated.
Flight Control Subsystem

The flight control subsystem was encoded in VHDL and followed the specification of the Airbus A330/A340 System from the University of Twente [8]. The system incorporated servo and gearing mechanisms as described in Pallett [9]. It can operate in two modes. In the automatic mode, the flight control subsystem provides manoeuvre envelope protection which prevents manoeuvres that are beyond the capabilities of the aircraft.

The maximum roll angle is limited to prevent excessive side-slip. If the maximum roll angle is exceeded, the flight control subsystem attempts to determine whether the pilot is trying to undertake corrective action, by monitoring the side stick x-axis movement. If the flight control subsystem does not detect a corrective action, it steers the ailerons to provide an opposing rolling torque.

The maximum pitch angle and altitude are limited to prevent stalling. At high altitudes, wing stalling can occur due to low atmospheric density. If the maximum pitch angle or altitude is exceeded, the flight control subsystem attempts to determine whether the pilot is trying to undertake corrective action, by monitoring the side stick y-axis movement. If the flight control subsystem does not detect a corrective action, it steers the elevators to provide an opposing pitching torque. The maximum linear and rotational velocities are also limited to safeguard the structural integrity of the aircraft.

The flight control subsystem attempts to alleviate the effects of wind gusts and provide consistent handling by detecting rapid changes in linear velocities in the absence of pilot inputs. Wind gusts in the x-axis are handled by rolling the aircraft to generate an opposing corrective force. Wind gusts in the y-axis are handled by adjusting the thrust generated by the engines and in the z-axis by increasing the angle of attack to adjust the amount of lift generated.

The flight control subsystem provides passenger comfort by limiting the linear and rotational accelerations. If a limit is exceeded, the flight control subsystem attempts to determine whether the pilot is trying to undertake corrective action, by monitoring the pilot commands. If the flight control subsystem does not detect a corrective action, it steers the control surfaces to provide an opposing corrective force.

The manual mode is designed to be used in emergencies. The pilot has full control over the aircraft. The flight control subsystem does not process the pilot’s commands.

The graphical display module is a Java front-end application designed to enhance comprehension of simulation results. The module is made up of four classes:

- ParseData
- AudioThread
- FlightDeck

Figure 2  Variations in the wing lift coefficient with respect to the y-axis velocity and flaps deployment at an angle of attack of 0 rad

The flight control subsystem provides passenger comfort by limiting the linear and rotational accelerations. If a limit is exceeded, the flight control subsystem attempts to determine whether the pilot is trying to undertake corrective action, by monitoring the pilot commands. If the flight control subsystem does not detect a corrective action, it steers the control surfaces to provide an opposing corrective force.
Simulator

During compilation and test of the flight control subsystem output produced by the Synplicity VHDL compiler [6] verified that the flight control subsystem is synthesizable. The flight control subsystem was tested using the signals provided by the pilot and sensors subsystems. The two modes were tested in isolation. In the manual mode, the signal values received form the pilot subsystem and the signal values transmitted to the aircraft subsystem were compared to verify the conversion accuracy between the signals of types real and signed. The accuracy was found to be 0.1%, as expected from the scaling factor of 1000. In the automatic mode, the tests converged on the limiting values to determine the operation of the flight control subsystem. Tests showed that the flight control subsystem was activated at the limits and reacted correctly.

The graphical display module co-ordinate calculations were tested using MATLAB [7]. The MATLAB was used to plot the circular path followed by the horizon (xPosition, yPosition) for $-2\pi \leq \beta \leq 2\pi$. The offset was chosen to be 100 pixels. Correct operation was visually verified.

Conclusion

The flight control system for the Airbus A320 was successfully encoded in Very High Speed Integrated Circuit Hardware Description Language (VHDL). The compiler verified that the controller is synthesizable on a silicon chip. Testing of the controller with a graphical user interface confirms its functionality when used on a software model of an Airbus A320 derived from aerodynamic first principles. The flight control system is therefore a successful first step towards developing an onboard silicon controller after the following developments. The flight control subsystem could be improved by performing an in-depth scientific analysis of the aircraft subsystem.

![Figure 3 – Series digital controller](image)

Figure 3 shows a series digital controller model that could be used for the flight control subsystem. $D(z)$ represents the controller, $E(z)$ represents the discrete error sequence and $F(z)$ represents the $z$ transform of the controller output sequence.

The aircraft model would have to be analysed to find the time domain transfer functions $G(s)$ for each output $C(z)$ from the inputs $R(z)$. The pulse transfer functions $GH(z)$ could be derived from the $G(s)$ and $H(s)$. The controller $D(z)$ could then be derived using loop analysis.

References

[6]. Model Technology ModelSim 5.5c, ModelSim SE FLI Reference, Online Help; Model Technology Inc.
[7]. MATLAB http://www.mathworks.com [last accessed 18/12/2003].