

Project Initial Description

Voltes Fly – Group 3

Goals and Objectives

There are several goals that were set regarding the system. Overall all the components were to work together seamlessly without any noticeable error. Voltes fly provides the user with an entertaining experience, and for pilots; a form of training without having to fly an actual plane. Also, the simulator was to be good enough to compete with other low cost systems available online. Our system is broken up into 2 main modules as seen in **Figure 2.2.1**. For these modules, communication is carried out via serial. Additionally, there were specific goals related to each of these. The Interfacing System is composed of several components mostly designed by the group.



Figure 2.2.1 Flight Simulator

Interfacing System: The main objective of this system was to provide realism to the virtual flight experience through input and output devices. There are three components within this system which provide communication between the pilot and the simulator as seen in **Figure 2.2.2**; these devices are monitored by an electronic component called the SPU.

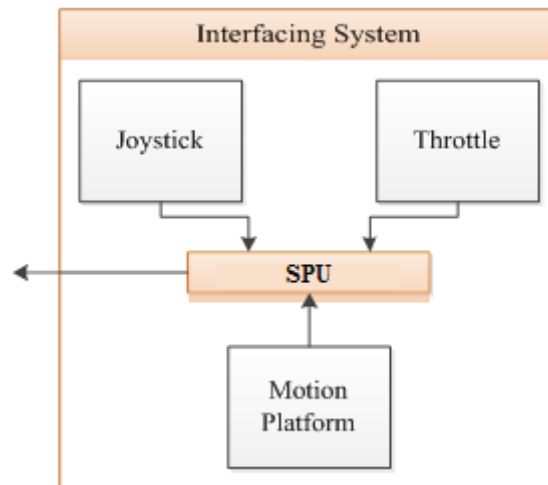


Figure 2.2.2 Interfacing Components

Any information coming or going to the flight computer is arranged and controlled by this unit. These devices are now explained and their goals and objectives are defined:

Joystick: The objective for this hardware peripheral was to be able to translate the user's hand movements into digital X and Y coordinate values that can be sent to the flight computer. The reaction time had to be low and the moving parts should have caused as less friction as possible so that the device could have a significant lifetime. The outside frame design consisted of an arcade style joystick. As far as the force feedback, the goal was to have several different levels of intensity available. There was flexibility regarding the type of feedback to be experienced: a static feedback which responds to the force applied by the pilot, or a dynamic one which deals with events within the game. In the second case the software and hardware interface must have provided enough software events to experience the effect to the maximum. In conclusion, the flight computer expected values from 0-100 percent for both the x and y axis of movement.

Throttle: It was only necessary to implement one throttle to support the kind of planes to be simulated. It was placed to the left of the joystick module. The user is able to set it at a value and the throttle stays in place until the next interaction. This lever component didn't allow for reversing the engines, just turning them on from zero to their max value. Any moving parts (gears etc.) experienced the least friction possible and had minimum lag when interfacing with the software. No buttons or flip switches were implemented; however, the possibility was there if time allowed. The value expected by the flight computer was an integer from 0-100.

SPU: The motion controller unit must have effectively outputted the right voltages to the servomotors according to the pitch and roll values which entered from the flight computer. The algorithms implemented had to be able to realistically translate these values and have as little lag and as little awkward movement as possible. Just like the other components, the parts used for this subsystem had to be cost efficient. It had to effectively pack and send both joystick and throttle information, simultaneously receiving the pitch and roll values from the flight computer.

The flight computer contained a crucial software component, which controlled several aspects of the flight simulator. It helped all the different systems work together. This system is the Instructor Operator Station and its goals are described below.

Instructor Operator Station: The goal of the Instructor Operator Station (IOS) was to allow an instructor to control simulation setting such as environment and starting airport runway. The IOS was the only user interface to the simulation software. The IOS also included modules responsible for sending and receiving packages of data from the Signal Processing Unit for the motion base, throttle, and joystick. Included with the IOS is a Flight Test page which allowed the viewing and setting of flight data parameters during an active simulation for the testing phase of the project.

Scope of Work

Flight Simulators have been around for some years already, but they usually present an expensive solution for those who often use them. Companies and enthusiastic fans are always trying to build their projects at a lower cost without compromising other features like safety,

reliability, accuracy, and scalability. Our project achieved all these functionalities at a lower cost, making possible the reproduction of this project by whoever is interested.

Objectives

The simulator built gave both pilots and non-pilots a satisfactory experience. Most importantly, the objective of this project was to impress the committee which witnessed a demonstration of Voltes Fly in December 2011; consequently allowing our group members to graduate.

Overall Specifications

The final product was designed to be portable, light weight and easy to use. The user had the ability to choose the aircraft of his preference from the available list on the X-Plane simulator. The front external windows were recreated by a single projector display. By having this configuration, the “pilot” achieved the desired degrees of sight. All the necessary changes and/or adjustments to the software simulator were made by the Instruction Operating System or IOS. In addition, the system was able to send readable information from the Joystick through USB to serial connector linking the flight computer to the SPU; this information contains the position of the joystick on the X and Y axes at any given time. The SPU was the main unit of our system, every piece of information and/or command went through it.

Moreover, the system was able to read and interpret the pitch and roll variables coming out of the software simulator. As before, this information is sent using serial communication. Once the SPU captured these variables, it sent the appropriate commands to each servo-motor, accomplishing the desired motion-platform position. In case of specific events like a plane crash, turbulence or landing, the system should have reflected those actions by recreating a rumbling on the joystick end in order to achieve a higher level of reality.

Joystick

The joystick was built from the scratch. It was made using two DC motors that create the force feedback effect. Two single-turn rotary potentiometers were used to track the movement of the X and Y axes. Also, several mechanical parts were used for the system: universal joint and a wooden shaft for the handle to allow for user input, a system that may center the handle automatically to its origin. Also, a decent range of motion was permitted so that a rotation of at least 45 degrees in each direction could be achieved.

Throttle

The mechanical system was attached to a rotary potentiometer whose position was altered by a single lever. The rotary potentiometer sent the analog signal to our SPU unit.

IOS

The Instructor Operator Station (IOS) provided communication between the other systems (SPU, X-Plane). It was broken into four modules: X-Plane Communication, SPU Communication, Rudder Pedal Communication, and IOS User Interface.

The X-Plane Communication module ran on a dedicated thread and controlled the bi-directional communication between X-Plane (contained the flight physicals model, and Image Generator (IG)). The bi-directional communication was implemented utilizing the User Datagram Protocol (UDP) via a local loop back (IOS and X-Plane ran on same machine). The IOS sent the

following data to X-Plane: roll, pitch, and weather settings. In return the IOS received those values from X-Plane.

SPU Communication module ran on a dedicated thread and controlled the bi-directional communication between the SPU and the IOS. The bi-directional communication was implemented using a subset of a serial protocol. Data was sent and received using data packets as the payload in a serial Data Packet. The IOS sent the following data to the SPU: roll and pitch. In return the IOS received the following from the SPU: joystick position and throttle position.

The IOS User Interface module handled the input from the instructor and controlled the communication between the X-Plane Communication and SPU Communication. When the IOS was launched, the User Interface (UI) appeared on a dedicated monitor and provided the instructor with the following options: start simulation, end simulation, set environmental parameters, and view flight parameters. It was the primary interface for communication between an instructor and the system.

Motion Base

The Motion Base was leased from a local company in Longwood, whose name is Servos and Simulation, Inc Voltes Fly incorporated this motion platform on the picture for many reasons. The first reason was, of course, cost. One of this motion platforms' price could easily be on the thousand dollars, so the group saved a lot of money by using it without cost. The second reason referred to the specifications. This unit met all the requirements the group had in mind in order to build the system effectively.