Flight Simulator Concept Using Java 3D and Java Speech

Ionel IORGA, Claudia STATE

Faculty of Mathematics and Computer Science,
Department of Computer Science,
University of Craiova, Romania
ioneliorga,cldstate@yahoo.com

Abstract. In this paper we propose an implementation of a flight simulator concept, for the purpose of testing various hypothesis regarding the movement of an aerodyne through the atmosphere, using a java 3D representation and Java Speech Api [2]. In particulary, we employed the Java 3D package [1], the speech recognizer Sphinx-4 [12] and the speech synthesizer FreeTTS [14].
We try to gain an insight into prevention of unwanted phenomena's like pilot induced oscillations (PIO) [13] in the context of commands given by voice. It is generally known that pilot induced oscillation is an inadvertent couple of the pilot and the aerodyne and even the newest airplanes, equipped with modern fly-by-wire or fly-by-light can have this problem due to their oversensitivity.
Keywords: flight simulator, virtual world, java 3d, java speech api, pilot induced oscillation
Math. Subject Classification 2000: 68T30; 68T35

1 Introduction

In the world of flight simulators there are softs like **flightgear** [3] or **microsoft flight simulator** [5] that are available to private use. If in the first stages of the development of such flight simulators there were owned and used mainly by governments organizations or industrial corporations (as the aircraft's simulated by them) now even relatively small private organizations can challenge difficult tasks like designing, testing, simulating and even building an aircraft, see for example, the groups which tried to claim the **Xprize** [4] - which was a serious task since involved, in the last flight stage, movement of the machine through extra-atmospheric space.
In literature are described implementations of various flight simulators, for example: F. Brabec in [6] construct a flight simulation using pure java, translating a C library into java in order to manipulate the 3D entities, J. Berndt shows in [7] the work at JSBSim open source software library modeling the flight dynamics of an aerospace vehicle, written in C++ using xml files for configuration, execution and control etc.

The psychological issues regarding the control by voice of the aircraft [8] have to be taken into account in order to diagnose correctly the possible appearance of PIO phenomena.

2 Pilot induced oscillations

Pilot induced oscillations occur when the pilot inadvertently commands the aircraft resulting antagonist movements. These oscillations tend to appear when sudden and disproportioned commands are executed in response to a "slight" disequilibrium of the flying machine. If we make an analogy with driving a car then this seems similar with a sudden and strong movement of the wheel into direction of the road when the car has partially, by mistake, left the road, resulting the loss of the control and possibly an accident. The proper action in this case is to reduce the speed and to gentle command the car back into the road. Also, in flight is recommended to command gentle the airplane with the excepted case of emergency situations (air combat etc.). So, for example, when an aircraft is piloted by voice - signifying the fact that it is added with some kind of automatic device which recognize and translate the commands given by voice into specific actions (loop, roll, yaw etc.) - then trajectory can be automatically regulated to match specific requirements in order to prevent instable patterns.

In order to know how to prevent the possibility of occurrence of PIO in flight we give the **PIO categories**:

- I. Essential linear pilot-vehicle system oscillations.
- II. Quasi-linear pilot-vehicle system oscillations with rate or position limiting.
- III. Essential non-linear pilot-vehicle system oscillations, such as multiple non-linearities, transitions in pilot behavior, etc.

It is easier to study category I PIO because they are essentially linear aircraft-pilot interaction and the dynamics of the augmented linear aircraft have been found to be the key factor for these problems.
Category I PIO prediction criteria are:

1. Neal-Smith (1971)
2. Smith-Geddes (Smith and Geddes, 1978)
3. Phase Rate Criterion (Gibson, 1982)
4. Gibson frequency domain Criterion (Gibson, 1982)
5. Gibson Time Domain Dropback (Gibson, 1982)
6. Bode Gain Template by Hess and Kalteis (Hess and Kalteis, 1991)
7. Bandwidth/phase delay (Hoh, et al, 1994)
8. Robust Stability Analysis (Anderson and Page, 1994)
9. Updated Dropback criterion by Mitchell and Hoh (Mitchell, et al., 1994)

10. Power Spectral Density Analysis of the Pilot Structural Model (Hess, 1997)

All these prediction criteria address stability aspects of closed-loop aircraft-pilot systems. The Neal-Smith, Smith-Geddes, Robust Stability Analysis criteria define a pilot model and use it for the analysis of the closed-loop system meanwhile other criteria use, instead, the open-loop aircraft with implicit model of the pilot [15], p. 78.

3 Employing the java language

Java 3D is a scene-graph based 3D application programming interface for java platform and it runs on top of either OpenGL or Direct3D.

A scene-graph represents in a logic manner the spatial representation of a graphic scene.

In the Figure 1 is plotted a scene graph for a plane.

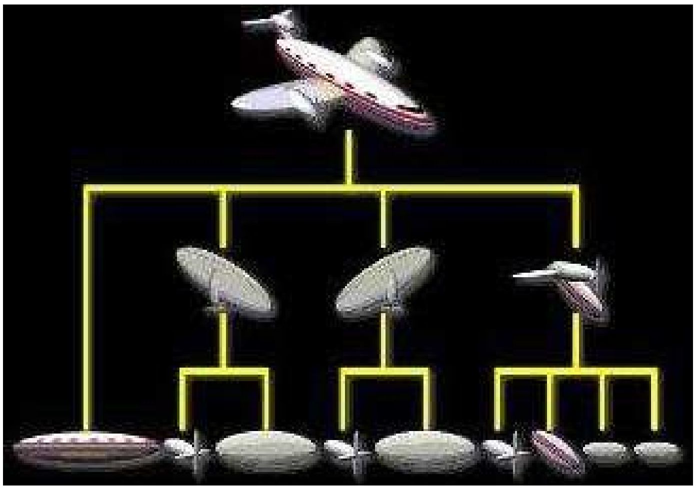

Fig. 1. The scene graph for a plane

Scene graphs are built from components including:

- Shapes (geometry and appearance)
- Groups and transforms
- Lights

- Fog and backgrounds
- Sounds and sound environments (reverb)
- Behaviors
- View platforms (viewpoints)

The above components are rendered using separate, independent and asynchronous threads.

In order to simulate motion rotations, translations, scaling and shear effects we used Klawonn[11], p. 130.

With Java Speech API we incorporate speech technology into our user interface and, using FreeTTS[14] and Sphinx [12] we benefit of open-source characteristics in order to provide a versatile-cross-platform application knowing the fact that speech technology is a very useful domain in the with applications in assistance of people with special needs, air traffic controllers or the pilots on modern airplanes.

The Apache Ant [10] has been used in order to build and integrate our application.

In Figure 2 we can see a screen shot of a flight simulated in the application:

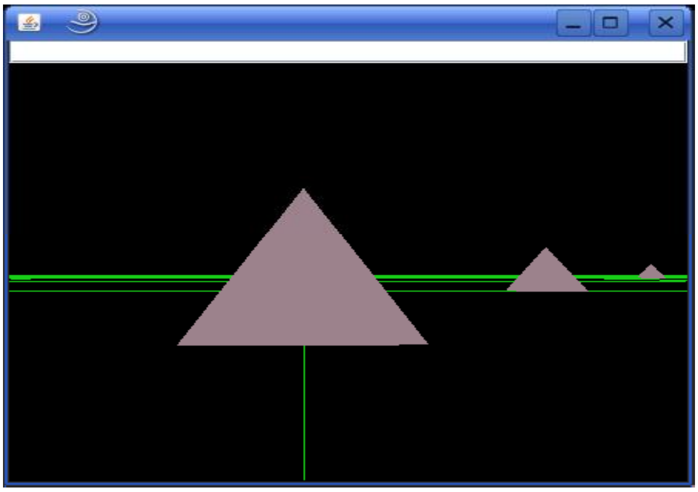

Fig. 2. 3D Flight screen shot

4 Experimental results

We considered the motion of the rigid aeroplane (i.e. **gross motion** [9], p. 2) since the use of *fine motion* (like **liquid sloshing** in the tanks of the aeroplane) determine a great amount of computability and was not relevant for our immediate goal: to provide a model checking of the added, voice driven, software system which improves the command of the flying machine simulated. However the method proposed has proved her effectiveness using the experimental procedure.

5 Conclusions and future work

The sessions of application, through the bounds of the variables imposed to the dynamic system, had confirmed the theoretical considerations regarding the stability pattern of the simulated flights.

As a future work, a more complicated dynamic model should be take into account in order to use non-linear PIO prediction criteria.

References

1. ***, http://java.sun.com/javase/technologies/desktop/java3d/index.jsp
2. ***, http://java.sun.com/products/java-media/speech/
3. ***, http://www.flightgear.org
4. ***, http://www.xprize.org/space/x-prize-cup
5. ***, http://www.microsoft.com/games/flightsimulatorx/
6. ***, http://www.cs.umd.edu/~brabec/flight/index.html
7. **J. S. Berndt**, JSBSim: An Open Source Flight Dynamics Model in C++, AIAA Modeling and Simulation Technologies Conference and Exhibit, August 2004, Rhode Island
8. ***, http://en.wikipedia.org/wiki/Direct_Voice_Input
9. **B. Etkin, L. D. Reid**, Dynamics of Flight Stability and Control, Third Edition, 1996, John Wiley and Sons,Inc.
10. ***, http://ant.apache.org/
11. **F. Klawonn**, Introduction to Computer Graphics Using Java 2D and 3D, Springer-Verlag, 2008
12. ***, http://cmusphinx.sourceforge.net/sphinx4/
13. ***, http://en.wikipedia.org/wiki/Pilot_induced_oscillation
14. ***, http://freetts.sourceforge.net/docs/index.php
15. ***, ftp://ftp.rta.nato.int/PubFullText/RTO/TR/RTO-TR-029/