

institut für elektronische musik und akustik



## **IEM Report 37/06**

### **Forschungsplattform Flugsimulation - Sound Modul**

#### **Verfasser:**

Alois Sontacchi

#### **Kooperationspartner:**

Techn. Universität Graz, APUS, Axis Flight Training Systems

Oktober 2006

## **Zusammenfassung**

Im Rahmen einer universitätenübergreifenden wird ein Flugsimulator entwickelt, der in Forschung und Lehre Verwendung finden soll. Von der mechanischen Modellierung der Flugdynamik über Untersuchungen von Piloten unter Stresseinfluss bis hin zur dreidimensionalen Sound-Simulation von Fluggeräuschen reicht die breite Palette an Themen, zu denen am Flugsimulator geforscht wird. Disziplinenübergreifende Thematiken brauchen eine disziplinenübergreifende Form der Zusammenarbeit: Die beteiligten Wissenschaftler schließen sich daher zur interuniversitären „Forschungsplattform Flugsimulation“ zusammen und wollen Forschung und Lehre in diesem Bereich gemeinsam vorantreiben. Zu den Projektpartner an dieser Forschungsplattform zählen alle drei Grazer Universitäten. Das Institut für Elektronische Musik und Akustik (KUG) ist zuständig für die Umsetzung der 3D Soundsimulation und für das Sounddesign.

## **Abstract**

In the context of interdisciplinary Project including TU-Graz, KFU, and KUG we developed a flightsimulator, which will be used in research and teachings. The broad range of possible applications covers the modeling of flight dynamics, stress resistance of pilotes, and reproduction of three dimensional sound environements. Therefore an extensive network is requested to cover all the different aspect of this interdisciplinary project. The institut of Electronic Music and Acoustics is concerned with the realisation of 3D Sound Simulation and Sound Design.

# Inhaltsverzeichnis

Zahlen & Fakten im Überblick

IEM Zwischenreport

IEM Abschlussbericht

Presseunterlagen

Publikationen bei der DAFX'05 in Madrid

- A New Functional Framework For A Sound System For Realtime Flight Simulation
- Sound-System Design for a Realistic Flight Simulator

Flugsimulator – Dokumentation der Stand-Alone-Version des Simulators

Flugsimulator – README-Datei zur Verzeichnisstruktur und zur Verwendung des Simulators

# **1. Zahlen & Fakten im Überblick**

## **1.1 Zeitpunkt der Gründung:**

Juli 2004

## **1.2 Laufzeit:**

IEM 15 Monate (Gesamtprojekt 2 Jahre)

## **1.3 Organisationsform:**

interuniversitäre und industrielle Forschungsk Kooperation

## **1.4 Personen und Finanzen:**

IEM 3 Mitarbeiter mit einem Projektbudget von 57.000 EUR

(Anzahl der MitarbeiterInnen u. genehmigtes Gesamtvolumen unbekannt,  
Fördersumme durch Zukunftsfonds Land Steiermark: 400.000 EUR).

## **1.5 Inhaltliche Schwerpunkte:**

Entwicklung eines Flugsimulators höchster Realitätsstufe

## **1.6 Ausrichtung:**

angewandte Forschung

## **1.7 Gesamtprojektpartner:**

TU-Graz, KFU-Graz, KUG, AXIS, Apus, Brightline, Rolls Royce.

Institut für Maschinenbau- und Betriebsinformatik (TU-Graz) mit den Arbeitsschwerpunkten Software, System Identification, Sounddesign,

Institut für Mechanik (TU-Graz) verantwortlich für die Bereiche Flugdynamik, Fahrwerksimulation, Ruderdrucksimulation,

Institut für Psychologie (KFU) Untersuchungen der Reaktionsvermögen unter Stressbedingungen beim Fliegen.

Institut für Elektronische Musik und Akustik (KUG) zuständig für die Umsetzung der 3D Soundsimulation und Sounddesign

Darüber hinaus erfolgt die Zusammenarbeit mit den Grazer Firmen Axis Flight Training Systems, Apus und Brightline sowie mit dem weltweit bekannten Triebwerkshersteller Rolls Royce. Die Gesamtleitung obliegt dem Institut für Maschinenbau- und Betriebsinformatik (TU-Graz) unter Univ. Prof. Dr. Siegfried Vössner.

Das Projekt wird freundlicherweise unterstützt von den Austrian Airlines, AMD, nVidia, pny und NEC.

## 2. Aufgaben des IEM i. R. der Forschungsplattform im Detail

### 2.1 Aufgaben

Dem IEM obliegt dabei die Aufgabe der Umsetzung eines echtzeitfähigen steuerbaren Soundmoduls. Dieses Soundmodul ist zusammen mit dem Audiomodul, welches die Aufgabe der Interkommunikation und ggf. der Generierung von Eigengeräusche der Bordenlemente hat, für eine möglichst naturgetreue Nachbildung aller Schallereignisse im Cockpit von Flugzeugen im oben genannten Flugsimulator zuständig. Basierend auf einem funktional motiviertem Ansatz wird hierzu ein Schallfeld resynthetisiert, welches dem realen Schallfeld möglichst nahe kommen soll (s.o.). Mit diesem System ist es prinzipiell möglich, die Schallereignisse jedes beliebigen Flugzeugmodells auf Simulatoren abzubilden. Weil sich die verwendete Technik jedoch auf Aufnahmen eines ganz bestimmten Flugzeuges stützen muss, wird das fertige Produkt lediglich auf Maschinen eines festgelegten Typs, nämlich auf Maschinen des Typs Fokker 100, eingehen können. Das Gesamtsystem wird aus einem Cockpit mit naturgetreuen identischen Bedienelementen (inkl. force feedback), mit einem hochqualitativen realen Vision und Audio System, sowie einer Motion Base ausgestattet sein.

### 2.2 Quellenseparation

Das Soundmodul hat die Aufgabe sämtliche Geräusche die in einem Cockpit stattfinden getrennt voneinander zu verändern und einzeln zuspiesen zu können. Das Problem, dass sich jedoch dabei in der Praxis ergibt ist folgendes: Bei der Aufnahme liegen sämtliche Quellen oft gleichzeitig überlagert vor. Das wissenschaftliche Stichwort dazu heißt Quellenseparation. Als bildhaftes Beispiel sei folgende Aufgabe zu nennen: *„Stellen Sie sich eine fertige Eierspeise vor. Ziel ist es nun Eigelb und Eiklar voneinander zu trennen“*.

Bei der Schallfeldsituation im Cockpit ist es ähnlich - jedoch haben wir hier nicht nur 2 sondern mehrere/viele unterschiedliche Geräusche und Geräuschquellen. Um die wichtigsten globalen Komponenten zu nennen:

- die Flugzeugform,
- das Triebwerk,
- das Fahrwerk und
- die Umgebung - Environment (Wetter & Terrain)

Diese Punkte lassen sich wieder in Unterquellen für die jeweilige Flugsituation zerlegen. Einige Geräusche, die diesen Quelle zugeordnet werden können, sind vom Flugzustand unabhängig. Ein Beispiel: *das mechanische Verriegelungsgeräusch des Fahrwerks. Dieses klingt in 500 Fuß gleich wie in 20000 Fuß.*

Das Strömungsgeräusch ist deutlich vom Flugzustand abhängig. Denken Sie an Fluggeschwindigkeit, Flughöhe, Luftanströmrichtung ... um nur einige Faktoren – *Flugparameter* - zu nennen.

Zur Lösung dieser Aufgabe werden zum einen spezielle Aufnahmeverfahren – Stichwort Binauralmesstechnik aber auch die Arraytechnologie (Beamforming) verwendet. Zum anderen werden neben der Verfügbarkeit von Flugdaten die Einzelkomponenten durch fortgeschrittene Signalverarbeitung (advanced signal processing) isoliert.

### **2.3 Spatialisierung (Raumklang)**

Das wesentliche Novum für Flugsimulatoren stellt die Spatialisierung – d.h. die räumliche Verteilung der Geräusche und Klänge dar. So können nicht nur die realitätsnahen Klänge aus einer bestimmten Richtung abgebildet werden, sondern es wird jedem Klang auch seine reale räumliche Dimension zugeordnet. Dies erzeugt somit eine weitere Steigerung des Realitätsgrades. Dadurch befindet man sich in einem Klangraum – d.h. tatsächlich vom Schall wie im echten realen Raum umgeben.

Da im realen Cockpit nicht beliebig viele freie d.h. unbenötigte Plätze vorhanden sind, ergibt sich dadurch auch ein zusätzliches Optimierungsproblem.

### **2.4 Ergebnisse**

Die Anforderungen im Rahmen des Level-D Standards sehen für den Schalldruckpegel, bei beliebigen definierbaren Flugzuständen, eine Wiedergabe innerhalb eines definierten Toleranzbandes über der Frequenz vor. Diese Anforderungen werden von uns nicht nur erfüllt, sondern um einen Faktor 3 unterschritten. Darüber hinaus werden sogar die räumlichen Aspekte der Geräuschquellen berücksichtigt, die im Standard noch gar nicht festgelegt wurden.

Zusammengefasst heißt dass: es handelt sich bei unserem Sound Modul nicht um eine Kopie oder Next-Step Entwicklung sondern hier handelt es sich um eine neue Generation.

## **2.5 Repräsentations- und wissenschaftlicher Output**

Zwei Publikationen bei einer internationalen Konferenz (DAFX'06) in Madrid.

Vössner S., Braunstingl R., Ploner-Bernard H., Sontacchi A., "[A New Functional Framework for a Sound System for Realtime Flight Simulation](#)", DAFx Proceedings, 8th Int. Conference on Digital Audio Effects (DAFx'05), Madrid, Spain, September 20-22, 2005.

Ploner-Bernard H., Sontacchi A., Lichtenegger G., Vössner S., "[Sound-System Design for a Professional Full-Flight Simulator](#)", DAFx Proceedings, 8th Int. Conference on Digital Audio Effects (DAFx'05), Madrid, Spain, September 20-22, 2005.

Öffentliche Präsentation mit Pressekonferenz am 15.09.2005.

TV Sendebeitrag, Erstaussendung: Modern Times am 07.10.2005 um 22.35 in ORF2.

Weitere Informationen und Beschreibungen zum Projekt im Internet:

<http://mbi.tugraz.at/forschungsplattform/index.html>

[http://www.iem.at/projekte/dsp/flugsimulator/project\\_view](http://www.iem.at/projekte/dsp/flugsimulator/project_view)

<http://moderntimes.orf.at/>



## Soundmodul Flugsimulator

Project Manager: Alois Sontacchi (sontacchi@iem.at)

Contributors : Helmuth Ploner-Bernard, Robert Höldrich

**Das IEM hebt ab ! oder: Entwicklung und Implementierung des Soundmoduls für einen Flugsimulator (Level D).**

### 1. Einleitung

Das IEM entwickelt gemeinsam mit der TU-Graz (Inst. f. Allg. Mechanik, Inst. f. Strömungslehre, Inst. f. Maschinenbau Betriebsinformatik, Thermodynamische Turbomaschinen, Inst. f. Informationssysteme und Computer Medien) und Grazer Firmen, unter Fördergeldern des Landes Steiermark, einen „Level D“ Flugsimulator (höchste Ausbau und Entwicklungsstufe eines Flugsimulators – d.h. maximal realitätsnahe).



**Abbildung 1:** Das IEM hebt ab! – Die „Flight Simulation Partner“

Das Gesamtsystem wird aus einem Cockpit mit naturgetreuen identischen Bedienelementen (inkl. force feedback), mit einem hochqualitativen realen Vision und Audio System, sowie einer Motion Base ausgestattet sein.



**Abbildung 2:** Flugsimulator (Level D) – und seine Bestandteile: Cockpit, Audio System, Vision System, Motion Base



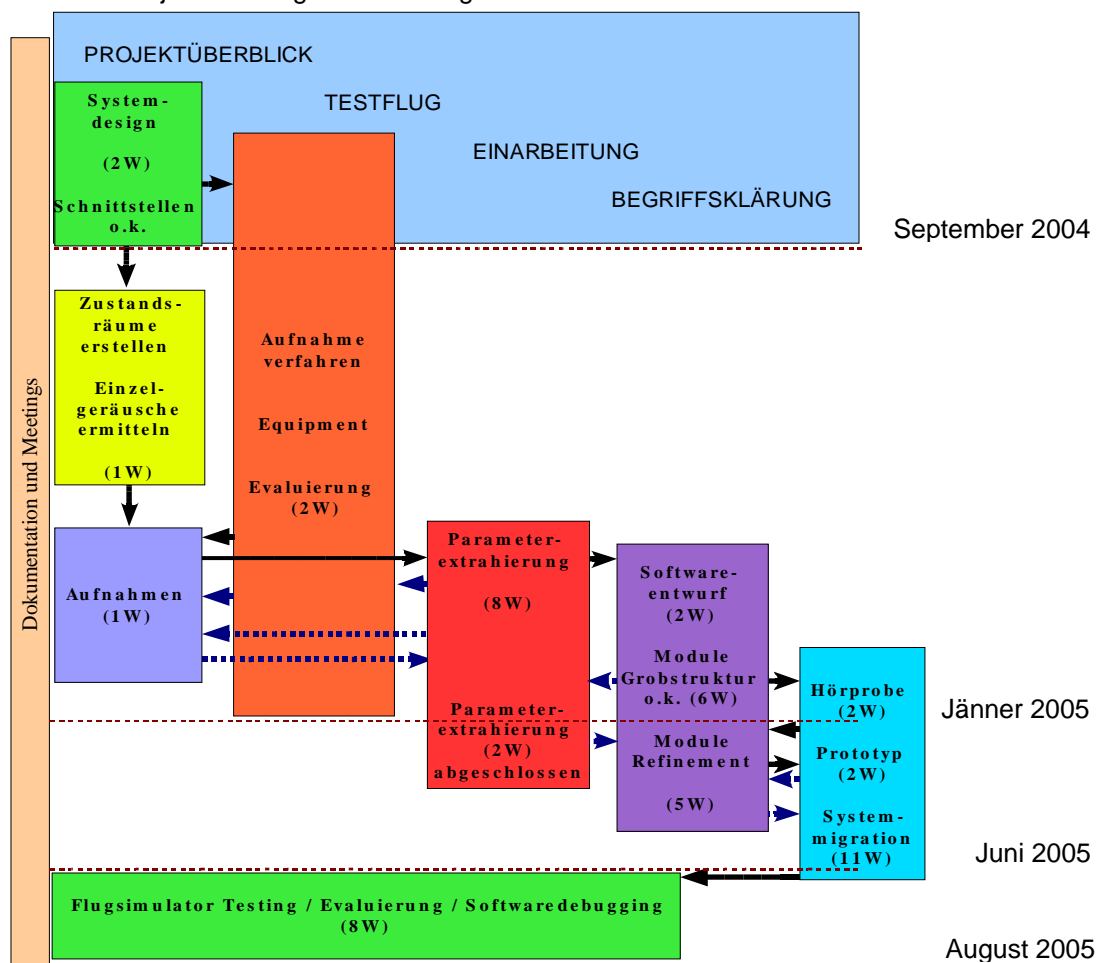
## 2. Aufgabenstellungen für das Soundmodul

Das *Sound-Modul* ist zusammen mit dem Audio-Modul, welches die Aufgabe der Interkommunikation und Generierung der Eigengeräusche der Bordenlemente – falls nicht vorhanden - hat, für eine möglichst *naturgetreue Nachbildung* aller Schallereignisse im Cockpit von Flugzeugen in einem Flugsimulator zuständig. Basierend auf einem funktional motiviertem Ansatz wird hierzu ein *Schallfeld resynthetisiert*, welches dem realen Schallfeld möglichst nahe kommen soll. Mit diesem System ist es prinzipiell möglich, die Schallereignisse jedes beliebigen Flugzeugmodells auf Simulatoren abzubilden. Weil sich die verwendete Technik jedoch auf Aufnahmen eines ganz bestimmten Flugzeuges stützen muss, wird das fertige Produkt lediglich auf Maschinen eines festgelegten Typs eingehen können.

### Auflistung der notwendigen Arbeitsschritte:

- Voraufnahmen und Begriffs Screening
- Strategien für die Aufnahmen im Flugzeugcockpit
- Analyse der aufgenommenen Schalle.
- Synthesestrategien
- Spatialisierung
- Implementierung
- Dokumentation

Daraus resultiert das Projektflussdiagramm wie folgt:



In den folgenden Abschnitten werden die einzelnen wesentlichen Punkte jedoch nur oberflächlich und zum Teil noch unvollständig besprochen.

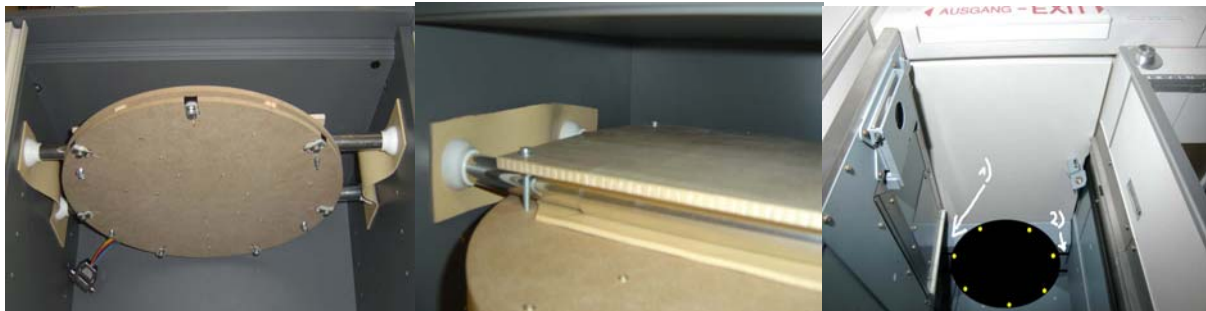
### 3 Das Aufnahme Equipment

Das Schallfeld im Cockpit wird durch mehrere Geräuschquellen hervorgerufen, die sich zeitlich, spektral sowie auch örtlich (z. T. zeitvariant) überlagern. Die Schallausbreitung erfolgt über Körper- als auch Luftschall (structure-borne & airborne sound). Unter diesen Gesichtspunkten resultieren unterschiedliche Auswahlkriterien für das Aufnahme Equipment bzw. Setup, wobei Vor- und Nachteile entsprechend abzuwiegen sind. Dazu zählen Kalibrierbarkeit, Qualität bzgl. des verfügbaren Übertragungsbereiches, SNR, Grenzschalldruck, Speicheraufwand, Hardwareaufwand, etc..

Folgende Lösungsansätze werden verfolgt:

- Einzel(mess)mikrofon Aufnahmetechnik
- Binaurale Aufnahmetechnik
- Arraytechnologie

Ein wesentlicher Punkt beim Aufnahme Equipment ist auch dessen Halterung bzw. Anbringung im Cockpit. Während die Anbringung und Befestigung eines Einzelmikrofons noch eine relativ einfache lösbare Aufgabe darstellt, geben sich bei den anderen Verfahren, vor allem bei der Verwendung von Aufnahmen bei Linienflügen größere Hürden. Als eine Innovative Entwicklung kann das Halterungssystem für ein Mikrofonarray, welches zusammen mit dem MBI entwickelt wurde genannt werden.



**Abbildung 3:** Speziell entwickeltes Mikrofonarray (IEM) und Halterung (IEM&MBI)  
für den Einsatz bei Linienflügen im Flugzeugcockpit.

### 4 Analyse

Der Hauptaufgabe der Analyse ist die Trennung von stochastischen (geräuschartigen) und deterministischen (harmonischen) Komponenten und deren Zuordnung bzw. Einordnung in den Flugzustandsraum und deren Zuordnung von bestimmaren oder unbestimmaren Schallereignisorten. Der Flugzustandsraum wird durch relevante Flugdaten wie: Drehzahlen der Triebwerke, Flughöhe, Fluggeschwindigkeit, Orientierung, Stellung der Flaps usw. beschrieben.

Dabei werden folgende Aspekte zusätzlich unterschieden:

- Zeitliche Abläufe
- Spektrale Unterscheidung
- Räumliche Separation / Zuordnung

## 5 Synthesestrategien

Grundsätzlich kann die Synthese nicht unabhängig von der Analyseseite betrachtet werden. Jedoch ergibt sich hier im vorliegenden Projekt die zusätzliche Erschwernis, dass im realen Cockpit die Aufnahmen nicht für alle möglichen Zustände im Flugzustandsraummodell – wie diese bei einem Flugsimulator erwartet werden - erhoben werden können (bedingt durch extremen zeitlichen Aufwand und auch durch Sicherheitsgründe). Diese Zustände müssen im Sinne der Level D Vorgaben weitest gehend aus den vorhandenen Daten extrapoliert werden oder aber auch sinngemäß generisch synthetisiert werden.

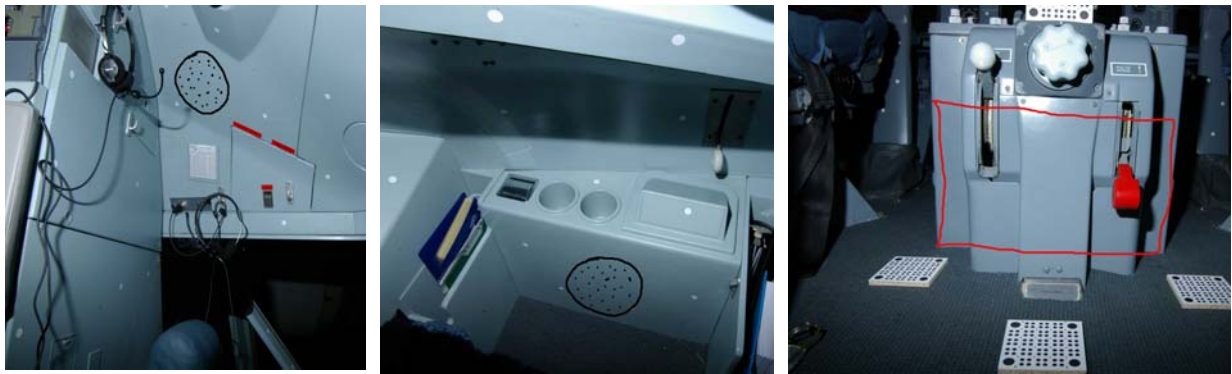
Basierend auf den Aufnahmedaten und den extrahierten Features können z. T. quasi-stationäre Schalle z.B. durch einfache Quelle-Filter Modelle nachgebildet werden. Dabei ändert sich zeitlich nur das Filter jedoch das Quellen- oder Generatorsignal bleibt unangetastet.

Außerdem müssen Zustandsraum (relativ) unabhängige Ereignisse, die einem speziellen zeitlichen Verlauf unterworfen sind wiederum anders behandelt werden (z.B. Öffnen der Fahrwerksklappen und Ausfahren des Fahrwerks durch variable geloopte Samples)

## 6 Spatialisierung

Das Schallfeld in der Kabine setzt sich aus richtungsunabhängigen (Druckkabine) und richtungsabhängigen Komponenten (lokalisierbare Quellenrichtung) zusammen.

Für die Wiedergabe im Flugsimulator sind mindestens 8 Lautsprecherpositionen (unabhängige Wiedergabekanäle) vorgesehen. Diese werden an möglichst unauffälligen Positionen maximal effizient angebracht.



**Abbildung 4:** Einzelne voraussichtliche Lautsprecherpositionen.

Da die Kabine beim Flugsimulator nur bedingt identisch nachgebaut wird (z.B. Instruktorplatz hinter dem Cockpit), kann bei der akustischen Nachbildung nicht von einer gleichen Wiedergabebedingung wie im realen Cockpit ausgegangen werden. Daher werden unterschiedliche Wiedergabestrategien zum Einsatz kommen. So werden z.B. breitbandige Geräusche die keine deterministische Abbildungsrichtung besitzen durch eine Kombination von vollständig korreliert (Druckkammer – für tiefe Frequenzen) und weitestgehend dekorreliert (Einhüllendes Ambiente – für mittlere bis hohe Frequenzen) extrahierten Geräuschkomponenten zusammengesetzt.

## **7 Implementierungsvorgaben**

Die Implementierung erfolgt unter PD (pure data) auf einem handelsüblichen PC unter Linux mit einer RME Soundkarte.

Das Sound-Modul muss auf ein angefordertes Geräusch (bedingt durch andere Module/Bedienelemente) vor Ablauf einer gewissen Zeit mit der Ausgabe eines entsprechenden Geräusches reagieren können. Die Anforderungen bzgl. minimaler Systemlatenz sind durch die Synchronisierung mit der Motion Base vorgegeben und beträgt somit 31ms.

Die minimale Systemlatenz setzt sich aus folgenden Größen zusammen:

- Garantierte obere Schranke des UDP für Send-Receive: 5ms
- Verzögerung des Videosystems: 10ms
- Verzögerung der Motion Base zu Videosystem: 16ms

## **8 Dokumentation**

Die Dokumentation erfolgt in Teildokumenten parallel zu den einzelnen Arbeitsschritten, wobei diese neben den üblichen schriftlichen Aufzeichnungen, Vereinbarungen, Konzepte und Lösungsstrategien auch aus originalen und strukturierten Schallaufzeichnungen, Softwareskripts, Fotos, Filme und dergleichen bestehen. Diese Unterlagen dienen als Vorlage für Gesamtdokumentation.



## Soundmodul Flugsimulator

Project Manager: Alois Sontacchi (sontacchi@iem.at)

Contributors : Helmuth Ploner-Bernard, Robert Höldrich

### Abschlussbericht der Entwicklung und Implementierung des Soundmoduls für einen Flugsimulator höchster Realitätsstufe (Level D).

#### 1. Einleitung

Im Rahmen einer universitätenübergreifenden Plattform wird ein Flugsimulator entwickelt, der in Forschung und Lehre Verwendung finden soll.

Von der mechanischen Modellierung der Flugdynamik über Untersuchungen von Piloten unter Stresseinfluss bis hin zur dreidimensionalen Sound-Simulation von Fluggeräuschen reicht die breite Palette an Themen, zu denen am Flugsimulator geforscht wird. Disziplinenübergreifende Thematiken brauchen eine disziplinenübergreifende Form der Zusammenarbeit: Die beteiligten Wissenschaftler schließen sich daher zur interuniversitären „Forschungsplattform Flugsimulation“ zusammen und wollen Forschung und Lehre in diesem Bereich gemeinsam vorantreiben.



**Abbildung 1:** Das IEM hebt ab! – Die „Flight Simulation Partner“

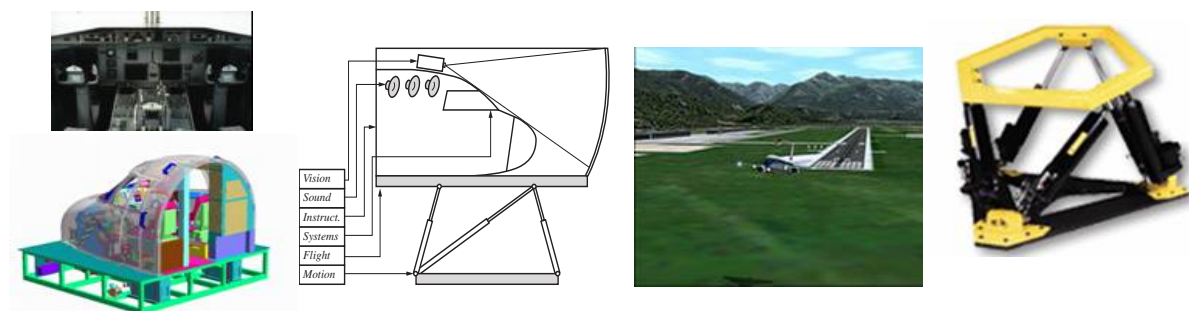
Zu den Projektpartnern an dieser Forschungsplattform (vgl. Abb.1) zählen alle drei Grazer Universitäten – die Technische Universität Graz (TU-Graz), die Karl-Franzens-Universität Graz (KFU) und die Kunstuniversität Graz (KUG). Diese sind vertreten durch folgende Institute:

- Institut für Maschinenbau- und Betriebsinformatik (TU-Graz) mit den Arbeitsschwerpunkten Software, System Identification.
- Institut für Mechanik (TU-Graz) verantwortlich für die Bereiche Flugdynamik, Fahrwerksimulation, Ruderdrucksimulation.
- Institut für Psychologie (KFU) Untersuchungen der Reaktionsvermögen unter Stressbedingungen beim Fliegen.
- Institut für Elektronische Musik und Akustik (KUG) zuständig für die Umsetzung der 3D Soundsimulation und Sounddesign.

Darüber hinaus erfolgt die Zusammenarbeit mit den Grazer Firmen Axis Flight Training Systems, Apus und Brightline sowie mit dem weltweitbekannten Triebwerkshersteller Rolls Royce. Das Projekt wird freundlicherweise unterstützt von den Austrian Airlines, AMD, nVidia, pny und NEC. Das Land Steiermark fördert im Rahmen des Zukunftsfond Steiermark dieses Projekt mit 400.000 EUR.

Das zentrale Ziel der Entwicklungen, i.R. der Forschungsplattform, ist einen Flugsimulator der „Level D“ Klasse nach JAA (European Aviation Authorities) d.h. höchste Ausbau und Entwicklungsstufe eines Flugsimulators, die den Anspruch auf „maximal realitätsnahe“ legt. Das Gesamtsystem wird aus einem Cockpit mit naturgetreuen identischen Bedienelementen (inkl. force feedback), mit einem hochqualitativen realen Vision und Audio System, sowie einer Motion Base ausgestattet sein.

Dem IEM obliegt dabei die Aufgabe der Umsetzung eines echtzeitfähigen steuerbaren Soundmoduls. Dieses Soundmodul ist zusammen mit dem Audiomodul, welches die Aufgabe der Interkommunikation und ggf. der Generierung von Eigengeräusche der Borelemente hat, für eine möglichst naturgetreue Nachbildung aller Schallereignisse im Cockpit von Flugzeugen im oben genannten Flugsimulator zuständig. Basierend auf einem funktional motiviertem Ansatz wird hierzu ein Schallfeld resynthetisiert, welches dem realen Schallfeld möglichst nahe kommen soll (s.o.). Mit diesem System ist es prinzipiell möglich, die Schallereignisse jedes beliebigen Flugzeugmodells auf Simulatoren abzubilden. Weil sich die verwendete Technik jedoch auf Aufnahmen eines ganz bestimmten Flugzeuges stützen muss, wird das fertige Produkt lediglich auf Maschinen eines festgelegten Typs, nämlich auf Maschinen des Typs Fokker 100, eingehen können. Das Gesamtsystem wird aus einem Cockpit mit naturgetreuen identischen Bedienelementen (inkl. force feedback), mit einem hochqualitativen realen Vision und Audio System, sowie einer Motion Base ausgestattet sein.



**Abbildung 2:** Flugsimulator (Level D) – und seine Bestandteile:  
Cockpit, Audio System, Vision System, Motion Base

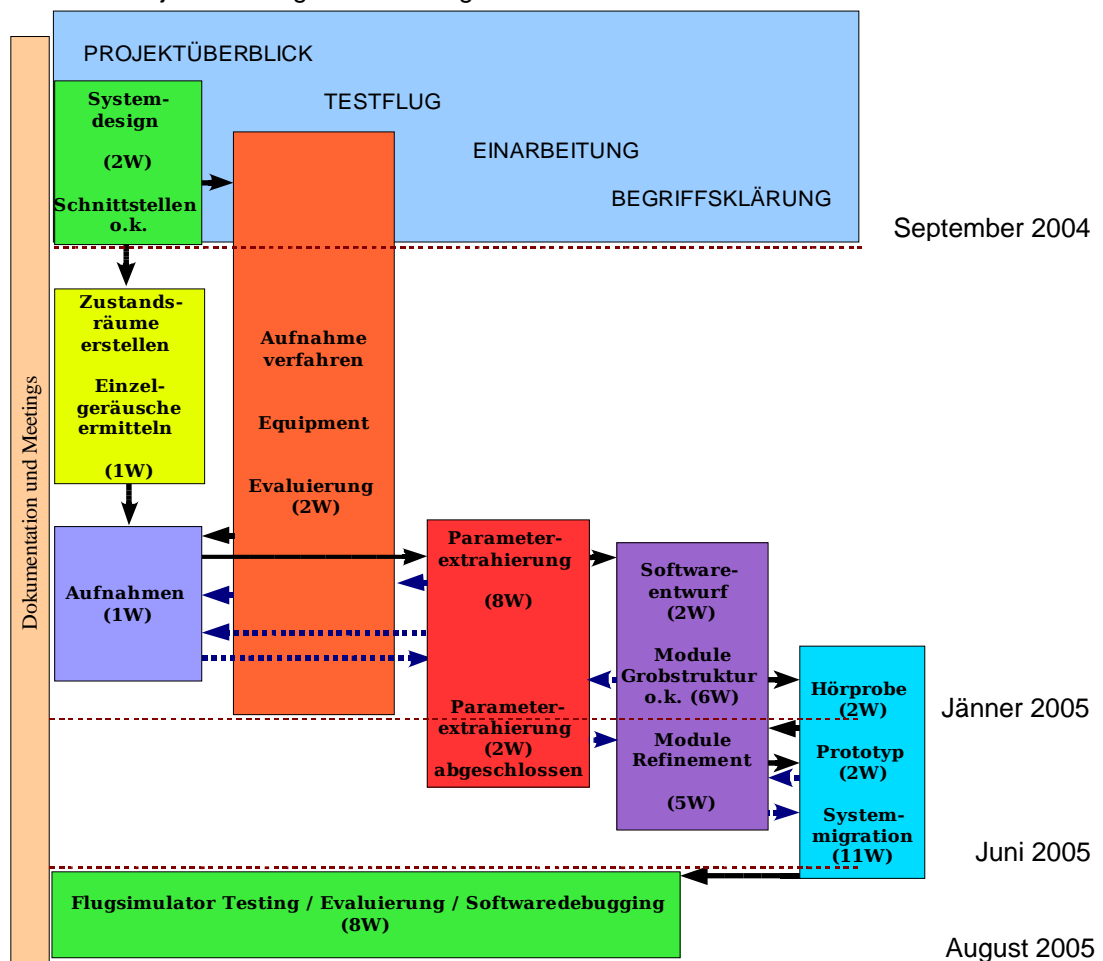
## 2. Aufgabenstellungen für das Soundmodul

Das *Sound-Modul* ist zusammen mit dem Audio-Modul, welches die Aufgabe der Interkommunikation und Generierung der Eigengeräusche der Borelemente – falls nicht vorhanden - hat, für eine möglichst *naturgetreue Nachbildung* aller Schallereignisse im Cockpit von Flugzeugen in einem Flugsimulator zuständig. Basierend auf einem funktional motivierten Ansatz wird hierzu ein *Schallfeld resynthetisiert*, welches dem realen Schallfeld möglichst nahe kommen soll. Mit diesem System ist es prinzipiell möglich, die Schallereignisse jedes beliebigen Flugzeugmodells auf Simulatoren abzubilden. Weil sich die verwendete Technik jedoch auf Aufnahmen eines ganz bestimmten Flugzeuges stützen muss, wird das fertige Produkt lediglich auf Maschinen eines festgelegten Typs, nämlich auf Maschinen des Typs *Fokker 100*, eingehen können.

### Auflistung der notwendigen Arbeitsschritte:

- Voraufnahmen und Begriffs Screening
- Strategien für die Aufnahmen im Flugzeugcockpit
- Analyse der aufgenommenen Schalle.
- Synthesestrategien
- Spatialisierung
- Implementierung
- Dokumentation

Daraus resultiert das Projektflussdiagramm wie folgt:





In den folgenden Abschnitten werden die wesentlichen Punkte im Überblick behandelt und umrissen.

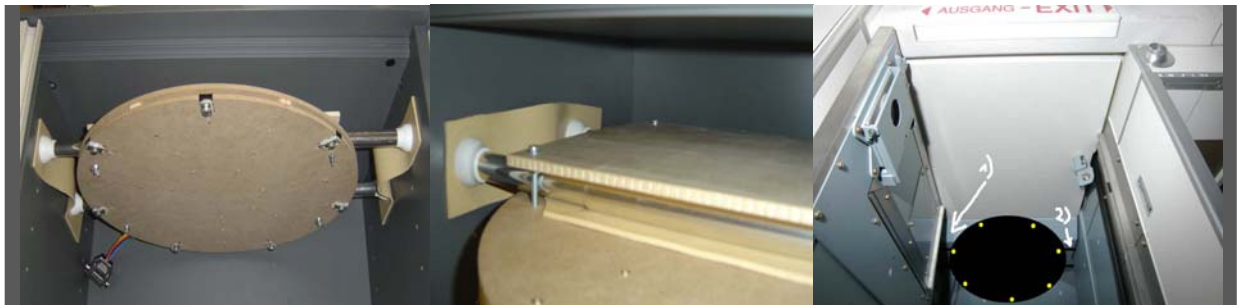
### 3 Das Aufnahme Equipment

Das Schallfeld im Cockpit wird durch mehrere Geräuschquellen hervorgerufen, die sich zeitlich, spektral sowie auch örtlich (z. T. zeitvariant) überlagern. Die Schallausbreitung erfolgt über Körper- als auch Luftschall (structure-borne & airborne sound). Unter diesen Gesichtspunkten resultieren unterschiedliche Auswahlkriterien für das Aufnahme Equipment bzw. Setup, wobei Vor- und Nachteile entsprechend abzuwiegen sind. Dazu zählen Kalibrierbarkeit, Qualität bzgl. des verfügbaren Übertragungsbereiches, SNR, Grenzschalldruck, Speicheraufwand, Hardwareaufwand, etc.

Folgende Lösungsansätze werden verfolgt:

- Einzel(mess)mikrofon Aufnahmetechnik
- Binaurale Aufnahmetechnik
- Arraytechnologie

Ein wesentlicher Punkt beim Aufnahme Equipment ist auch dessen Halterung bzw. Anbringung im Cockpit. Während die Anbringung und Befestigung eines Einzelmikrofons noch eine relativ einfache lösbare Aufgabe darstellt, geben sich bei den anderen Verfahren, vor allem bei der Verwendung von Aufnahmen bei Linienflügen größere Hürden. Als eine Innovative Entwicklung kann das Halterungssystem für ein Mikrofonarray, welches vom IEM zusammen mit dem MBI entwickelt wurde, genannt werden.



**Abbildung 3:** Speziell entwickeltes Mikrofonarray (IEM) und Halterung (IEM&MBI)  
für den Einsatz bei Linienflügen im Flugzeugcockpit einer Fokker 100.

### 4 Analyse

Der Fokus bei der Beschreibung der Schallfeldsituation im Cockpit liegt auf der Quellenseparation und in folge auf deren nähren Strukturanalyse. Zu den wichtigsten globalen Geräuschkomponenten zählen:

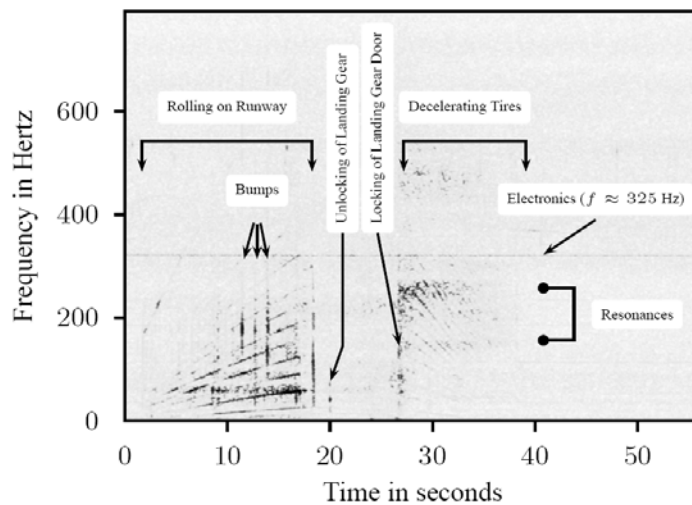
- die Flugzeugform
- das Triebwerk
- das Fahrwerk und
- die Umgebung - Environment (Wetter & Terrain)



Der Hauptaufgabe der Geräuschanalyse ist die Trennung von stochastischen (geräuschartigen) und deterministischen (harmonischen) Komponenten und deren Zuordnung bzw. Einordnung in den Flugzustandsraum und deren Zuordnung von bestimmaren oder unbestimmaren Schallereignisorten. Der Flugzustandsraum wird durch relevante Flugdaten wie: Drehzahlen der Triebwerke, Flughöhe, Fluggeschwindigkeit, Orientierung, Stellung der Flaps usw. beschrieben.

Dabei werden folgende Aspekte zusätzlich unterschieden:

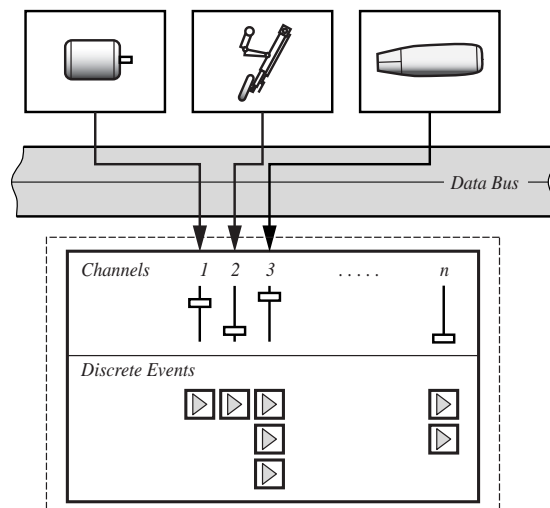
- Zeitliche Abläufe
- Spektrale Unterscheidung
- Räumliche Separation / Zuordnung



**Abbildung 4:** Zeitliche und spektrale Geräuschbeschreibung.

## 5 Synthesestrategien

Grundsätzlich kann die Synthese nicht unabhängig von der Analyseseite betrachtet werden. Somit erfolgt diese nach dem Konzept der „Quelle zugeordneten Geräusche“ wie bereits oben beschrieben wurde.



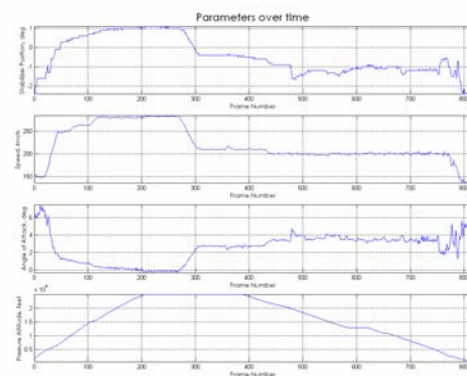
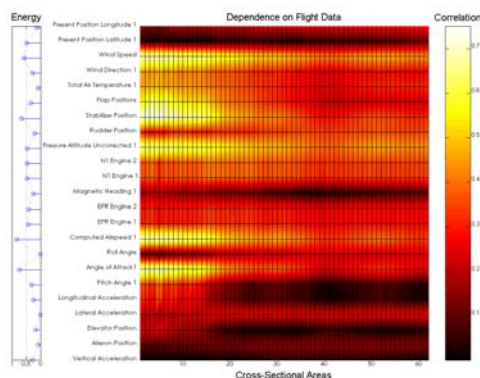
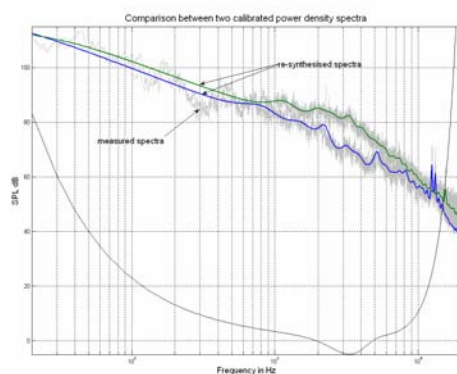
**Abbildung 5:** Quellenmodulares Synthesekonzept

Im vorliegenden Projekt ergibt sich die zusätzliche Erschwernis, dass im realen Cockpit die Aufnahmen nicht für alle möglichen Zustände im Flugzustandsraummodell – wie diese bei einem Flugsimulator erwartet werden - erhoben werden können (bedingt durch extremen zeitlichen Aufwand und auch durch Sicherheitsgründe). Diese Zustände müssen im Sinne der Level D Vorgaben weitest gehend aus den vorhandenen Daten extrapoliert werden oder aber auch sinngemäß generisch synthetisiert werden.

Basierend auf den Aufnahmedaten und den extrahierten Features können z. T. quasi-stationäre Schalle z.B. durch einfache Quelle-Filter Modelle nachgebildet werden. Dabei ändert sich zeitlich nur das Filter jedoch das Quellen- oder Generatorsignal bleibt unangetastet.

### Bsp: Strömungsgeräusch

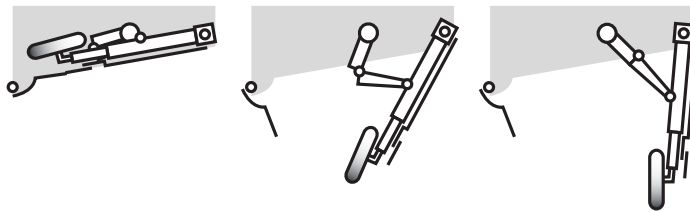
- Nachbildung durch gefiltertes Rauschen
- Filterkurven parametrisiert durch:
  - Flughöhe (Pressure Altitude)
  - Anstellwinkel (Angle of Attack)
  - Geschwindigkeit entlang der Flugbahn (True Airspeed)
  - Position der Höhenflosse (Stabilizer Position)



**Abbildung 6:** Untersuchungsbeispiel anhand des Strömungsgeräusches

Oben links: Analyse und Resynthese mit LPC, Oben rechts: Quellenzuordnung - Airframe  
 Unten links: Zusammenhang LPC Koeffizienten (bzw. Querschnittsfl.) mit Flugzustandsdaten  
 Unten rechts: Auszug aus den relevanten Flugdaten für einen Flug Wien-München.

Außerdem müssen Zustandsraum (relativ) unabhängige Ereignisse, die einem speziellen zeitlichen Verlauf unterworfen sind wiederum anders behandelt werden (z.B. Öffnen der Fahrwerksklappen und Ausfahren des Fahrwerks durch z.B. flugdatengesteuerte Sampleplayer)

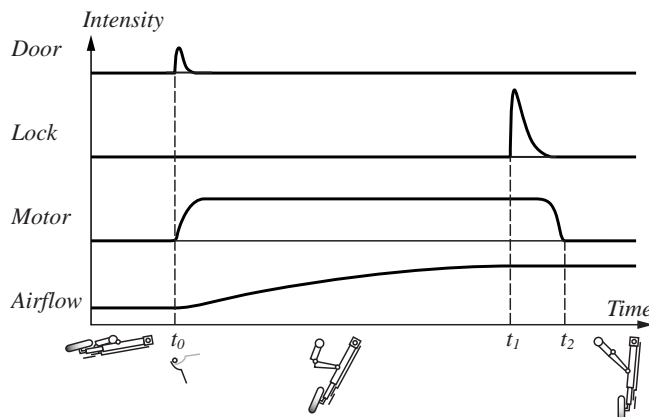


#### Mechanisches Schema für Fahrwerk

Fahrwerksklappe

Fahrwerksstellmotor

Fahrwerksverriegelung



#### Akustische Ereignisse (Normalfall)

Fahrwerksklappe öffnet sich.

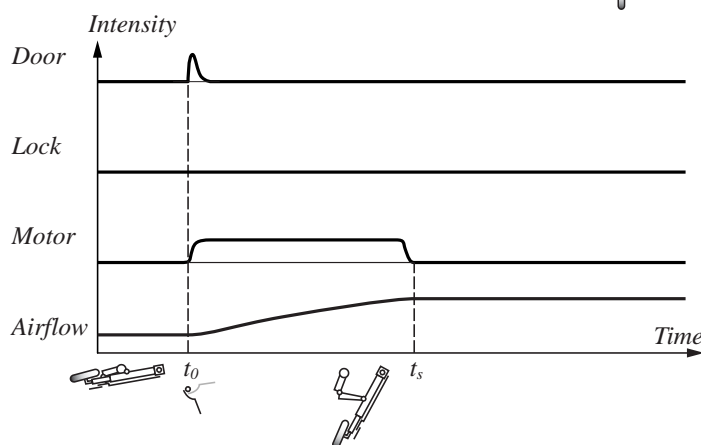
Fahrwerksstellmotor wird aktiviert.

Strömungsgeräusch (nicht diskret!) steigt an.

Fahrwerksverriegelung bei Erreichen der Endposition.

Fahrwerksstellmotor wird deaktiviert.

Strömungsgeräusch bleibt erhöht.



#### Akustische Ereignisse (Störfall)

Fahrwerksklappe öffnet sich.

Fahrwerksstellmotor wird aktiviert.

Strömungsgeräusch (nicht diskret!) steigt an.

Fahrwerksstellmotor ist defekt erreicht nicht die Endposition f. Fahrwerk.

Strömungsgeräusch bleibt erhöht.

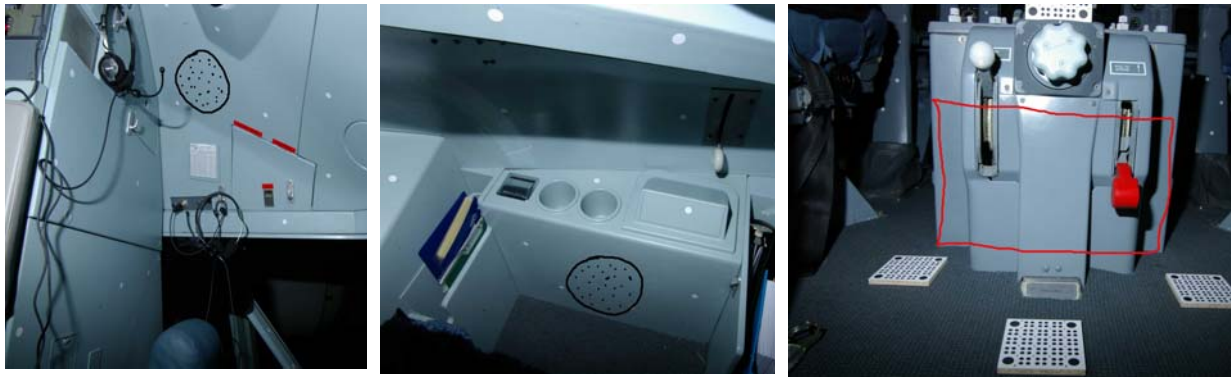
**Abbildung 7:** Beispiel zu Zustandsraum unabhängige, zeitdiskrete Geräuschevents anhand des Fahrwerks.

*Strömungsgeräusch (Airflow) ist natürlich kontinuierlich, wird aber auch durch diesen Event auch modifiziert.*

## 6 Spatialisierung

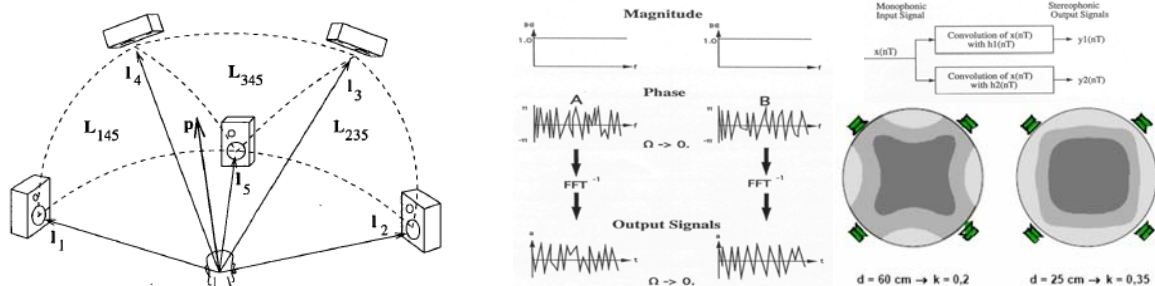
Das Schallfeld in der Kabine setzt sich aus richtungsunabhängigen (Druckkabine) und richtungsabhängigen Komponenten (lokalisierbare Quellenrichtung) zusammen.

Für die Wiedergabe im Flugsimulator sind mindestens 8 Lautsprecherpositionen (unabhängige Wiedergabekanäle) vorgesehen. Diese werden an optisch möglichst unauffälligen Positionen maximal effizient angebracht.



**Abbildung 8:** Einzelne voraussichtliche Lautsprecherpositionen.

Da die Kabine beim Flugsimulator nur bedingt identisch nachgebaut wird (z.B. Instruktorplatz hinter dem Cockpit), kann bei der akustischen Nachbildung nicht von einer gleichen Wiedergabebedingung wie im realen Cockpit ausgegangen werden. Daher kommen unterschiedliche Wiedergabestrategien zum Einsatz. So werden z.B. breitbandige Geräusche die keine deterministische Abbildungsrichtung besitzen durch eine Kombination von vollständig korreliert (Druckkammer – für tiefe Frequenzen) und weitestgehend dekorreliert (Einhüllendes Ambiente – für mittlere bis hohe Frequenzen) extrahierten Geräuschkomponenten zusammengesetzt.



**Abbildung 9:** Spatialisierungsstrategien

Links: lokalisierbare Geräusche werden anhand von unterschiedlichen Lautstärkeverhältnissen nachgebildet (VBAP).

Rechts: Räumlich globale ausgedehnte Geräusche werden durch Dekorrelation im Raum spatialisiert.

## 7 Implementierungsvorgaben

Die Implementierung erfolgt unter PD (pure data) auf einem handelsüblichen PC unter Linux mit einer RME Soundkarte.

Das Sound-Modul muss auf ein angefordertes Geräusch (bedingt durch andere Module/Bedienelemente) vor Ablauf einer gewissen Zeit mit der Ausgabe eines entsprechenden Geräusches reagieren können. Die Anforderungen bzgl. minimaler Systemlatenz sind durch die Synchronisierung mit der Motion Base vorgegeben und betragen somit 31ms.

Die minimale Systemlatenz setzt sich aus folgenden Größen zusammen:

- Garantierte obere Schranke des UDP für Send-Receive: 5ms
- Verzögerung des Videosystems: 10ms
- Verzögerung der Motion Base zu Videosystem: 16ms

## **8 Dokumentation**

Die Dokumentation erfolgt in Teildokumenten parallel zu den einzelnen Arbeitsschritten, wobei diese neben den üblichen schriftlichen Aufzeichnungen, Vereinbarungen, Konzepte und Lösungsstrategien auch aus originalen und strukturierten Schallaufzeichnungen, Softwareskripts (Source Code), Fotos, Filme und dergleichen bestehen.

Projektfertigstellung: 30. September 2005

15. September 2005

## **„Forschungsplattform Flugsimulation“ Präsentation einer interuniversitären Forschungsinitiative**



### **Gesprächspartner:**

O.Univ.-Prof. Dipl.-Ing. Dr. Hans Sünkel  
Rektor der TU Graz

Dipl.-Ing. Leopold Schöggel  
Landeshauptmann-Stellvertreter

Univ.-Prof. Dipl.-Ing. Dr. Siegfried Vössner  
Projektleiter, Institut für Maschinenbau- und Betriebsinformatik der TU Graz

# **„Forschungsplattform Flugsimulation“**

## **Präsentation einer interuniversitären Forschungsinitiative**

**Der Traum vom Fliegen ist an der TU Graz gelandet: Im Rahmen einer universitätenübergreifenden Plattform unter Federführung des TU-Instituts für Maschinenbau- und Betriebsinformatik entwickeln Wissenschaftler einen Flugsimulator, der in Forschung und Lehre Verwendung finden soll.**

Von der mechanischen Modellierung der Flugdynamik über Untersuchungen von Piloten unter Stresseinfluss bis hin zur dreidimensionalen Sound-Simulation von Fluggeräuschen reicht die breite Palette an Themen, zu denen am Flugsimulator der TU Graz geforscht wird. Disziplinenübergreifende Thematiken brauchen eine disziplinenübergreifende Form der Zusammenarbeit: Die beteiligten Wissenschaftler schließen sich daher zur interuniversitären „Forschungsplattform Flugsimulation“ zusammen und wollen Forschung und Lehre in diesem Bereich gemeinsam vorantreiben.

„Mit der 'Forschungsplattform Flugsimulation' bauen wir die Kooperation der führenden Forschungsinstitute des Landes Steiermark untereinander und mit High-Tech-Unternehmen weiter aus“, begrüßt TU-Rektor Hans Sünkel die Initiative. Beteiligt sind bisher neben der TU Graz die Karl-Franzens-Universität Graz und die Kunstuniversität Graz. Neben namhaften heimischen Unternehmen gelang es mit dem Triebwerkshersteller Rolls Royce auch einen internationalen Player mit an Bord zu holen.

Das Land Steiermark, Abteilung für Wissenschaft und Forschung, fördert das Projekt mit insgesamt 400.000 Euro. Landeshauptmann-Stellvertreter Leopold Schögggl: „Wir freuen uns, dass die Steiermark damit neben dem automotiven Bereich und der Weltraumforschung nun auch im Bereich Luftfahrt weiter zur Spitze aufschließt. Die wissenschaftlichen Erkenntnisse werden sowohl Forschern und Studierenden, als auch den beteiligten Unternehmen zugute kommen und so einen entscheidenden Wettbewerbsvorteil für die Steiermark erzielen.“

### **„Maximale Realitätsnähe“: Simulator für ein voll verkehrstaugliches Flugzeug**

„Wir wollen ein Simulationssystem höchster Realitätstreue schaffen, das auch die strengsten internationalen Luftfahrtbestimmungen erfüllt“, formuliert Projektleiter Siegfried Vössner, Vorstand des Instituts für Maschinenbau- und Betriebsinformatik der TU Graz, das Projektziel. Die von ihm und Mechanik-Professor Reinhard Braunstingl ins Leben gerufene universitätenübergreifende Forschungsinitiative bringt dazu Vertreter verschiedener ingenieurwissenschaftlicher Disziplinen, Sounddesigner und auch Psychologen zusammen.

An der TU Graz wird dazu für Forschung und Lehre ein voll funktionsfähigen Flugsimulator eines Verkehrsflugzeuges vom Typ Fokker 100 zur Verfügung stehen. Die wichtigsten Kernmodule und Hardware-Komponenten des Flugsimulators sind bereits vor Ort. Das vollständige Cockpit mit authentischen Bedienelementen folgt in den nächsten Monaten. „Die 'Forschungsplattform Flugsimulation' verfügt dann über einen Flugsimulator der höchsten Ausbau- und Entwicklungsstufe, der entsprechend internationaler Richtlinien Anspruch auf 'maximale Realitätsnähe' legen kann“, merkt Professor Vössner an. Keine andere Hochschuleinrichtung in Österreich verfügt derzeit über einen ähnlich hoch entwickelten Simulator, die notwendigen Simulations-Algorithmen werden an Vössners Institut gemeinsam mit dem TU-Institut für Mechanik entwickelt.

Das Institut für Elektronische Musik und Akustik der Kunstuniversität Graz steuert ein echtzeitfähiges Soundsystem bei, welches sich der neuesten Technologien zur vollständigen Raum-Zeit-Erfassung von Schallfeldern bedient. Zusammen mit dreidimensionalen Lautsprecherwiedergabetechniken kann so ein räumlicher Höreindruck realitätsnah vermittelt werden. Erste „Hörproben“ sind bereits möglich.

## **Spielwiese für diverse Disziplinen: Ingenieurwissenschaften, Musik und Psychologie**

Der Forschungsbereich Flugsimulation stellt große Herausforderungen an verschiedenste Disziplinen. An der TU Graz widmet man sich etwa der mechanischen Modellierung der Flugdynamik, dem Software Engineering oder der Triebwerksmodellierung. Experten der Kunstuniversität arbeiten unter anderem an einer realistischen dreidimensionalen Geräuschsimulation von Strömungsgeräuschen bei unterschiedlichen Flugzuständen und Psychologen der Karl-Franzens-Universität untersuchen Leistungsfähigkeit und Reaktionsgeschwindigkeit unter dem Einfluss verschiedener Stressbedingungen beim Fliegen.

„Das Thema des Forschungsprojekts übt eine große Anziehung auf Studierende und Wissenschafterkollegen gleichermaßen aus, die alle hoch motiviert mit dabei sind“, freut sich Vössner. „Simulation und insbesondere Flugsimulation erweist sich als hervorragende Forschungsplattform, sowohl interdisziplinär als auch intradisziplinär, also in den jeweiligen Spezialgebieten.“ Die TU Graz und die beteiligten Institute der Kunstuniversität Graz und der Karl-Franzens Universität Graz setzen den Flugsimulator bereits jetzt im Rahmen von Lehre und Forschung ein. „Dies wird sich im Laufe des Projektes noch deutlich steigern“, ist Vössner überzeugt.

### **Bisher beteiligte Universitäts-Institute:**

TU Graz:

Institut für Maschinenbau- und Betriebsinformatik

Institut für Mechanik

Kunstuniversität Graz:

Institut für Elektronische Musik und Akustik

Karl-Franzens-Universität:

Institut für Psychologie

### **Beteiligte Firmen und Projektspensoren :**

AXIS Flight Systems

Brightline

AMD

nVidia

pny

NEC

Rolls Royce

### **Rückfragen:**

Mag. Alice Senarclens de Grancy

TU Graz - Medienarbeit

Tel 0316 873 6006

Mobil 0664 60 873 6006

Email [alice.grancy@TUGraz.at](mailto:alice.grancy@TUGraz.at)

<http://presse.TUGraz.at>



## A NEW FUNCTIONAL FRAMEWORK FOR A SOUND SYSTEM FOR REALTIME FLIGHT SIMULATION

*Siegfried Vössner, Reinhard Braunstingl*

Dept. of Engineering- and Business Informatics,  
Department of Mechanics  
Graz University of Technology  
Kopernikusgasse 24/III, A-8010 Graz, Austria  
{voessner, r.braunstingl}@tugraz.at

*Helmuth Ploner-Bernard, Alois Sontacchi*

Institute of Electronic Music and Acoustics  
University of Music and Dramatic Arts  
Inffeldgasse 10/III, A-8010 Graz, Austria

{ploner, sontacchi}@iem.at

### ABSTRACT

We will show a new sound framework and concept for realistic flight simulation. Dealing with a highly complex network of mechanical systems that act as physical sound sources the main focus is on a fully modular and extensible/scalable design. The prototype we developed is part of a fully functional Full Flight Simulator for Pilot Training.

### 1. INTRODUCTION

Advances in Computer Technology and modeling make it possible to simulate complex systems and machines for design or training purposes.

Education Programs in Aviation rely heavily on simulation based training. The higher the degree of simulated reality the higher are the requirements for realistic simulation of the aircraft, its motion, vision, and sound. The American (FAA) and European (JAA) aviation authorities define various categories of flight simulators, where the highest quality level of flight simulation is represented by a so-called *Level D* simulator [1]. In such a *full-flight* simulator training lessons in the simulator can replace all lessons in a real aircraft when experienced pilots have to learn to fly a particular aircraft type – and especially to handle abnormal situations. Especially in emergency situations the sound at the flight deck may be a highly disturbing factor for the flight crew and should be simulated during flight training as realistic as possible.

In this paper we show a new approach for a realistic and fully modular sound concept for such a full-flight simulator based on real sound data enriched with digital audio effects. This paper will address mainly the underlying system design and principles. The topics *data acquisition*, *feature extraction*, *sound synthesis*, and *3D rendering* and their challenges are being discussed in an additional specialized paper [2].

### 2. EXISTING CONCEPTS

Sound-Systems are an important component of flight simulators because of their strong contribution to creating the illusion of realism. However the focus of traditional system design is mainly on flight mechanics and functionality of aircraft systems. Also, the certification guidelines for flight simulators do not specify the requirements for the sound systems with the same precision as for

other aircraft systems. As a result, the realization of sounds lags far behind the other functional modules of a flight simulator. The use of modern sound recoding, analysis and rendering techniques is very limited [3][4][5][6].

### 3. FLIGHT SIMULATION FRAMEWORK

In order to understand the basic design requirements for a flight simulation sound system it is necessary to understand the basic building blocks of a flight simulator:

- Input/Output Devices
  - Cockpit Instruments
  - Controls (Thrust Lever, Flight Controls, Pedals,...)
- Aircraft-Systems Representation
- Flight Dynamics
- Motion Platform
- Sound System
- Vision System
- Instructor Station

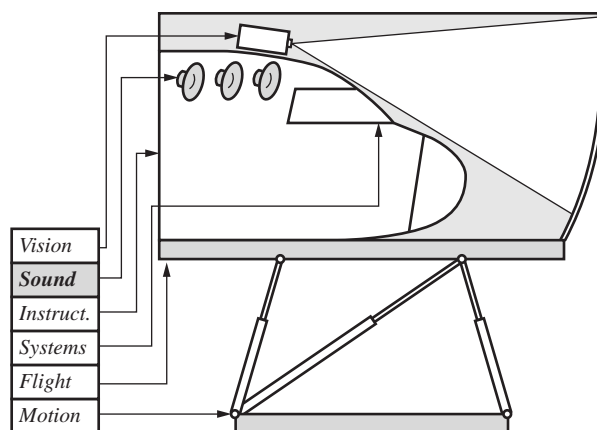


Figure 1: System components of a flight simulator.

Whereas all cockpit components and controls are exact, physical replicas or even genuine aircraft parts, the underlying simula-

tion engine is a complex computer program based on analytical models and flight data of the real aircraft.

The Motion-, Sound- and Vision System are part of the human interface to the pilot and mainly responsible for creating the illusion of realism. A system configuration which is common to full-flight simulators is shown in figure 1. Usually each module runs on a separate computer for performance reasons, which is indicated by the rack-type setup.

### 3.1. Aircraft System Model and Components

The aerodynamic model is the heart of the flight simulator. It represents the aircraft as a rigid body exposed to aerodynamic-, propulsion- and other forces induced by the environment as shown in Figure 2. Usually all effects of elasticity or multibody interaction are delegated to the respective models (gear, aerodynamics, fuel sloshing) [8].

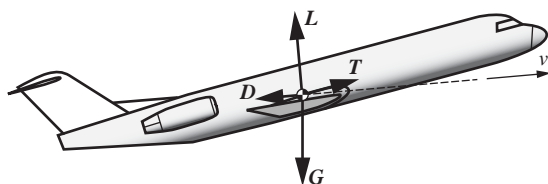


Figure 2: Elemental forces (Gravity  $G$ , Lift  $L$ , Drag  $D$ , and Thrust  $T$ ) and resulting speed of an aircraft.

### 3.2. Discrete Time Simulation and Hardware Components

The equations of motion represent a system of second order differential equations. Normally they are reduced to first order and solved for discrete and constant time intervals of 20 ms or less by numerical integration [7] in order to simulate the aircraft's motion in three-dimensional space.

A controller program generates at that clock rate events at which the current simulation inputs (controls, environment, ...) are read and flight-mechanics related outputs like position, attitude, speed, and other data are being computed. (Simulation of other, slowly reacting aircraft systems is done by particular models using bigger time steps.) Data are shared over a common high-speed data bus.

Hardware concepts for flight simulators vary – however, it is common to separate the computation of flight dynamics, aircraft systems, instrument/control I/O, motion, vision and sound for performance reasons as can be seen in Figure 1. It must be shown during certification of a flight simulator that the data flow through the distributed system shows a time-lag within given tolerances.

## 4. SOUND FRAMEWORK

Due to the structure of the simulated system, which is from our point of view a highly complex network of sound emitting mechanical systems, it is necessary to develop a fully modular design. This allows on one hand to manage the sound complexity and ensures scalability on the other hand.

### 4.1. Requirements

As it is true for all sound simulation systems, the basic requirement is to generate a realistic sound impression in real time. When used in flight simulation for pilot training purposes it is especially important to model sounds that are being created by operation and failure of the various aircraft system components. It is therefore not sufficient to generate an excellent sound impression as a whole but rather required to model all effects individually.

Another requirement is to generate a realistic 3D sound impression in a rather small cockpit replica in terms of direction and characteristics of sounds. This requires careful design of the sound rendering algorithms as well as of the positioning of the speakers due to space constraints.

The most important requirement however is full modularity to allow for easy configuration, exchange, and testing of system components.

### 4.2. Design Principles

**Discretization:** The sound simulated is considered to be the sum of various discrete sources that create sounds based on their internal state or operation.

**Superposition:** The discrete sources are combined by use of superposition to generate the resulting sound. Effects generated by interference of the individual sounds have been evaluated and considered being neglectable.

**Local Sound Creation:** Sound creation in the discrete sources (like aircraft system components) depends on their sound generating interaction with the environment. This interaction is determined locally by internal mechanics and functionalities and can be described with a set of relevant sound parameters. These sound parameters can efficiently only be generated in the discrete sources themselves – not outside.

**Central Sound Rendering:** Contrary to sound generation, sound rendering can only be done in a central module that combines all sounds from all the sources and renders them for 3D output by taking into account position of the individual sources and sound transmission functions with respect to the cockpit.

**Modularization:** All systems modeled in the simulator are seen as individual modules having a clearly defined functionality and interface to the rest of the simulation system. Real aircraft systems can be exchanged provided that they have the same functionality (e.g. it is possible to change the engines). In order to provide that same possibility in a flight simulator it is necessary to store all module specific data – mechanical as well as sound data needed for rendering – in the module context.

### 4.3. Sources and Types of Sounds

There are multiple sources of sounds that can be observed from an aircraft's cockpit. The most dominant ones are being caused by noise due to air current around the plane's airframe and by the engines. Depending on the flight maneuvers some additional sources become dominant as well. An example would be the landing gear during taxiing on the ground or additional air-current noise caused by extended flaps. Figure 3: shows the most important sources of sounds in an aircraft.

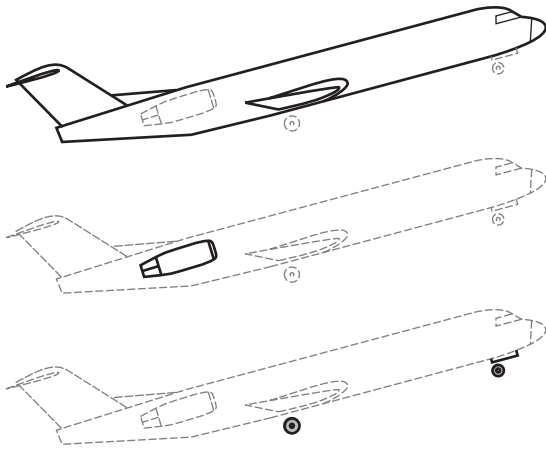


Figure 3: Sound sources in an aircraft: airframe, engine, landing gear.

We define two basic types of sounds that can occur:

**Discrete Sounds (event driven):** Sounds that are generated by discrete system events and can not be interrupted by the simulation are called "discrete sounds". They can incorporate the sound of a sequence of sub-events and may last for several seconds. An example for a discrete sound would be the (big) bang the landing gear makes at touch down.

**Continuous Sounds (permanent):** Sounds that are generated by continuous processes and that can be changed/interrupted during the simulation are called "continuous sounds". Continuous sounds represent the corresponding sound to the current system state. Air-current noise is a good example for this type: Depending on the aircraft's altitude, airspeed, angle of attack, flaps and landing gear configuration and other environmental conditions, the airframe will continuously generate noise that varies in intensity and spectral characteristic.

All other sounds, even of complicated nature can be realized using these two building blocks. To illustrate this let us consider the following example: Approaching an airport for landing the pilot pulls the lever for lowering the landing gear, thus triggering the following sequence of events: The latches in the hull open, a hydraulic motor lowers the landing gear. The resulting air resistance creates additional air-current noise depending on the degree the gear is extended. If the landing gear has reached the final position, the mechanics locks with a possibly loud rumble so that the gear is secured in place (Figure 4).

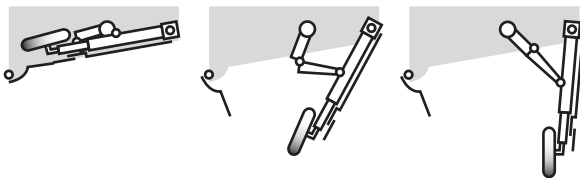


Figure 4: Example extension of landing gear.

We model the corresponding sound sequence with two continuous and two discrete sounds. Continuous sounds are:

- Air-current noise which is (simplified for this example) determined by the parameters airspeed and the landing gear extension parameter (0-100%)
- Hydraulic motor noise, the characteristic of which depends solely on the motor speed (0-100% of max. rpm) for both, repeating sound patterns – similar to a starter engine in an automobile – and fundamental frequency. Let us assume it takes the motor about 5 seconds to lower the gear fully.

The following two events are modeled as discrete sounds:

- Latch opening creates a non interruptible characteristic noise that lasts about 0,5 seconds.
- Gear locking causes a fairly loud and short rumble.

According to our design principles the computation of the sound influencing parameters like airspeed, motor speed, and gear extension is done within the individual modules. The same is true for the triggering of the discrete events. Figure 5 puts the sound events on a simulation timeline and shows the corresponding mechanic situation.

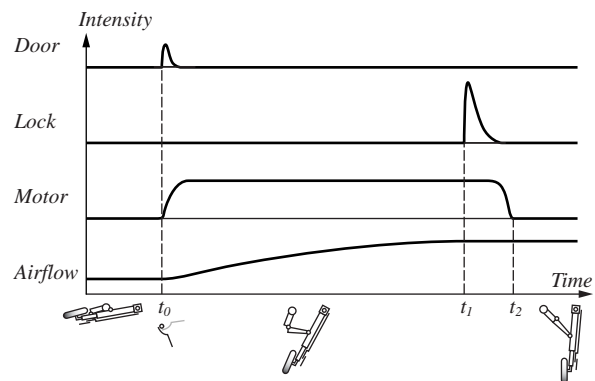


Figure 5: Landing gear simulation timeline.

To demonstrate the modularity of the concept, let us now assume the following mechanical landing gear extension failure: the hydraulic motor spins up to 50%, stays there for a second and gets stuck. As a result the landing gear will be extended to about 60%. What would be the effect on our sound simulation?

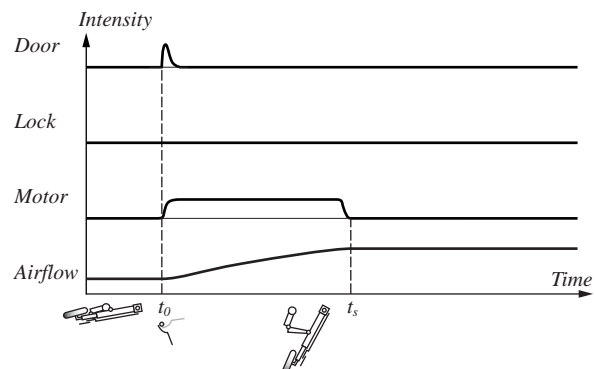


Figure 6: Example extension of landing gear gets stuck.

The latches will open with the same sound as before. Since the motor is modeled as a continuous sound device it would create the sound that corresponds to its speed. Because its speed will not exceed 50% of maximum speed, the sound simulation will mimic exactly this behavior. Getting stuck means for the motor that it will stop creating sounds (speed=0). The landing gear extension parameter provided by the corresponding simulation module will also stop at 60%. As a result the noise caused by air current will be a little less (gear extension is a sound parameter) and the gear locking bang will not happen, since the landing gear is stuck at 60% (Figure 6).

This shows an important feature of a sound system framework that closely models the mechanical behavior of the real system: Should it for example become necessary to incorporate other system failures – especially ones that generate audible effects (like explosions) these could easily be incorporated: One would have to add the failure logic to the respective module and add the desired sound either as additional discrete or continuous sound.

The advantages compared to other approaches which create a strong linkage between the logic of sound creation and sound rendering are simplicity and independence from knowledge of system internals.

4.4. Core Components

The core elements of the sound system are:

- **Modules** representing system components with system logic, mechanical and audio data.
- **Sound Rendering Engine** which has two closely related functions:
  - o Generating the individual sounds of all system components by using their defined sound parameters and data.
  - o Creation of a virtual three dimensional sound field by placing the generated sounds on their position with respect to the cockpit. In our case we assume that all our system components have a constant position during the entire simulation. This simplifies the rendering algorithm since they can pre-compute structure damping and other correction factors.
- **Sound Generator** produces the individual sound signals for the built in speaker system in the simulator cockpit taking into account the speaker configuration and special hardware setting.

The last two core elements usually run on a separate computer for performance reasons.

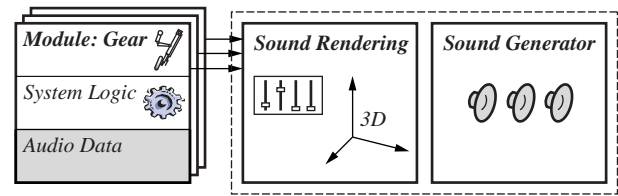


Figure 7: Core components.

Figure 7 shows the three core components with their functionality and data attached.

4.5. Integration in the Flight Simulation Framework

In order to integrate the sound simulation system seamlessly into the entire flight simulation framework we have chosen to use the same data bus that is common to all other modules. By design this data bus ensures a reliable and fast communication and satisfies the requirements for real time simulation.

Figure 8 shows the interaction of simulation modules and Sound Rendering Engine over the data bus.

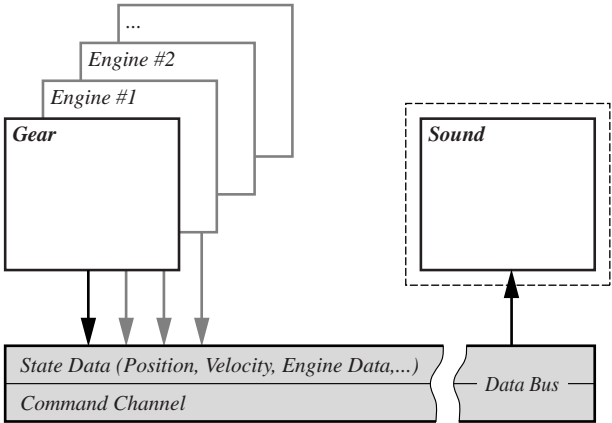


Figure 8: Interaction between Modules and Sound Rendering Engine.

At initialization the Sound Rendering Engine registers all sound emitting system components and loads the list of sounds and their parameters. For the simulation of the module “hydraulic motor” from the example above this would mean that the module registers one continuous sound with motor speed as the only parameter and an optional “bang” as a discrete sound with one parameter for the intensity of the effect. The module also provides all data like position information and component specific sound data necessary to render the sound. Figure 9 shows this for the “hydraulic motor” example.



Hydraulic Motor					
					
System Logic					
Audio Data					
Continuous			Discrete		
#	Name	Parameter	#	Name	Parameter
1	Motor RPM	n	1	Explosion	Volume
#	Data		#	Data	
1	Sound(n)		1		

Figure 9: Sound Data in System Modules.

Once initialized, the Sound Rendering Engine gets all information necessary to create the continuous sounds by listening on the data bus for the respective sound parameters. Discrete events require in addition a special trigger message containing event number and optional parameters. Figure 10 shows the principle of sound generation within the flight simulator framework.

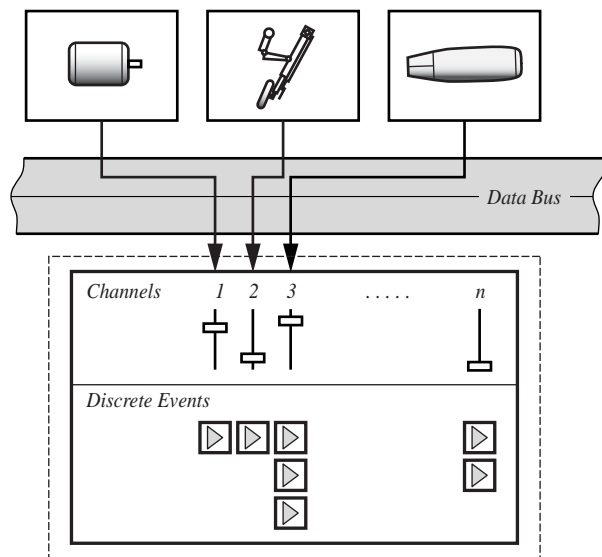


Figure 10: Multi-Channel Transmission of Sound Information.

Within this context it is important to point out that continuous sounds are *continuously on* which means that not only the airframe noise is generated permanently by observing the data bus for the relevant sound parameters (airspeed and gear extension in our example) but also the sound of the hydraulic motor is *on* – however, creating audible sound only if the motor speed is greater than zero. This can be compared to a professional studio mixing equipment where many channels are attached but only a few are opened.

With a limited number of noise creating system components, this approach has the advantage that the sound system can react immediately to changes of system state. The computational overhead for rendering zero-signals can of course be reduced by filtering only for non-zero channels.

#### 4.6. Data Acquisition and Signal Processing

The sound data used in our research project are based on elaborate audio recordings and flight data analysis from real aircraft. Techniques and tools are not discussed as part of this paper [2].

### 5. CONCLUSIONS AND FURTHER WORK

As a proof of concept we have applied the framework laid out in this paper to designing a sound system prototype for a Full Flight Simulator for a particular airplane as a joint project with industry. Despite some technical and time constraints we were able to realize the core components and functionality of our framework.

The sound prototype was tested with airline pilots – some flying that very airplane on a daily basis and got high acceptance ratings. It turned out the additional effort in sound modularization and discretization leads to a sophisticated, detailed sound simulation, which is a major advantage in comparison to state of the art sound systems for flight training devices.

We will investigate further possibilities in modularization of this concept in the future and try to applying it to other areas like automotive.

### 6. ACKNOWLEDGEMENTS

We would like to thank Austrian Airlines for their support as well as the Styrian Government for supporting this project as part of a research grant (GZ: A3-10 F 59-04/1). We furthermore thank our project team and especially Gerald Lichtenegger for valuable discussions and contributions.

### 7. REFERENCES

- [1] Federal Aviation Administration, *Airplane Simulator Qualification*, Advisory Circular 120-40B, 29 July 1991.
- [2] H. Ploner-Bernard, A. Sontacchi et al., "Sound-System Design for a Professional Full-Flight Simulator", submitted to DAFX'05, Madrid, Spain, Sept. 2005.
- [3] United States Air Force. "An Introduction to Structural Models" (USAF ASC-TR-93-5008). Dayton, OH: Wright-Patterson AFB, 1993.
- [4] L. Howard, L. Bass "Structural Modeling for Flight Simulators," pp. 876-881. *Proceedings of the 1993 Summer Computer Simulation Conference – SCSC '93*. Boston, MA: July 1993.
- [5] G. Abowd, L. Bass, L. Howard, L. Nomaop, "Structural modeling: An application framework and development process for flight simulators", Tech. Report CMU/SEI-93-192, *Software Engineering Institute, Carnegie Mellon University*, August 1993.
- [6] B.T. Sullivan, P.A. Soukup, "The NASA 747 -400 flight simulator: National resource for aviation safety research" presented at *American Institute of Aeronautics and Astronautics*, San Diego, CA (AIAA-96-3517), July 1996.
- [7] B. Stevens, F. Lewis, "Aircraft Control and Simulation", *J. Wiley & Sons*, 2004.
- [8] P. H. Zipfel "Modeling and Simulation of Aerospace Vehicle Dynamics", *AIAA Education Series*, 2000.



## SOUND-SYSTEM DESIGN FOR A PROFESSIONAL FULL-FLIGHT SIMULATOR

*Helmuth Ploner-Bernard, Alois Sontacchi*

*Gerald Lichtenegger, Siegfried Vössner*

University of Music and Dramatic Arts Graz  
Institute of Electronic Music and Acoustics  
Inffeldgasse 10/III, 8010 Graz, Austria  
{ploner,sontacchi}@iem.at

Graz University of Technology  
Dept. of Engineering- and Business Informatics  
Kopernikusgasse 24/III, 8010 Graz, Austria  
{gerald.lichtenegger,voessner}@TUGraz.at

*Reinhard Braunstingl*

Graz University of Technology  
Department of Mechanics  
Kopernikusgasse 24/III, 8010 Graz, Austria  
r.braunstingl@TUGraz.at

### ABSTRACT

In this paper, we present a sound system to be integrated in an accredited realistic full-flight simulator, used for the training of airline pilots. We discuss the design and implementation of a corresponding real-time signal-processing software providing three-dimensional audio reproduction of the acoustic events on a flight deck. Here, the emphasis is on an aircraft of a specific type. We address issues of suitable data acquisition methods, and, most importantly, of functional signal analysis and synthesis techniques.

### 1. INTRODUCTION

In a multidisciplinary research project, we are designing a flight simulator the behavior of which should be as realistic as possible. In this context, we aim at meeting the strict requirements of the highest international standard for full-flight simulators [1].

As can be seen in Figure 1, the flight simulator includes a realistic copy of the flight deck mounted on a motion base. Within certain limits, this allows for movement in all three dimensions. The visual illusion is provided by a projection system with an appropriate vision dome. Last but not least, the sound system is responsible for a lifelike reproduction of the sound field.

#### 1.1. Embedding of Sound System in Flight Simulator

Topologically, the flight-simulator system consists of a network of independent modules managed by a central control unit (Figure 1). In general, these modules compute output parameters from a set of relevant input parameters (computation of the flight path; computation of the behavior of the engines). Conversely, the task of the sound-system module is the reproduction of a sound-field in reaction to input parameters. These input parameters may be continuous (true air speed, height), or binary (ground contact on landing gear).

More details on the concept and the structure of the sound system can be found in [2].

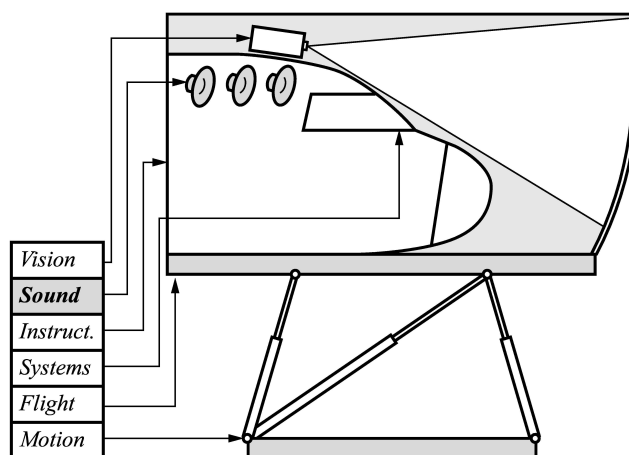


Figure 1: Scheme of Flight Simulator, including Sound System

#### 1.2. Audio Hardware

In our sound reproduction system we employ half a dozen high-quality full-range wall loudspeakers mounted inside the cockpit. We have chosen their number and positioning in conformity with constructional as well as with acoustical requirements. The loudspeakers are distributed as uniformly as possible across the walls of the flight deck of the simulator.

In addition, the sound system includes a powerful subwoofer in order to intensify the illusion created by the simulator.

When choosing the audio hardware for the sound system, we had to consider that the simulator's motion base can produce high accelerations (up to 4 g in all three directions), which the loudspeakers will be exposed to.

#### 1.3. Software and Computer Hardware

The signal processing part of the sound module is implemented using the platform-independent open-source software *Pure Data*,

a graphical programming environment for real-time audio signal processing developed by Miller Puckette [3]. The sound simulation runs on today's standard PCs.

#### 1.4. Objective and Subjective Tests

By means of objective measurements we will demonstrate that the sound produced by our system complies with the situation in a real cockpit in regard to (1) true sound pressure level, (2) frequency range, (3) localization of discrete sources, and (4) degree of spatial correlation.

Furthermore, we are continually validating our results with experienced pilots in order to match the overall sound experience presented by the simulator as closely as possible to the situation in a real cockpit.

## 2. ACOUSTIC EVENTS

In this section, we describe the most significant sounds that typically occur in the flight deck of an aircraft. On the basis of quite simplistic models, we deduce potential dependencies on physical parameters. In the target simulator, such parameters will constitute the input into the modules that are responsible for the creation of the different sounds. However, the sound of the aircraft of the specific type we are simulating need not depend on all parameters presented here. Actually, some sounds cannot even be heard at all.

We know from experience that most of the parameters change rather slowly (on the order of 100 ms). As a consequence, the sounds can be considered (quasi)stationary.

### 2.1. Continuous Sounds

Some acoustic events on the flight deck depend on parameters that are updated continuously in the simulation, i. e. at fixed time intervals. Such acoustic events are called "continuous sounds". This does not imply, however, that continuous sounds are always audible during normal flight conditions. The continuous sounds are:

#### 2.1.1. Noise Due to Air-Current

In principle, the shape of the airframe of the aircraft does resemble streamlines. However, the shape is not perfectly smooth, but divided into sub-sections for technical reasons, e. g. the glasses of the front windshield. So, whenever there is relative movement between an aircraft and the surrounding air, turbulences emerge at these discontinuities. In acoustical terms, the turbulences result in a spectrally shaped noisy sound that seems to originate from where the front windshield and the body of the aircraft join.

Theoretically, the noise due to air-current should display dependencies on speed and on the direction of arrival of the air-stream relative to the aircraft. This is in turn influenced by the aircraft's attitude as well as by the direction of potentially present winds. Moreover, the sound depends on air density, which varies with altitude and air temperature.

Figure 2 shows the sound pressure level for two different flight states. In both states, the energy seems to be concentrated in low-frequency regions. Note that the plots corresponding to the different states are similar yet clearly distinct.

On flight decks of the aircraft type we are simulating, the air-current noise is the predominant acoustic event.

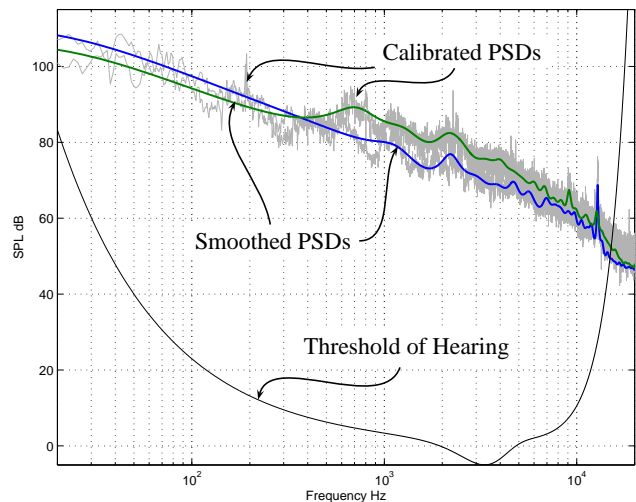


Figure 2: Calibrated Power Spectral Density (PSD) estimates (Welch's method of averaging modified periodograms) of the true sound pressure level for two different flight states. The smoothed spectra are computed by means of linear prediction analysis. For reference, the threshold of hearing is also plotted. The distinct peaks in the PSD estimates are discussed in Sections 2.1.2 and 2.1.6.

#### 2.1.2. Engine Noise

Some noise evolves from the moving parts (i. e. rotor blades) present inside the engines of an aircraft. A specific rotor produces a harmonic sound with a fundamental frequency determined by its rotary speed. Note that in contrast to the noisy sound due to the air-current, the sound from the engines has distinct sinusoidal components.

Furthermore, the engine sound is affected by air-speed and to some extent by the attitude of the aircraft relative to its trajectory. As for the acoustically relevant properties of the air, the same considerations apply as in the previous subsection.

When the engines are used as auxiliary brakes after landing, i. e., when they are in reverse thrust mode, the thrust generating airflow is redirected contrary to the direction of the aircraft movement. The exhaust air of the engines thus flows toward the flight deck. This causes additional noise.

The engine sound propagates partly through the structure of the aircraft, partly through the air outside. Due to the reflection off the runway, this latter propagation path is more distinct when the aircraft is on ground.

In the aircraft we are modeling there are two jet-engines, each with two internal rotors. Thus, we deal with a total of four rotors not generally revolving at exactly the same frequency. This lack of permanent complete synchronism causes characteristic beatings in the sound (also known as fluctuation strength).

The engine noise is reflected in Figure 2 by some of the distinct peaks, namely the ones at  $f \approx 190$  Hz and multiples thereof.

#### 2.1.3. Sound of Rolling Aircraft Wheels

When the aircraft rolls on the runway during take-off and landing phases, a structure-borne sound is produced, the cause of which is the friction between the tires of the landing gears and the ground,

in combination with the unevenness of the runway on different scales (cf. Figure 3).

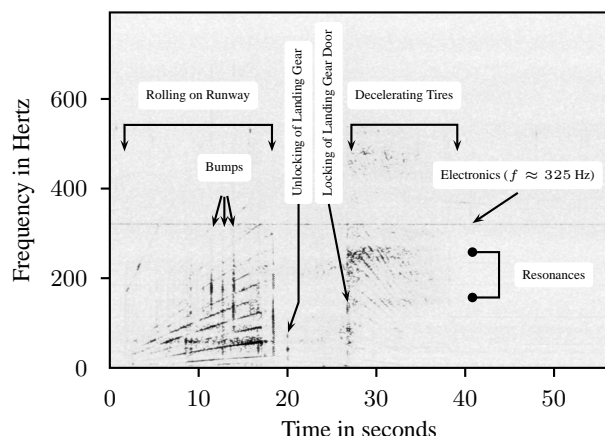


Figure 3: Spectrogram, around the time of take-off. See various subsections for details.

The quality of the sound of rolling aircraft wheels depends on the relationship between rotatory speed of the wheels and sliding speed of the aircraft, as well as on the contact pressure of the tires on the ground. Furthermore, the sound is influenced by the condition of the tires (profile), the brake system (e. g. ABS), and the suspension of the landing gear. Finally, water, snow, or ice present on the ground are determinant factors as well.

#### 2.1.4. Sound of APU

The auxiliary power unit (APU) is a source of energy located at the tail of the aircraft. During operation, it emits a loud noisy sound. Like the engine noise, the sound of the APU audible on the flight deck is partly structure-borne and partly airborne.

Before engine start, the APU can be switched on independently of other on-board systems. In contrast to most other acoustic events, its sound can therefore be recorded separately.

#### 2.1.5. Sound of Rain on Front Windshield

When an aircraft flies through a rain shower, a clearly audible, pattering sound is produced by the rain drops. The heavier the rain, the louder the sound. Furthermore, the quality of the sound depends on the cruising speed and the attitude of the aircraft. Clearly, the direction of arrival of the corresponding sound is more or less the whole front part of the aircraft, in particular the front windshield.

#### 2.1.6. Sound of Air Condition and Electronics

On the flight deck, there is a contribution to the sound field from electronic equipment such as inverters and avionics. The buzzing sound is responsible for the distinct peaks in the Power Spectral Density shown in Figure 2. See also the continuous line at the frequency  $f = 325$  Hz in Figure 3.

#### 2.1.7. Sound of Moving Parts

The moving parts on the outside of an aircraft—flaps, landing gears, and speed brakes—are positioned by servo motors. These

motors do produce sounds, which differ for various kinds of motors, e. g. hydraulic vs. electrical.

The sounds in this class can be considered temporary, i. e. they persist only as long as the corresponding part is in motion.

Our aircraft being large, basically the sounds of the moving parts located in the rear are almost inaudible. Therefore, we can neglect potential sounds from the flaps and the speed brakes.

## 2.2. Discrete Sounds

Unlike the continuous sounds described in the previous section, certain sounds occur in consequence of specific events. Approximately, these “discrete” sounds do not depend on any parameter, are reproduced when needed, and this reproduction cannot be stopped prematurely. The sounds in this class include:

### 2.2.1. Bumps on Runway

In addition to the sound of the rolling aircraft wheels described in Section 2.1.3, there is a squealing sound at the moment the aircraft touches down. We can also observe distinct bumps whenever the aircraft moves over the joints between concrete slabs on the runway. An example of such bumps can be seen in Figure 3.

### 2.2.2. Sound of Decelerating Tires

Consider the phase immediately after take-off, when the landing gears are already retracted. According to our information, the tires are then decelerated by an external force. A harmonic sound with falling fundamental frequency results. For the time being, we suppose that the rate of this decay is approximately constant. However, the wheel well of the landing gears causes pronounced resonances with time-invariant center frequencies and bandwidths. Both effects are clearly visible in the spectrogram in Figure 3.

### 2.2.3. Sounds of Locking Parts

There are often additional sounds when a part starts moving respectively when moving parts lock into their final positions (e. g. landing gears). Again, an example can be found in Figure 3.

## 3. DATA ACQUISITION

All necessary recordings were made in the cockpit of an aircraft during several scheduled flights. Consequently, as was inevitable, the recordings are deteriorated by the pilots’ interfering communication in the cockpit. Even so, since the quality and the quantity of the acquired data sufficed for our purposes, we did not have to use noise-reduction techniques. Instead, we relied on the “clean” segments as representative.

In order to obtain a most thorough acoustic picture of the cockpit, we used two recording techniques in parallel, as described in the rest of this section.

### 3.1. Binaural Recording Technique

The first recording technique comprised a high-quality binaural microphone system by Brüel & Kjær. The small and lightweight system allowed for extremely comfortable mounting and use. Furthermore, it features a full-range frequency response and can be calibrated, thus providing the basis for objective measurements. In



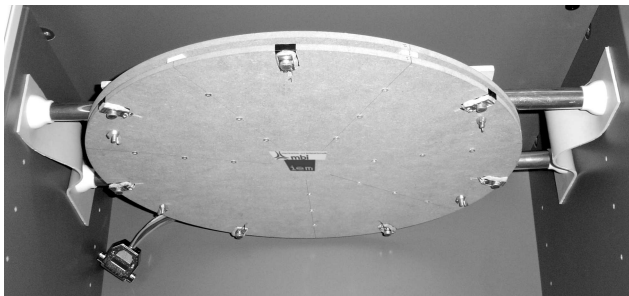


Figure 4: Microphone array mounted in a niche. The setup is similar to the one on the flight deck.

addition, the binaural signals are well suited for a subjective judgment about where in three-dimensional space the particular sounds come from.

On the other hand, the recordings made with the binaural microphone system are affected by whatever head movements the person wearing it—the pilot—has made during recording, which has to be considered during signal analysis.

### 3.2. Multi-Channel Recording Technique

As a second recording technique, we employ a planar array of microphones. Our prototype is shown in Figure 4. As a beamformer, such an array is the basis for spatial sampling. More specifically, it can be used to computationally assess the direction of arrival of the different sounds.

The cockpit geometry requires that the microphone array be positioned behind the pilot's seat in a niche<sup>1</sup> about half a meter in diameter. Consequently, we chose the diameter  $d$  of our prototype array as  $d = 0.45$  m. Based on our recording equipment, we can use a total of  $N = 7$  microphone channels.

The frequency range of the low-cost microphones themselves spans  $250 \text{ Hz} < f < 6 \text{ kHz}$  at  $-6 \text{ dB}$ . Such a limited range is justifiable because at frequencies below 380 Hz, beamforming in the sense of a delay-weight-sum (DWS) beamformer [4] is not feasible anyway given the array aperture, and because the power density spectrum of the sound pressure level decays toward higher frequencies (Figure 2).

## 4. SIGNAL ANALYSIS

In this section, we describe the methods and techniques applied to the analysis of the recorded data for each of the acoustic events discussed previously.

Additionally, we performed simple inspection of the waveform and of the spectral structure of a time-signal when needed. For example, in the case of the sound of decelerating tires, we determined both center frequency and bandwidth of the resonance directly from a spectrum estimate. Finally, mechanical data recorded simultaneously during the flight sometimes helps, e. g. the rotary speed of the engines is known and can be used in the analysis of the engine noise, and likewise for the sound of the aircraft rolling on the runway.

<sup>1</sup>Unfortunately, this causes unrecoverable alterations in the sound, e. g. standing waves.

### 4.1. Linear Prediction Analysis (Noise Due to Air-Current)

On an experimental basis, we discovered that the overall acoustic situation in the cockpit as recorded by the binaural system may appropriately be reproduced by some kind of filtered noise. Then, the dependence of the sound on certain parameters (Section 2) becomes manifest in different power spectral densities (Figure 2). In other words, the different flight states can be described by different filter curves.

In order to parameterize the different flight states, we use the technique of linear prediction analysis [5]. This technique models a signal  $x[n]$  by

$$x[n] = \sum_{j=1}^P a_j[n]x[n-j] + \sigma_v^2 \nu[n], \quad (1)$$

where  $\{a_j[n]\}_{j=1}^P$  are IIR filter coefficients at discrete time  $n$  and  $\nu[n]$  denotes samples drawn from a white-noise process with variance  $\sigma_v^2$ . As for the probability density function of  $\nu[n]$ , an analysis of prediction-error signals corresponding to several different flight situations showed that the random variable  $\nu[n]$  is best modeled as Gaussian distributed.

The representation according to Eq. (1) provides a very compact, small-set parameterization of the sound in a given flight state. For example, when modeling 31 poles, we get only  $31 \cdot 2 + 1 = 63$  parameters per flight state.

In order to provide a higher degree of realism, a few sinusoidal components should be added to the filtered noise. This is further discussed in Sections 4.2 and 5.3.

### 4.2. Peak Picking (Air Condition and Electronics)

The distinct peaks in the spectrum (Figure 2) can be attributed to the sounds from electronics and the air condition. Their exact location can be found by a standard peak-picking algorithm fed with the prediction-error signal [5] from the linear prediction analysis, with the contribution from the engine noise subtracted. Moreover, since the positions of the peaks stay relatively constant over time, temporal information can be incorporated in the peak detection.

In general, only ten to twenty peaks lie above the masking threshold.

### 4.3. Matching Pursuit Algorithm (Bumps)

The extraction of the bump sounds on the runway was accomplished with a matching pursuit (MP) algorithm [6, 7]. Compared to methods such as the Digital Fourier Transform (DFT), the MP algorithm provides a better trade-off between time resolution and frequency resolution [7]. It does not assume a harmonic structure in the signal. Instead, damped exponential sequences are used as the signal kernels [6]:

$$x[n] = \sum_{k=1}^Q \rho_k S_k \underbrace{r_k^{n-\Delta_k} e^{j\omega_k(n-\Delta_k)} u[n-\Delta_k]}_{\text{damped exponential}}. \quad (2)$$

Here,  $\omega_k$ ,  $r_k$ , and  $\rho_k$  are the frequency, the damping constant, and the amplitude of the damped exponential sequences, respectively.  $S_k$  is a scaling constant.  $\Delta_k$  denotes a time shift and  $u[n]$  the unit step sequence so that the  $k$ th exponential sequence is applied at time  $\Delta_k$ .

Extracting the bump sound by means of the MP algorithm with damped exponential sequences is adequate because as a kind of impulse response, the sound resembles an exponentially decaying sequence. Moreover, the MP algorithm is well suited for the analysis of transients since it uses asymmetric atoms [6].

Based on the MP analysis, a time signal of the “clean” bump sound can be computed and stored for later playback.

#### 4.4. Linear Filtering, Notch Filtering (Sound of Rain)

For the isolation of the sound of the pattering rain we used a combination of gentle high-pass filtering and notch filtering. More exactly, a comparison between the power spectral densities of segments with rain respectively without rain revealed that the energy of the sound of rain is concentrated in the region of higher frequencies. On the other hand, we tuned notch filters manually at the harmonics of the fundamental frequency corresponding to the instantaneous rotary speed so as to eliminate the engine noise.

In conclusion we may say that this simple procedure worked surprisingly well.

#### 4.5. Beamforming (Localizing Sounds)

We plan to use a combination of a constrained robust near-field beamformer with a conventional DWS beamformer in order to localize our sounds. For example, if for a known sound, most of the energy comes from a certain direction, the source of that sound will be mapped to this direction.

### 5. SIGNAL SYNTHESIS AND SPATIALIZATION

The synthesis techniques complementary to the analysis are presented in this section.

#### 5.1. Linear Prediction Synthesis

When synthesizing the noise due to air-current, we implement the IIR filtering in Eq. (1) by an all-pole lattice structure, parameterized by reflection coefficients [5].

The all-pole lattice filter implementation is more convenient in terms of stability in the bounded-input, bounded-output (BIBO) sense when transitions between two different flight states need to be computed. More specifically, imagine that at time  $n = 0$  our lattice filter is in state  $\{\kappa_j[0]\}_{j=1}^P$ . Then, a smooth transition to the set of new reflection coefficients  $\{\kappa_j^{\text{new}}\}_{j=1}^P$  corresponding to a new flight state can be obtained by the first-order recursion

$$\kappa_j[n+1] = (1 - \gamma) \kappa_j^{\text{new}} + \gamma \kappa_j[n], \quad 1 \leq j \leq P, \quad (3)$$

where the constant  $\gamma < 1$  determines the rate at which the new set of reflection coefficients  $\{\kappa_j^{\text{new}}\}_{j=1}^P$  is reached. If the absolute value of all coefficients from both sets is bounded by unity, the filter described by the coefficients  $\kappa_j[n]$  in Eq. (3) is always BIBO-stable, based on a property of reflection coefficients [5].

#### 5.2. Looping – Wavetable Readout

As soon as we have succeeded in isolating sounds such as bumps on the runway, the pattering sound of rain during the flight, or the sounds of moving parts, synthesis is easy for this class of sounds: we simply loop the corresponding takes with constant read-out

speed (bumps) or with variable read-out speed (rain), whereas the sounds of moving parts are reproduced playing back a wavetable just once.

In case the transition of a moving part stops for any reason, the wavetable playback is interrupted at once. Then, however, further sounds can emerge. For example, when the nose landing gear is blocked without reaching its final position, the landing gear door cannot close, and the resulting cavity produces an additional noisy sound (cf. Section 2.1.1).

#### 5.3. Additive Synthesis

As for the sounds produced by electronics and air condition, simple additive synthesis of few dozens sinusoidal signals with the correct amplitudes and frequencies is performed.

#### 5.4. FM-Synthesis

The harmonic spectra of sounds like engine noise, the sound of the rolling aircraft wheels, and the sound of decelerating tires can easily be synthesized using frequency modulation (FM), with both modulation frequency and carrier frequency tuned to match the rotary speed of the corresponding moving part. The modulation index is adjusted depending on the bandwidth of the desired spectrum. Subsequent resonance filtering might be necessary.

#### 5.5. Peak Filtering

The sound of decelerating tires is re-synthesized by filtering the signal from an FM-synthesizer with a second-order band-pass filter with adjustable center frequency and bandwidth.

#### 5.6. Spatialization

From a signal processing point of view, the generator modules incorporated in our sound system are all single output systems. Obviously, in order to provide the required degree of realism, these monophonic signals need to be spatialized properly. Depending on the type of a specific sound, different spatialization strategies are pursued.

##### 5.6.1. Positioning of Localizable Sounds

In principle, from the pilot's seat some of the sounds described in Section 2 should be localizable quite well, e. g. the sounds emanating from the APU, the engines, or the landing gears. Ideally, such sounds are best reproduced by a loudspeaker located at their exact physical position. But in the cockpit of the simulator we have a limited number of loudspeakers at predetermined positions not generally matching the desired ones (cf. Section 1.2).

However, we can do “virtual sound source positioning” using vector-base amplitude panning (VBAP), a technique proposed by Ville Pulkki that is computationally efficient for static sound sources [8]. With this technique, static virtual sources can be positioned anywhere within the spherical triangles spanned by loudspeaker triples.

##### 5.6.2. Providing Spatial Extent

In contrast to the situation described in the previous subsection, sounds such as air-current noise do not seem to originate from a

concisely defined point source. Instead, they apparently possess spatial extent.

Starting again from a single-channel signal, such spatial extent can be rendered by decorrelation techniques like those proposed in [9]. If, for example, two loudspeakers are fed by uncorrelated noise signals, the noise appears to fill the complete space between them [10].

In a similar fashion,  $k$  different decorrelated versions  $d^{(k)}[n]$  of the monophonic signal  $s[n]$  to be spatialized can be obtained from linear filtering:

$$d^{(k)}[n] = s[n] * a^{(k)}[n] \quad \forall k. \quad (4)$$

Here, the symbol  $*$  denotes discrete-time convolution and  $a^{(k)}[n]$  is the impulse-response of a filter with flat magnitude response but random phase response, i. e. an all-pass filter [9]. In our case,  $k$  equals the number of loudspeakers in the setup, i. e. we employ one filter per channel.

The procedure presented in the previous paragraph accomplishes decorrelation of a signal over the whole frequency range. In certain circumstances though, it might be better to provide a source with different degrees of spatial extent in various frequency bands. More precisely, consider splitting a monophonic source signal  $s[n]$  in a high-pass channel  $s_{HP}[n]$  and a low-pass channel  $s_{LP}[n]$ . Then—from a physical point of view—it might be clever not to decorrelate the low-pass channel  $s_{LP}[n]$ , since the spatial distribution of sound pressure level is approximately constant at low frequencies.

## 6. CONCLUSION AND FUTURE WORK

In this paper, we presented a sound system for a realistic, modular flight simulator. We showed that viable audio material can be obtained from a combination of binaural and multi-channel recordings, because then spatial information, too, can be assessed. We discussed both continuous and discrete acoustic events typically occurring on the flight deck of an aircraft and their dependencies on physical parameters. We proposed various techniques for the extraction of these acoustic events from real data. Above all, linear prediction analysis has proven useful for this purpose. Finally, issues of spatialization were addressed.

Up to now, in our project we have mainly dealt with recording respectively system-design issues. Additionally, we have assembled a rudimentary prototype (Figure 5), where in the presence of actual pilots we have proven the feasibility of our concepts. In future work we will focus on analysis and signal processing techniques, e. g. beamforming and blind deconvolution [5]. Moreover, the matching pursuit algorithm could be a helpful tool not only for the extraction of bumps, but for other sounds as well.

## 7. ACKNOWLEDGEMENT

The authors wish to thank Austrian Airlines for their support as well as the Styrian Government for funding this project as part of a research grant (GZ: A3-10 F 59-04/1).

We thank Robert Höldrich (Institute of Electronic Music and Acoustics, Graz) for his valuable contributions to this project.

## 8. REFERENCES

- [1] “Aeroplane flight simulators (amendment 3),” <http://www.jaa.nl/section1/jarsec1.html>, 2005.
- [2] Siegfried Vössner, Reinhard Braunstingl, Helmuth Ploner-Bernard, and Alois Sontacchi, “A new framework for a sound system for realtime flight simulation,” submitted to DAFx'05, Madrid, Spain, September 2005.
- [3] Pure Data web site, “<http://pd.iem.at>,” March 2005.
- [4] Michael Brandstein and Darren Ward, Eds., *Microphone Arrays*, Springer, 2001.
- [5] Simon Haykin, *Adaptive Filter Theory*, Prentice Hall, 4th edition, 2002.
- [6] Michael Goodwin, “Matching pursuit with damped sinusoids,” in *Proc. ICASSP '97*, Munich, Germany, April 1997, vol. 3, pp. 2037–2040.
- [7] Garry Kling and Curtis Roads, “Audio analysis, visualization, and transformation with the matching pursuit algorithm,” *Proceedings of the 7th International Conference on Digital Audio Effects (DAFx'04)*, pp. 33–37, October 2004.
- [8] Ville Pulkki, “Virtual sound source positioning using vector base amplitude panning,” *J. Audio Eng. Soc.*, vol. 45, no. 6, pp. 456–466, June 1997.
- [9] Guillaume Potard and Ian Burnett, “Decorrelation techniques for the rendering of apparent sound source width in 3d audio displays,” in *Proceedings of the 7th International Conference on Digital Audio Effects (DAFx'04)*, Naples, Italy, October 2004, DAFx, pp. 280–284.
- [10] Gary S. Kendall, “The decorrelation of audio signals and its impact on spatial imagery,” *Computer Music Journal*, vol. 19, no. 4, pp. 71–87, 1995.

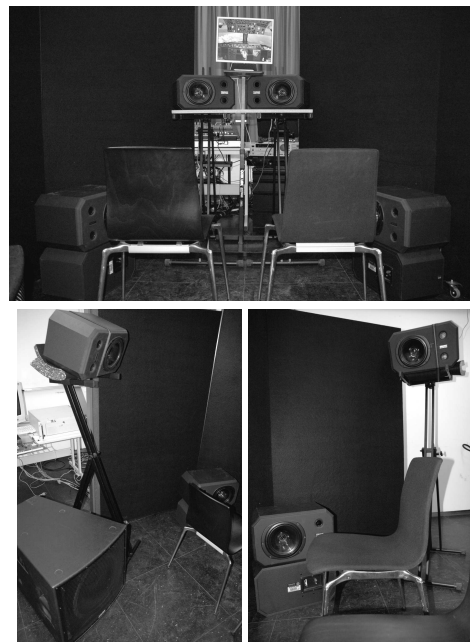


Figure 5: Prototype of sound system

Projekt

# **Flugsimulator – Dokumentation der Stand-Alone-Version des Simulators**

Helmuth Ploner-Bernard

---

Institut für Elektronische Musik und Akustik  
Universität für Musik und darstellende Kunst Graz  
Vorstand: o. Univ.-Prof. Dr. DI Mag. Robert Höldrich

Projektleiter: Univ. Ass. DI Dr. Alois Sontacchi

Graz, 27. September 2005

# 1 Allgemeines

Die vorliegende Dokumentation beschreibt die Stand-Alone-Version des Simulators und bezieht sich auf den Patch `simulator.pd` im Verzeichnis `fokker/simulator/` in der Version vom 27. September 2004.

Die Steuerung des Simulators erfolgt wahlweise über die Steuerelemente im Hauptpatch beziehungsweise über das USB-/MIDI-Interface „Behringer BCF 2000“. Änderungen der einen Steuerung ziehen Änderungen der jeweils anderen Steuerung mit sich.

## 2 Hardware

Der Aufbau des Prototyps umfasst folgende Elemente:

- 1 x Midisteuerung BCF2000 (Behringer)
- 6 x aktive Lautsprecher (Tannoy)
- 1 x passiven Subwoofer (EAW)
- 2 x passive Körperschallgeber
- 1 x aktive Frequenzweiche
- 1 x Mischkonsole O1V (Yamaha) mit ADAT-Einschub
- 1 x ADAT-Wandler ADI-8 PRO (RME)
- 1 x Verstärker Thomann S-75 (für Körperschallgeber)
- 1 x Verstärker Thomann S-150 (für Subwoofer)
- Stative, Verkabelung, Sitzgelegenheiten

## 3 Aufbau

Je zwei Lautsprecher vorne, zwei Lautsprecher an den Seiten, zwei Lautsprecher schräg hinter den Sitzgelegenheiten, ein Subwoofer an beliebiger Stelle und je ein Körperschallgeber unter den Sitzgelegenheiten. Midisteuerung an Position der Mittelkonsole.

## 4 Start-Up

Auf dem Rechner mit der unter 2 beschriebenen Hardware und unter Debian-Linux ist der Simulator wie folgt zu starten:

1. Alle Geräte einschalten und hochfahren lassen.
2. Am Mischpult alle Fader stumm schalten
3. **hdspmixer** ausführen und die gewünschten Einstellungen vornehmen
4. Am Behringer den USB-Modus aktivieren.
5. Die Behringer-Steuerung routen.
  - (a) Hierzu **qjackctl** ausführen
  - (b) Im Punkt „Connect“ unter dem Reiter „MIDI“ folgende Verbindungen stecken:  
72:Virtual Raw MIDI 1-0 0:VirMIDI 1-0 ↔ 80:BCF2000 0:BCF2000 MIDI 1  
80:BCF2000 0:BCF2000 MIDI 1 ↔ 72:Virtual Raw MIDI 1-0 0:VirMIDI 1-0
  - (c) Das Programm beenden
6. In das Verzeichnis **fokker/simulator** wechseln
7. PD und den Patch starten mit (abschließendes Et-Zeichen beachten) **pd simulator.pd &**
8. Warten, bis der Patch geladen und initialisiert ist
9. Über grünen Toggle-Button den DSP einschalten
10. **Das Windgeräusch über den Toggle-Button WIND aktivieren**
11. **Über die entsprechenden Toggle-Buttons die Autopiloten 1-4 aktivieren**
12. Langsam die Lautstärke im Patch und am Mischpult erhöhen

## 5 Funktionalität und Bedienung

### 5.1 Strömungsgeräusch

Das Stömungsgeräusch ist von folgenden vier Parametern abhängig:

- Position der Höhenflosse (Stabilizer Position) in Grad
- Fluggeschwindigkeit (True Airspeed) in Knoten
- Anströmwinkel (Angle of Attack) in Grad
- Flughöhe (Pressure Altitude) in Fuß

Die Analyse basiert auf den Daten des B & K-Binauralmessmikrophons vom 8. August 2005, Flüge Wien – München und retour<sup>1</sup>. Die Analyse durch Lineare Prädiktion wurde mit 62 Koeffizienten auf Segmenten durchgeführt, die händisch auf Freiheit von (allzu lauten) Störgeräuschen überprüft wurden.

Die Synthese des Strömungsgeräusches basiert auf o. g. Analysedaten und wurde mittels multipler Regression mit linearen und rein quadratischen Gliedern in der Domäne der Reflexionskoeffizienten<sup>2</sup> durchgeführt.

### 5.1.1 Manuelle Steuerung versus Autopilot

Über die entsprechenden Regler am Behringer lassen sich die vier Parameter in gewissen Bereichen einstellen. Dabei wurden die Begrenzungen des Wertebereiches anhand der Analysedaten gewählt, sodass der Bereich Extrapolation nach Möglichkeit gemieden wird.<sup>3</sup>

Um das Strömungsgeräusch händisch steuern zu können, sind die entsprechenden Autopiloten am Behringer oder im Patch zu deaktivieren.

Alternativ zur händischen Steuerung bietet der Patch die Möglichkeit, durch einen wirklich durchflogenen Satz von Parametern zu iterieren. Dabei wird nach Ablauf eines Zyklus die Meldung „autopilot: bang“ ausgegeben. Standardmäßig wird der Flug von Wien nach Hamburg abgespielt.

## 5.2 Regen und Hagel

Isoliert wurde das Regengeräusch aus Aufnahmen der Cessna Citation durch lineare Filterung und Notch-Filterung.

Das Hagelgeräusch wurde synthetisch aus der Sammlung von Audioeffekten zusammengestellt.

---

<sup>1</sup>Beim Rückflug München – Wien ist für die Phasen Start und Landung auf den Originalaufnahmen der Pegel abgesenkt worden, um eine Übersteuerung zu vermeiden. Diese Absenkung des Pegels ist vor der Analyse natürlich ausgeglichen worden.

<sup>2</sup>Nicht der Querschnittsflächen!

<sup>3</sup>Für künftige Versionen des Simulators sollen erweiterte Flugdaten diese Einschränkung mindern.

Regen und Hagel sind über den Toggle-Button **RAIN** freizuschalten. Der Regler am Behringer stellt folgendes ein:

- 0 % bis 50 %: kein Regen bis Regen mit voller Intensität
- 50 % bis 100 %: zum Regen mit voller Intensität kommt Hagel mit steigender Intensität hinzu

### 5.3 (Explosive) Decompression

Hierbei handelt es sich um den Fehlerfall, dass der Druck in Cockpit/Kabine nicht aufrecht erhalten werden kann. Je nach Druckunterschied zwischen Innen und Außen erhöht sich das Windgeräusch und das Geräusch der Triebwerke um bis zu 20 dB. Dies lässt sich über den Regler am Behringer einstellen.

Das Geräusch kann von einer Explosion begleitet sein (Explosive Decompression). Diese Explosion wird abgespielt, wenn bei aktiviertem Toggle-Button **EXPLOSIVE\_DECOMPRESSION** bzw. bei aktiviertem Button am Behringer der Regler von 0 % startend erhöht wird.

### 5.4 Triebwerksgeräusch

Das Triebwerksgeräusch wird drehzahlabhängig (Einstellung am Regler) über Frequenzmodulation mit Resonanzfilterung synthetisiert. Dafür muss das Triebwerksgeräusch über den Toggle-Button **ENGINE** freigeschaltet sein.

### 5.5 Windshield Wiper

Die Scheibenwischanlage arbeitet zweistufig (LOW/HIGH), wobei beim Ein- und Ausschalten je ein zusätzliches Geräusch hörbar ist.

Die Audiodaten zur Scheibenwischanlage wurden mit MD aufgenommen. Sie werden geeignet aneinandergereiht geloopt ohne Variation abgespielt.

Die Scheibenwischanlage ist über den Drehregler am Behringer (voller Wertebereich in drei Zonen unterteilt) bzw. über die Radiobuttons im Patch (AUS - LOW - HIGH) zu steuern.

### 5.6 Bugfahrwerk

Das Modell des Bugfahrwerks kennt die Zustände „eingefahren“, „beim Ausfahren“ und „voll ausgefahren“. Wird das Fahrwerk über den Regler am Behringer



ausgefahren, rastet es nach einer gewissen Zeit automatisch im Zustand „voll ausgefahren“ ein. Wird es über die Radio-Buttons im Patch gesteuert, kann es beliebig lange im Zustand „beim Ausfahren“ bleiben.

Beim Entriegeln bzw. beim Einrasten sind entsprechende Geräusche hörbar. Diese Geräusche wurden mittels „Noise reduction“ im CoolEdit aus den erwähnten Originalaufnahmen isoliert.

Siehe auch 5.15.

## **5.7 Donnergrollen**

Das Geräusch von Donnergrollen wird über einen Button am Behringer oder den Toggle-Button THUNDER im Patch abgespielt.

Das Geräusch entstammt der Audiodatensammlung Hollywood-Effects.

## **5.8 Crash**

Das Geräusch eines Crashes wird über einen Button am Behringer oder den Toggle-Button CRASH im Patch abgespielt.

Das Geräusch ist eine Collage aus Samples der Audiodatensammlung digiffects.

## **5.9 Airconditioning**

## **5.10 Vogelschlag**

Das Geräusch des Aufpralls von Vögeln gegen das Flugzeug wurde in zwei Varianten realisiert:

1. Vogelschlag gegen Windschutzscheibe, helleres Geräusch, nach vorne spatialisiert
2. Vogelschlag gegen Radom (Radar-Dome, die Nase des Flugzeugs), dumpferes Geräusch, nach unten spatialisiert

Ausgelöst wird es wiederum über Buttons am Behringer oder über die Toggle-Buttons BIRDSTRIKE\_WINDSHIELD bzw. BIRDSTRIKE\_RADOM.

Das Geräusch ist eine Collage aus Samples der Audiodatensammlung digiffects.

## **5.11 Structural Damage**

Gemeint ist, dass sich im Flug ein Teil der Struktur des Flugzeugs löst (z. B. ein Teil des Flügels). Auslösen wiederum über Behringer bzw. über den Toggle-Button STRUCTURAL\_DAMAGE.

Das Geräusch ist eine Collage aus Samples der Audiodatensammlung digiffects.

## 5.12 Schleifen auf Landebahn

Ist das Fahrwerk beim Aufsetzen auf die Landebahn nicht ausgefahren, wird das Flugzeug auf der Landebahn schleifend zum Stillstand kommen (siehe 5.18). Während dieser Zeit wird ein entsprechendes Geräusch abgespielt.

Das Geräusch ist eine Collage (inklusive lineare Faltung) aus Samples der Audiodatensammlung digiffects.

## 5.13 Abbremsen der Reifen

Sind die Fahrwerke eingefahren und die Türen der Fahrwerksschächte geschlossen, werden die Reifen abgebremst. Das Geräusch wurde durch Filterung aus den Originalaufnahmen gewonnen und wird beim Einfahren des Fahrwerks automatisch abgespielt.<sup>4</sup> Es kann auch über den Toggle-Button `TIRE_BRAKES` getriggert werden.

## 5.14 Landebahnlichter

Beim Start ist es erlaubt und üblich, über in der Landebahn versenkte Lichter zu fahren. Das entsprechende Geräusch wurde aus Originalaufnahmen gewonnen. Es wird in entsprechenden Intervallen abgespielt, wenn sich bei aktiviertem Toggle-Button `BUMP` die Rollgeschwindigkeit erhöht (siehe 5.17). Die Intervalle wurden aus den Aufnahmen abgeschätzt. Es ist nicht vorgesehen, das Geräusch einzeln zu triggern.

## 5.15 Aufsetzen des Bugfahrwerks

Das Geräusch beim Aufsetzen des Bugfahrwerks wurde aus den Originalaufnahmen mittels Noise reduction im CoolEdit isoliert. Es wird bei der automatischen Landung (siehe 5.18) abgespielt, kann aber auch über den Toggle-Button `TOUCH_DOWN_NOSE` getriggert werden.

## 5.16 Fehlerfall eines Triebwerks

Es wird der Fehlerfall „engine stall“ simuliert. Triggerung über Midisteuerung oder über Toggle-Button `ENGINE_STALL`.

Das Geräusch ist eine Collage aus Samples der Audiodatensammlung digiffects.

---

<sup>4</sup>Dies ist natürlich unrealistisch, falls das Fahrwerk im Flug ausgefahren und gleich darauf wieder eingefahren wird.

## 5.17 Start

Um das Fehlen einer „intelligenten“ Steuerung auszugleichen, wurde zu Demonstrationszwecken die Möglichkeit geschaffen, automatisch zu starten bzw. automatisch zu landen. Diese Features wählen sinnvolle Werte für alle Parameter aus.

Ein automatischer Start wird wie folgt durchgeführt:

1. Im Patch den Toggle-Button **AUTOPILOT** deaktivieren
2. Im Patch den türkisfarbenen Button **getReadyForTakeOff** drücken. Dies setzt die Maschine auf die Landebahn und schaltet die einzelnen Autopiloten aus. Alternativ kann man im Subpatch `pd logics` → `pd takeOff` den Toggle-Button unter dem Objekt `r $0-brakeSet` deaktivieren.
3. Auf der Konsole die Parkbremse lösen lassen. Das Flugzeug setzt sich in Bewegung (Rollgeräusch!), hebt ab und begibt sich schließlich in den Steigflug. Nach Belieben kann man nun das Fahrwerk einfahren.

## 5.18 Landung

Siehe besonders 5.17.

Eine automatische Landung wird wie folgt durchgeführt:

1. Im Patch den Toggle-Button **AUTOPILOT** deaktivieren
2. Im Patch den roten Button **getReadyForLanding** drücken. Dies setzt die Maschine auf den Landebahnkurs. und schaltet die einzelnen Autopiloten aus
3. Im Patch den dunkelgrünen Button **doTheLanding** drücken. Dies veranlasst den Landeanflug und schließlich die Landung mit dem Ausrollen bis zum Stillstand

Wird das Fahrwerk nicht ausgefahren (siehe 5.6), kommt es zum Schleifen auf der Landebahn (siehe 5.12).

## 6 Verzeichnisstruktur der PD-Software

Alle zum Simulator gehörigen Dateien sind unterhalb des Verzeichnisses `fokker/simulator` abgelegt.

Es folgt eine Kurzbeschreibung der Unterverzeichnisse:

`init_files` Enthält Initialisierungsdateien und Analysefiles (für das Interpolationsmodul)

`lib` Enthält alle Bibliotheken, die vom Simulator benötigt werden

`pictures` Enthält das Bild, das die Skizze des Flugzeugs zeigt

`wav_files` Enthält die Allpass-Impulsantworten sowie alle Sounds, die geloopt oder einmalig abgespielt werden

Wichtig ist es, alle oben genannten Verzeichnisse in den PD-Pfad (Kommandozeile `-path`) eingebunden zu haben.

## 7 Grobstruktur des Patches

Das Hauptpatch enthält alle wichtigen Steuerelemente. Es sind dies im Einzelnen:

- Toggle-Buttons zum Ein-/Ausschalten der einzelnen Lautsprecher (`frL` = frontLeft, `frR` = frontRight, `flL` = floorLeft, `flR` = floorRight, `reL` = rearLeft, `reR` = rearRight, `sub` = subwoofer, `bass` = bassPumps). **Diese Reihenfolge legt übrigens die Nummerierung der einzelnen Kanäle fest!** Also von links nach rechts, von vorne nach hinten, von hoch nach tief
- Toggle-Buttons für den Autopiloten (`AUTOPILOT`, 1, 2, 3, 4), siehe die Beschreibung der Geräusche in diesem Dokument (5)
- `pd initialize` Initialisiert den Simulator mit Informationen aus den Dateien unter `init_files`. Im Subpatch `pd initTables` werden alle Sounds von der Festplatte geladen
- `pd logics` Enthält die Steuerlogik für automatischen Start und automatische Landung
- `pd input` Enthält die Eingänge für die Steuerung von den Fadern mit entsprechender Skalierung laut Initialisierungsdatei, sowie das Auslesen der Eingangswerte aus einer Datei (im Falle des Autopiloten)
- `pd generators` Für jedes Geräusch ist ein entsprechendes Generatormodul vorhanden
- `pd spatialization` Empfängt die Signale aus den Generatormodulen und sorgt für eine geeignete Spalisierung auf die einzelnen Output-Busse

- `pd output` Lautstärkeregelung für die Output-Busse mit der Möglichkeit zur Entzerrung, enthält das `dac~`-Objekt. Master-Lautstärkeregler

Projekt

# **Flugsimulator – README-Datei zur Verzeichnisstruktur und zur Verwendung des Simulators**

Helmuth Ploner-Bernard

---

Institut für Elektronische Musik und Akustik  
Universität für Musik und darstellende Kunst Graz  
Vorstand: o. Univ.-Prof. Dr. DI Mag. Robert Höldrich

Projektleiter: Univ. Ass. DI Dr. Alois Sontacchi

Graz, 30. September 2005

# 1 Die Applikation soundmodule

Die Applikation `soundmodule` stellt die Schnittstelle zwischen den übrigen Rechnern im Simulatorverbund und dem Programm zur Signalverarbeitung in Echtzeit PD dar. Mit anderen Worten ermöglicht es die Steuerung des Simulator-Patches über ein gemeinsames Protokoll. Die Applikation ist nur zur Verwendung unter Linux bestimmt.

## 1.1 Quellen

Der originale Quelltext zur Applikation findet sich im Verzeichnis `fokker/application/soundmodule`.

Weiters stützt sich die Applikation auf zwei Bibliotheken:

1. `fokker/interpol` zur Parameterinterpolation in Echtzeit
2. `fokker/lpc/lattice` zur Filterung mit einer Lattice-Struktur

*Bemerkung:* Beide Bibliotheken lassen sich nur auf Linux kompilieren. Die vorhandenen `Makefiles`<sup>1</sup> sind aber leicht auf Windows-Kompiler umzuschreiben. Damit ließe sich die Stand-Alone-Version des Simulators, die diese beiden Bibliotheken ebenfalls benötigt, auch auf Windows betreiben.

Außerdem sind auch der Simulator-Patch selbst `fokker/simulator` und die entsprechenden WAV- und Konfigurationsfiles `fokker/simulator/wav_files` bzw. `fokker/simulator/init_files` integrativer Bestandteil der Applikation.

All diese Quellen sind im Source-RPM `fokker/soundmodule-0.1-src.rpm` abgepackt.

## 1.2 Vorbereitung zur Installation

Um die Applikation `soundmodule` zum Laufen zu bringen, muss auf dem Zielrechner das Programm Pure-Data (PD) installiert sein. Dieses Programm liegt als `fokker/pd-0.38-4.src.tar.gz` vor. Installationsanweisungen sind im TAR-Ball enthalten.

Es werden außerdem zwei Bibliotheken benötigt:

1. `fokker/zexy-unix-1.3.tgz`
2. `fokker/iemlib.R1.15.lin.tgz`

---

<sup>1</sup>`fokker/interpol/src/Makefile` und `fokker/lpc/lattice/src/Makefile`

Auch diese Pakete enthalten entsprechende Installationsanweisungen.

Nach der Installation von PD und den beiden Bibliotheken muss man dafür sorgen, dass die sog. Abstractions der `fokker/iemlib` für PD verfügbar sind. Dazu muss man PD starten und im Menu `File -> Path...` das Verzeichnis `/usr/local/lib/pd/iemabs` eintragen.

Außerdem muss PD so konfiguriert sein, dass es mindestens 8 Audiokanäle bereitstellt. Als Samplingrate sind 44,1 kHz zu wählen.

Weiters müssen folgende Verzeichnisse im PD-Pfad liegen:

- `/usr/local/soundmodule/simulator/init_files`
- `/usr/local/soundmodule/simulator/lib`
- `/usr/local/soundmodule/simulator/wav_files`

Es müssen folgende Libraries geladen werden:

- `iemlib1`
- `iemlib2`
- `iem_matrix`
- `zexy`
- `lattice`
- `interpol`

Der Aufruf der Applikation `soundmodule` muss mit Root-Rechten erfolgen!

### 1.3 TODO Liste

Die Applikation konnte nicht für alle Fälle getestet werden, da es zum Einen nicht möglich war, sich auf eine vollständige Liste von SysIds zu einigen, und zum anderen sind nicht alle Zustände des Simulations-Kernels dokumentiert. Auch bei den Datentypen gibt es keine endgültige Version.

Daraus ergeben sich im Quelltext Änderungen an folgenden Stellen:

- `fokker/application/soundmodule/src/mathdata.h` Umrechnung der Ganzzahlwerte auf floats
- `fokker/application/soundmodule/src/etc/commconf.xml` Das Konfigurationsfile im Abschnitt `<DataDefinition>`



- `fokker/application/soundmodule/src/IEM_Configurator.cpp` Hier wird die Konfigurations-Datei `commconf.xml` gezielt nach SysIds durchsucht `IEM_Configurator::initSysIds()`
- `fokker/application/soundmodule/src/IEM_Configurator.h` Dokumentation anpassen
- `fokker/application/soundmodule/src/IEM_MessageListener.cpp` Empfangen der einzelnen Messages und Umwandlung auf eine für PD verständliche Form, je nach SysId
- `fokker/application/soundmodule/src/IEM_Kernel.cpp` Das Errorhandling wurde nicht besprochen

## 2 Verzeichnisstruktur

- `iemlib_R1.15_lin.tgz` PD-Bibliothek
- `zexy-unix-1.3.tgz` PD-Bibliothek
- `pd-0.38-4.src.tar.gz` Pure-Data
- `analyse` Alles zur Analyse der Windgeräusche
  - `flugdaten_parameter` Beschreibung der Flugdaten und Flugdaten selbst
  - `matlab-work` Sämtliche MATLAB-Analysefiles
- `application` Applikation `soundmodule`, siehe 1.1
- `array_measurement` Messung der fürs Array verwendeten AKG-Mikrophone, mit und ohne Phantomspeisespannung, PD-Patches und octave-Analyse-files (sweep-Messung)
- `cessna` Beschreibung des Testflugs mit der Cessna Citation
- `correspondence` Besprechungsnotizen, Briefverkehr
- `c_utils` Zu Testzwecken verwendete C-Programme
- `da40` Probeflug mit Diamond DA-40, Beschreibung der Aufnahmen
- `documentation`

- `a_priori.pdf` Untersuchungen, welche Wahrscheinlichkeitsdichtefunktion der LPC-Rauschgenerator haben soll
- `dafx` Veröffentlichung auf DAFx'05
- `design_document` für Firma AXIS
- `simulator_stand_alone` Dokumentation der Stand-Alone-Version des Simulators
- `sounds_overview` Untersuchungen über mögliche Geräusche im Cockpit und deren Ursachen
- `fokkerfotos` Bilder der Fokker am Boden (Danke an MBI)
- `holland` Montageanleitung (auch Step-by-Step) für Aufnahmeequipment, Checkliste für Montage, Checkliste für Equipment (ursprünglich hätte ein Probeflug in Holland stattfinden sollen)
- `interp1` PD-Bibliothek zur Parameterinterpolation in Echtzeit
- `lautsprecher` Hardware des Simulators: Lautsprecher und Verstärker, Verkabelung, Preise
- `literature` Wissenschaftliche Papers über diverse Themen, AXIS-Coding-Standard, Flughandbuch der Diamond DA-40, öffentlich zugängliche Spezifikationen über Level-D-Flugsimulatoren, Beschreibung von MS-Projekt-Fokker100, Vorschläge für Design-Dokument
- `lpc` Programme zur LPC-Analyse und Synthese
  - `analysis` C-Programm zur Bildung der Autokorrelationsfunktion und zur Berechnung eines LPC-Filters mittels Levinson-Durbin-Rekursion
  - `lattice` PD-Bibliothek zur Filterung mittels Lattice-Struktur, siehe 1.1
  - `levinson` PD-Bibliothek zur Berechnung der Levinson-Durbin-Rekursion
- `pd_utils` Zu Testzwecken verwendete PD-Patches
- `rpm` Dateien zur Erzeugung des Source-RPMs der Applikation (1.1), besonders das Spec-File und ein Makefile, um alle Dateien auf einen Platz zu kopieren
- `sh_utils` Shell-Skripts für diverse Aufgaben

- `simulator` Stand-Alone-Version des Simulators sowie applikationsgesteuerte Version, siehe 1.1 und eigene Dokumentation im Verzeichnis `fokker/documentation/simulator_stand_alone`
- `tv_sounds` Sounds aus der Stand-Alone-Version des Simulators für einen TV-Beitrag im ORF-Wissenschaftsmagazin „modern times“ vom 7. Oktober 2005