Matthias Oberhauser, MSc

The Virtual Reality Flight Simulator

Development, Evaluation, and Demonstration of a Tool for Human Factors Engineering

Doctoral Thesis

to achieve the university degree of

Doktor der technischen Wissenschaften

submitted to

Graz University of Technology

Supervisor

Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Reinhard Braunstingl

Co-supervisors

Dr.-Ing. Daniel Dreyer
Dr. Ioana Koglbauer

Institute of Mechanics

Faculty of Mechanical Engineering and Economic Sciences

Graz, February 2018

This document is set in Palatino, compiled with pdfLaTeX2e and Biber.

The LaTeX template from Karl Voit is based on KOMA script and can be found online: https://github.com/novoid/LaTeX-KOMA-template

I would have written a shorter letter,
but I did not have the time.

BLAISE PASCAL

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present doctoral thesis.

02/22/2018
Date

Signature

Acknowledgements

I would first like to thank my thesis advisor **Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Reinhard Braunstingl** at the Institute of Mechanics at the Graz University of Technology for always having an open door. Thank you for your guidance and the opportunity to work at the flight simulation facilities. This work, in particular the evaluation section, would not have been possible without the help of **Dr. Ioana Koglbauer** and her patience in teaching me the principles of experiment design of human factors studies. I would also like to thank **Dr. Fernando Gómez Bravo** from the University of Huelva, who agreed to be the external examiner of this thesis.

At Airbus Group Innovations, I want to thank **Dr.-Ing. Daniel Dreyer**, my advisor and colleague, for four years of hard work, life lessons, and the most joyful work environment. I would also like to thank **Sébastien Mamessier**, who not only shared an office but also his excellent coding skills with me. I am most grateful that I had the opportunity to work with **Dr. Thomas Convard**, at the time a researcher at Airbus Group Innovations in France, who introduced me to the Robot Operating System. I would also like to thank my superiors, **Marc Velten** and **Dr. Volker Ziegler** for their work and support to make this thesis and the related projects possible. This research also greatly benefited from the help of students that worked in our lab: **Grégoire Baranger**, **Primož Kovačič**, **Jake** and **Max Hershfield**, **William Rakotomanga** and **Saskia Tomes**.

I would like to thank all pilots that participated in the trials that where part of this research who had to wear a head-mounted display, had electrodes attached to their bodies and had to go through endless interviews and paperwork.

Parts of this research were conducted in the scope of European projects with funding from the European Union. During one of these projects, I am grateful to have met **Dr. Daniel Bandow** who shared his knowledge on head-up displays with me.

Above all, I would like to thank my wife **Silke Oberhauser** and my son **Kuno** for providing me the necessary time and support to complete this thesis. My wife's proofreading was irreplaceable.

Abstract

In this research, a Virtual Reality Flight Simulator (VRFS) for Human Factors Engineering (HFE) is presented. This simulator aims at aiding the flight deck design process by providing a cost-effective and flexible environment for conducting user studies, in particular in the early phases of the design process. This thesis presents the design, implementation, experimental evaluation, and practical application of the simulator.

The system consists of a Head-mounted Display (HMD), a head and finger tracking system, and a flight simulation software. The user is able to interact directly with the virtual cockpit or with the help of additional hardware components. With the latter approach, a so-called mixed mock-up is created, which increases the usability of the system. All components are connected with the help of the Robot Operating System (ROS), a network middleware for robotic applications. Due to this modular architecture, single system Human-machine Interface (HMI) prototypes can be integrated into a holistic and operational cockpit environment with low effort. With the help of the integrated Human Factors (HF) methods such as eye tracking or posture analysis, these integrated prototypes can be evaluated in user studies.

In an experimental study, the fidelity of the VRFS is compared to a conventional hardware simulator. These assessments show a certain degradation in performance, workload, and situation awareness in the virtual environment. Yet, all pilots were able to use the system and complete the given task successfully.

In four user studies, which represent different types of human factors engineering studies in different phases of the development process, the practical application of the virtual environment is presented. The user studies successfully demonstrated the integration of different types of HMI components, the evaluation with professional pilots, and the feasibility of operational scenarios.

This thesis concludes with a discussion on the fidelity of the system and the user studies in the Virtual Reality Flight Simulator. Based on this knowledge, design recommendations for virtual cockpits and future user studies are provided and possible future research in various fields of application is discussed.

Zusammenfassung

In dieser Dissertation wird ein virtueller Flugsimulator, der Virtual Reality Flight Simulator (VRFS), präsentiert. Dieser Simulator zielt darauf ab, den Cockpitentwicklungsprozess für das Flugzeug mit einem kostengünstigen und flexiblen System, insbesondere in einer frühen Phase der Produktentwicklung, zu unterstützen. In dieser Arbeit werden das Design, die Implementierung, die experimentelle Evaluierung sowie die praktische Anwendung dieses Simulators präsentiert.

Das System besteht aus einem Head-mounted Display (HMD), einem Kopf- und Fingertrackingsystem und einer Flugsimulationssoftware. Der Nutzer kann mithilfe dieser Komponenten in das virtuelle Cockpit eintauchen und mit virtuellen sowie mit physischen Kontrollelementen interagieren. Bei der Interaktion mit physischen Kontrollelementen wird ein sogenanntes „Mixed Mock-Up“ erzeugt, wodurch die Nutzbarkeit des Systems erhöht wird. Alle Komponenten sind durch das Robot Operating System (ROS), eine Netzwerk-Middleware für Robotikanwendungen, verbunden. Durch diese modulare Architektur können neue, prototypische Mensch-Maschine-Schnittstellen mit wenig Aufwand in eine ganzheitliche, operationelle Cockpitumgebung integriert werden. Durch die integrierten Human Factors Methoden, wie Eye Tracking (also die Erfassung von Blickbewegungen) oder Bewegungsanalysen, können diese integrierten Prototypen in Nutzerstudien evaluiert werden.

In einer Vergleichsstudie mit einem konventionellen Flugsimulator wird die Realitätsnähe der virtuellen Umgebung experimentell evaluiert. Diese Untersuchungen zeigen, dass die Leistung und das Situationsbewusstsein im virtuellen Simulator abnehmen während sich sowohl die kognitive als auch körperliche Belastung auf den Piloten erhöht. Trotzdem waren alle Piloten in der Lage, das System zu nutzen und die gestellte Aufgabe erfolgreich abzuschließen.

In vier Nutzerstudien wird die praktische Anwendung des VRFS in verschiedenen Phasen des Cockpitentwicklungsprozesses präsentiert. Diese Studien zeigen die Integration von Prototypen, die Evaluierung mit professionellen Piloten und Möglichkeiten, realistische operationelle Szenarien zu erstellen.

Diese Arbeit schließt mit einer Diskussion zur Realitätsnähe und den aus den Nutzerstudien gewonnenen Erkenntnissen. Basierend auf diesem Wissen werden Empfehlungen für das Design von Cockpits mithilfe des VRFS sowie von zukünftigen Nutzerstudien erarbeitet. Ein Ausblick auf mögliche neue Anwendungsfelder und weitere Forschungsaspekte wird ebenfalls am Ende dieser Arbeit präsentiert.

The following publications are part of the work presented in this thesis:

Oberhauser, Matthias, Daniel Dreyer, Reinhard Braunstingl, and Ioana Koglbauer (2018). "What's Real About Virtual Reality Flight Simulation? Comparing the Fidelity of a Virtual Reality with a Conventional Flight Simulation Environment." In: *Aviation Psychology and Applied Human Factors* (In Print).

Oberhauser, Matthias and Daniel Dreyer (2017). "A virtual reality flight simulator for human factors engineering." In: *Cognition, Technology & Work* 19.2, pp. 263–277. ISSN: 1435-5566. DOI: 10.1007/s10111-017-0421-7.

Oberhauser, Matthias, Daniel Dreyer, Reinhard Braunstingl, and Ioana Koglbauer (2017a). "Pilots' Flight Performance in a Virtual Reality Flight Simulator." In: *Proceedings of the 32nd Conference of the European Association of Aviation Psychology*. Ed. by Michaela Schwarz and Julia Harfmann. Groningen, NL: European Association for Aviation Psychology, pp. 322–335. ISBN: 978-90-815253-5-0.

Oberhauser, Matthias, Daniel Dreyer, Reinhard Braunstingl, and Ioana Koglbauer (2017b). "Pilots' Interaction with Hardware Controls in a Virtual Reality Flight Simulator." In: *Proceedings of the 32nd Conference of the European Association of Aviation Psychology*. Ed. by Michaela Schwarz and Julia Harfmann. Groningen, NL: European Association for Aviation Psychology, pp. 565–575. ISBN: 978-90-815253-5-0.

Dreyer, Daniel and Matthias Oberhauser (2016a). "Beyond the Push-Button: A study on system management operations in the flight deck." In: *Proceedings of the International Conference on Human Computer Interaction in Aerospace '16*. Ed. by Guy A. Boy. New York, NY, USA.: ACM.

Oberhauser, Matthias, Daniel Dreyer, Thomas Convard, and Sebastien Mamessier (2016). "Rapid Integration and Evaluation of Functional HMI Components in a Virtual Reality Aircraft Cockpit." In: *Advances in Ergonomics in Design*. Ed. by Francisco Rebelo and Marcelo Soares. Cham, Switzerland: Springer International Publishing, pp. 17–24. ISBN: 978-3-319-41983-1. DOI: 10.1007/978-3-319-41983-1_2.

Mamessier, Sebastien, Daniel Dreyer, and Matthias Oberhauser (2015). "Semantically Integrated Human Factors Engineering." In: *Digital Human Modeling. Proceedings, Part II*. Ed. by G. Vincent Duffy. Cham, Switzerland: Springer International Publishing, pp. 284–294. ISBN: 978-3-319-21070-4. DOI: 10.1007/978-3-319-21070-4_29.

Oberhauser, Matthias, Daniel Dreyer, and Primož Kovačič (2015). "Rapid Prototyping in einem funktionalen, virtuellen Flugzeugcockpit." In: *12. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung*. Ed. by Jürgen Gausemeier, Michael Grafe, and Friedhelm Meyer auf der Heide. Paderborn, Germany, pp. 9–20. ISBN: 978-3-942647-61-8.

Oberhauser, Matthias, Daniel Dreyer, Sebastien Mamessier, Thomas Convard, Daniel Bandow, and Axel Hillebrand (2015). "Bridging the Gap Between Desktop Research and Full Flight Simulators for Human Factors Research." In: *Engineering Psychology and Cognitive Ergonomics*. Ed. by Don Harris. Vol. 9174. Lecture Notes in Computer Science. Cham, Switzerland: Springer International Publishing, pp. 460–471. ISBN: 978-3-319-20372-0. DOI: 10.1007/978-3-319-20373-7_44.

Dreyer, Daniel, Matthias Oberhauser, and Daniel Bandow (2014). “HUD Symbology evaluation in a Virtual Reality Flight Simulation.” In: *Proceedings of the International Conference on Human-Computer Interaction in Aerospace '14*. New York, NY, USA: ACM, pp. 9–14. ISBN: 978-1-4503-2560-8. DOI: 10.1145/2669592.2669652.

Hillebrand, Axel, Matthias Oberhauser, Daniel Dreyer, and Daniel Meister (2014). “Modulare Erweiterbarkeit von Flugsimulationen für Human-Factors-Experimente.” In: *3. Interdisziplinärer Workshop Kognitive Systeme: Mensch, Teams, Systeme und Automaten*.

Mamessier, Sebastien, Daniel Dreyer, and Matthias Oberhauser (2014). “Calibration of Online Situation Awareness Assessment Systems Using Virtual Reality.” In: *Digital Human Modeling. Applications in Health, Safety, Ergonomics and Risk Management: 5th International Conference, DHM 2014, Held as Part of HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014. Proceedings*. Ed. by Vincent G. Duffy. Cham: Springer International Publishing, pp. 124–135. ISBN: 978-3-319-07725-3. DOI: 10.1007/978-3-319-07725-3_12.

The following patent applications are based on work presented in this thesis:

Dreyer, Daniel, Matthias Oberhauser, Andreas Prücklmeier, and Sebastien Mamessier (2017). "Device and Method for Testing Function or Use of a Head Worn See Through Augmented Reality Device." European Patent 15198726.0 - 1562.

Dreyer, Daniel and Matthias Oberhauser (2016b). "Method and device for testing a device operated in an aircraft." European Patent EP3067874A1.

Dreyer, Daniel and Matthias Oberhauser (2016c). "Selection unit to select or control different states or functions of an aircraft system." European Patent EP3018571A1.

Dreyer, Daniel, Matthias Oberhauser, and Philipp Schmidt (2015). "Fire extinguishing system for an aircraft." European Patent EP20150161127.

Parts of the following chapter are submitted for publication:

Chapter 5: Parts are submitted to the Aviation Psychology and Applied Human Factors (APAHF) Journal.

Contents

List of Figures

List of Tables

Listings

List of Abbreviations

2D Two-dimensional

3D Three-dimensional

6DOF Six Degrees of Freedom

ACROSS Advanced Cockpit for Reduction of Stress and Workload

AI Artificial Intelligence

ALICIA All Condition Operations and Innovative Cockpit Infrastructure

ANOVA Analysis of Variance

AOI Area of Interest

API Application Programming Interface

APPR Approach

ART Advanced Realtime Tracking

ATC Air Traffic Control

BPM Beats per Minute

BSD Berkeley Software Distribution

CAD Computer-aided Design

CAVE CAVE Automatic Virtual Environment

CAVOK Ceiling and Visibility OK

COTS Commercial Off-the-shelf

CPU Central Processing Unit

CRM Crew Resource Management

CRT Cathode Ray Tube

CSV Comma-separated Values

DERP Design Eye Reference Point

DH Decision Height

ECG Electrocardiogram

ECAM Electronic Centralised Aircraft Monitor

EEAG External Expert Advisory Group

EFB Electronic Flight Bag

EVE Enhanced Virtual Environment

FBW Fly-by-wire

FCOM Flight Crew Operating Manual

FCU Flight Control Unit

FFS Full Flight Simulator

FMS Flight Management System

FOR Field of Regard

FOV Field of View

FP 7 Seventh Framework Programme

FPV Flight Path Vector

FTD Flight Training Device

GA General Aviation

GPS Global Positioning System

GPU Graphics Processing Unit

GUI Graphical User Interface

HCI Human–computer Interaction

HF Human Factors

HFE Human Factors Engineering

HMD Head-mounted Display

HMI Human-machine Interface

HR Heart Rate

HRV Heart Rate Variability

HTA Hierarchical Task Analysis

HUD Head-up Display

HUGS Head-up Guidance System

HWD Head-worn Device

IAF	Initial Approach Fix
IBI	Interbeat Interval
ICAO	International Civil Aviation Organization
IFR	Instrument Flight Rules
ILS	Instrument Landing System
INS	Inertial Navigation System
IR	Infrared
I-Vision	Immersive Semantics-based Virtual Environments for the Design and Validation of Human-centered Aircraft Cockpits
JSON	JavaScript Object Notation
LCD	Liquid Crystal Display
LCOS	Liquid Crystal on Silicon
LED	Light-emitting Diode
MFD	Multi-function Display
MSFS	Microsoft Flight Simulator
NASA	National Aeronautics and Space Administration
NASA-TLX	NASA Task Load Index
OHP	Overhead Panel
OLED	Organic Light-emitting Diode
P3D	Prepar3D
PBO	Pixel Buffer Object
PF	Pilot Flying
PFD	Primary Flight Display
PNF	Pilot Not Flying
PVHD	Peripheral Vision Horizon Display
QPAC	QualityPark AviationCenter
RFB	Remote Frame Buffer
RMS	Root Mean Square
RWY	Runway
ROS	Robot Operating System
RQT	ROS QT Framework
RVIZ	ROS Visualization Tool
SA	Situation Awareness
SART	Situation Awareness Rating Technique
SCFM	System/Component Failure/Malfunction
SD	System Display
SDK	Software Development Kit
SAGAT	Situation Awareness Global Assessment Technique
Soft-ICP	Soft Interface Control Panel
SSQ	Simulator Sickness Questionnaire
STL	STereoLithography
TC/IP	Transmission Control Protocol/Internet Protocol
TCAS	Traffic Collision Avoidance System
TCT	Task Completion Time
TF	ROS Transform Library
TFT	Thin-film Transistor
UDP	User Datagram Protocol
UML	Unified Modeling Language
USB	Universal Serial Bus
VFR	Visual Flight Rules
VNC	Virtual Network Computing
VR	Virtual Reality
VRFS	Virtual Reality Flight Simulator
XML	Extensible Markup Language
XP	X-Plane

1. Introduction

> You can use an eraser on the drafting table or a sledge hammer on the construction site. (Frank L. Wright)

This quote, attributed to American architect Frank L. Wright, describes the fundamental challenge of every product development process: In an early phase, changes to a product can be implemented with low effort; in later phases, the costs of changes increase significantly. This means that important design decisions with far-reaching consequences have to be made at a point, when only limited information on the final product and its users are available. This challenge also applies to the flight deck development process. As a consequence, it became common practice to include pilot feedback at early stages of the design process in order to create a human centered flight deck design that takes into account human performance, perception, and limitations, right from the beginning. In other words, a design that considers Human Factors (HF). Or, as W. F. Moroney and B. W. Moroney (2009) explain:

> The cost of implementing a change to a system tends to increase geometrically as the project moves from conceptual designs to completed development. Cost considerations alone may require a priori or ad hoc approaches, where a human factors evaluation process is carried out in a manner that allows the needed changes to be made when the cost impact is low. Ideally, evaluation of complex aviation systems would require human factors consultation throughout the conceptual (predesign), design, and implementation process. The involvement of a human factors practitioner during the process would guarantee consideration of the users' needs and insure an optimal degree of usability (p. 4-5).

Product life cycles in the civil aviation industry are fifty years and more (Shaw, 2007, p. 148). In addition, aircraft manufacturers like Airbus have a commonality strategy. This means that new aircraft have a similar flight deck design as legacy products, in order to reduce crew training costs (Vasigh, Taleghani, and Jenkins, 2012, p. 114). As a consequence of these economic factors, the development of a new flight deck from scratch rarely happens. However, technological advances cannot be ignored. Hence, the incorporation of novel technologies into current flight decks (i.e. retrofitting) and the continuous evolutionary enhancements of flight deck designs is a common and reoccurring task for flight deck and human factors engineers. Yet, the mere integration of technologies can lead to problems. Clamann and Kaber (2004) observed:

> More displays are added to cockpits without re-engineering the control array and are fit where there is still available space. Over time, this has resulted in a complex array of knobs, switches, and displays that does not necessarily integrate intuitively (p. 400).

Such evolved and cluttered designs will eventually lead to human factors issues and could contribute to incidents or accidents.

1.1. Motivation

Whether it is the development of a completely new cockpit, technology retrofitting, or an evolutionary enhancement, a holistic, user-centric view that takes into account human factors will always add value. In order to do so, it is necessary to demonstrate and evaluate proposed designs in all stages of the design process. Preferably, these demonstrations or evaluations take place in high fidelity flight simulators that create an experience close to the final product. Yet, the integration of new, single technologies or the design of completely new flight decks can be time and cost consuming when conventional simulators are used.

A simulator that, as Kaiser and Schroeder (2002) state, makes "...a greater use of the virtual to replace the physical" (p. 466), can be a more flexible alternative for the early stages of the flight deck design process. Following this statement, in this research, a Virtual Reality Flight Simulator (VRFS) for Human Factors Engineering (HFE) is developed. The system is supposed to aid the early stages of the flight deck design process by presenting novel Human-machine Interface (HMI) concepts in a holistic, operational but still flexible flight deck environment. This simulator will use immersive Virtual Reality (VR) technology combined with a state-of-the-art flight simulation environment. With the help of virtual reality, a user can immerse into a virtual, Three-dimensional (3D) space using a Head-mounted Display (HMD) or other devices. This technology has been well known for decades but only recently experienced rising interest from the consumer industry which now leads to rapid technological progress in this field.

1.2. Contribution

The development of the Virtual Reality Flight Simulator is the main contribution of this research. In addition, contributions to the fields of human factors engineering and virtual reality research are expected.

The main contribution, the development of the VRFS, will provide engineers and human factors practitioners with a flexible environment to integrate, demonstrate, and evaluate novel cockpits and HMI concepts. The user in the virtual cockpit will be able to use and fly the simulated aircraft in operational scenarios.

Another major contribution is the evaluation of usability and fidelity of this virtual environment in a comparative study, a task that has not been yet conducted according to the available literature. This will lead the way for further enhancements of this system and will help developers understand and overcome the challenges of these kind of simulators with regard to usability and fidelity.

The third contribution is providing easy-to-use methods in order to integrate new components into the simulator; this can either be single system prototypes or complete cockpits. With the focus on commonly used tools and frameworks, some of which are open source, the VRFS will be highly accessible for third party developers and researchers.

The fourth contribution expected will be the demonstration of the integrated human factors methods. With these methods, some used in an immersive dynamic virtual environment for the first time, the human behavior can be assessed. The provided methods enable human factors practitioners to conduct user studies and evaluate cockpit concepts or single HMI components. Thus, this integration contributes to future research in the field of virtual reality that is aiming at analyzing and quantifying the human behavior in dynamic virtual environments.

1.3. Structure of the Thesis

In this chapter, the motivation for the development of the Virtual Reality Flight Simulator and the expected contributions for the field of human factors engineering and virtual reality research have been presented. The further structure of the thesis is as follows:

In Chapter 2, Background, information on the the flight deck development process, human factors engineering in the flight deck, and virtual reality technology will be provided. Not all existing methods and tools will be presented; the focus lies on techniques that are implemented into the VRFS and will be relevant in later sections of this thesis.

In Chapter 3, Analysis and Design, a set of requirements will be gathered based on the experience of human factors practitioners, a previous prototype of the VRFS, and prior flight deck development processes. A method called use case analysis will be used to compile the requirements for the VRFS. In addition, a review of existing virtual reality hardware and flight simulation software will be conducted - two crucial parts of the Virtual Reality Flight Simulator.

In Chapter 4, Implementation, the realization of the VRFS based on the collected requirements will be presented. In particular, the development of the base system, the integration of external prototypes, and the integration of human factors methods will be described.

In Chapter 5, Evaluation, two studies will be presented with the goal of evaluating the usability and fidelity of the developed virtual environment. For both studies, a comparative experiment setup was chosen, in which the VRFS was compared with a General Aviation (GA) simulator located at the Graz University of Technology.

In Chapter 6, User Studies, the practical application of the VRFS will be demonstrated. Thus, four user studies ranging from basic research projects to more mature engineering development studies will be described. In addition, implications on the fidelity, usability, and integration into the flight deck development process will be presented.

In Chapter 7, Discussion and Future Work, the results of the usability and fidelity evaluation as well as the implications on the presented user studies will be discussed. For future user studies recommendations for the experiment design will be provided. The chapter closes with an outlook on future improvements and the further development of the VRFS. New fields of applications, in particular training applications, will be discussed in this chapter as well.

Chapter 8, Conclusion, closes this thesis by highlighting the contributions and limitations of the presented Virtual Reality Flight Simulator.

2. Background

2.1. The Human and the Modern Flight Deck

In the early pioneer years of aviation, when aircraft had very limited capabilities with regard to speed, altitude, and endurance, the workplace of a pilot was only equipped with basic instrumentation and was exposed to the elements in an open compartment. As the capabilities of aircraft improved, the complexity increased rapidly as well. This development then extended the operational limits, but eventually exceeded the human cognitive capabilities (Koonce and Debons, 2009, p. 1-1). This led to the introduction of more and more controls and instrument panels in the cockpit (Curtis, Jentsch, and Wise, 2010, p. 439) and to cockpits with a team of five or even six crew members to deal with the complex task of operating the aircraft (Koonce and Debons, 2009, p. 1-8). With the rise of jet engines in the 1950s and by leveraging computers and automation, the number of pilots in the cockpit decreased again. The disappearance of the flight engineer in the early 1980s led to the two person cockpit that is still common today in commercial aviation (Hicks and de Brito, 1999, p. 46).

The technical evolution also improved safety: Whereas a pilot in the first years of aviation still had to fear the loss of structural integrity of his or her flying machine, this cause for an accident is almost non-existent today (Koonce and Debons, 2009, p. 1-2). Other critical safety factors were unreliable complex piston engines. The replacement of those with more reliable jet engines led to a significant reduction of engine failures (Airbus S.A.S., 2014). These developments and many more increased the technical reliability of commercial aircraft to an extremely high level. As a consequence, the human factor in the flight deck received more attention. Research suggests that today almost 70 percent of all aircraft accidents are caused by or related to some sort of human error (Shappell et al., 2007, p. 228). Human error can occur on multiple levels of a complex system (Reason, 1990). Yet, the pilot in the flight deck is the last line of defense to contain these errors and prevent accidents; a fact that often exposes pilots to finger pointing if an accident actually happens, or as Reason (2000) puts it: "Blaming individuals is emotionally more satisfying than targeting institutions" (p. 768). Yet, errors can occur, can be detected, and can be resolved on every level, whether the flaw lies in regulations, organizations, operations, or in the design of the aircraft itself. For the flight deck design Harris (2004) states:

> Human error is now the main cause of aircraft accidents. However, in many cases the pilot simply falls into a trap that has been left for him/her by the poor design of the flight deck (dust cover).

Therefore, it is necessary to consider human factors aspects in all phases of the flight deck design process, a method known as human factors engineering. This implies a profound knowledge of human physiological and cognitive capabilities, i.e. human factors. With the advent of human factors research beginning at the end of the last century, multiple concepts have been developed. Selected concepts that are relevant for this research will be described in the following sections.

2.1.1. Situation Awareness

In recent years, the concept of Situation Awareness (SA) gained importance in the human factors research community (Wickens, 2008). According to Endsley (1995), most definitions of SA simply point to "knowing what is going on". A formal definition by Endsley (1988) subdivides SA into three levels: perception (L1), comprehension (L2), and anticipation (L3). L1 is the mere sensing of elements of the surrounding environment, especially their dynamics, and the status of the own vehicle (Endsley, 1995). L2 requires the subject to form a holistic picture of the surrounding based on the elements perceived in L1 (Endsley, 1995). This level of SA requires former experience, knowledge, and expertise (Sarter and Woods, 1991). L3, the highest level of situation awareness, enables a human to anticipate future events based on the SA gathered from L1 and L2 (Endsley, 1995).

In aviation, there are three important components of SA. Spatial awareness, system awareness, and task awareness (Wickens, 2002). Spatial awareness covers factors such as the awareness of the spatial position of the aircraft in the three-dimensional air space, i.e. the primary flight parameters. The status of complex systems of the aircraft is covered by the concept of system awareness. This is especially important when monitoring highly automated systems (Parasuraman and Riley, 1997). If these automated systems have different modes, mode awareness, a sub-component of system awareness, is an issue (Sarter and Woods, 1995). Task awareness describes the need to be aware of the tasks that have to be conducted to operate the aircraft (Wickens, 2002). In some cases, e.g. for system management tasks, checklists can be a way to support task awareness (Degani and Wiener, 1990).

2.1.2. Workload

According to Hart (2006), the term workload "... represents the cost of accomplishing mission requirements for the human operator ... " (p. 904). This cost can be physical, i.e. physical workload, or mental, i.e. mental workload or mental effort. Physical workload is a well defined phenomenon that has been explored in ergonomics and industrial sciences and is reflected in physiological effects like increased cardiovascular load, muscle fatigue or pain (Bokranz and Landau, 1991, p. 53).

For mental workload on the other hand, there is no widely accepted definition (Hart and Staveland, 1988). One approach is to explain mental workload with the help of information technology, i.e. describing the human as a central processor with limited resources (Gopher and Donchin, 1986). Ways to evaluate workload are commonly used methods like self-rating questionnaires (e.g. NASA Task Load Index (NASA-TLX)) or inducing and evaluating the performance of a secondary task.

2.2. Human Factors Methods

The concepts of workload and situation awareness can be assessed in experiments using human factors methods. There are quantitative and qualitative methods. Quantitative methods quantify the human factors concepts using scales or relational measures. Qualitative methods (e.g. observations or interviews) on the other hand rely on the interpretation of collected data (Creswell, 2013, p. 4). Both types of methods can be based on subjective data, i.e. feedback from the user, or objective data, i.e. data from independent sources. In this section, human factors methods will be described that are relevant for this research in order to assess workload and situation awareness.

2.2.1. Evaluating Situation Awareness

Situation Awareness Rating Technique

Selcon and Taylor (1990) presented the Situation Awareness Rating Technique (SART), a post-trial self-rating questionnaire. It uses ten dimensions to quantify a subject's situation awareness. This subjective method is non-intrusive and easy to use. It relies on a standard questionnaire that does not require any adaption to a specific use case. Yet, research suggests that such subjective methods are not valid to measure the actual SA. Instead, only a subject's perceived SA can be determined (Endsley et al., 1998).

Visual Attention

Visual attention is a metric that can serve as an indicator for situation awareness. Attention can be shifted overt, i.e. with an eye-movement, and covert, i.e. without an eye movement (Posner, 1980). An overt shift of attention can be directly measured using an eye tracking system. Modern eye tracking systems use the image of an infrared camera to identify the position of both the cornea and the retina to calculate the position of the eye gaze (Jacob and Karn, 2003). In the subject's field of vision, Areas of Interest (AOIs) can be specified, which can then be used as a basis for calculating eye tracking metrics like fixations, dwell time or the number of saccades (Poole and Ball, 2006).

In terms of situation awareness, eye fixation is only an indicator for perception (L1), i.e. visual perception is not necessarily accompanied by cognitive perception (Just and Carpenter, 1980). Nevertheless, eye tracking data is valuable for qualitative assessments of the pilots' information gathering or, as Jorna and Hoogeboom (2004), state: "The use of eye based data is attractive as it reveals the strategies that are being used by the flight crew to access the visual data available on the flight deck" (p. 266).

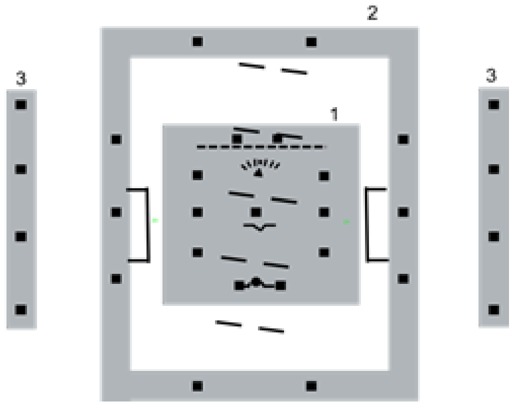


Figure 2.1.: The position of appearance of ATTENDO targets to study the distribution of attention in a head-up display (Hillebrand, Oberhauser, et al., 2014).

ATTENDO

The ATTENDO (Latin for "I pay attention") method developed by Hillebrand, Wahrenberg, and Manzey (2012) is used for assessing the allocation of attention. The method is based on the secondary task paradigm: The secondary task is the reaction to targets, which appear in the field of view of a subject for a short timeframe, usually less than 500 milliseconds. The number of perceived targets and the reaction time are metrics to evaluate the allocation of attention (Hillebrand, 2013). Figure 2.1 shows the distribution of ATTENDO targets in a head-up display.

Performance

The performance of a pilot can be another indirect indicator for situation awareness and also for workload. Performance can be quantified by observing flight parameters and comparing them with target values. E.g. a poor performance in maintaining a reference airspeed can indicate a lack of spatial SA. Yet, this lack of performance can also be caused by other factors (e.g. inappropriate workload); in other words, poor performance can occur despite accurate Situation Awareness (Salmon et al., 2006). Nevertheless, depending on the respective use case, performance data can be a valid and non-intrusive source of information.

2.2.2. Evaluating Workload

NASA Task Load Index

The NASA Task Load Index is a subjective self-rating questionnaire that was developed by Hart and Staveland (1988). It consists of six dimensions: mental demand, physical demand, temporal demand, performance, effort, and frustration. The subject has to weigh the dimensions first and then rate each of them on a scale after the trial. The questionnaire covers both mental and physical workload.

Since its introduction, the NASA-TLX has become widely adopted in numerous fields of research (Hart, 2006). De Winter (2014) even states that "... workload has now become almost synonymous with the TLX, while any attempts to launch alternative scales appear to be short lived" (p. 296). Nevertheless, like all self-rating questionnaires the NASA-TLX has some shortcomings, such as the validity of reports that are created after a trial (de Waard and Lewis-Evans, 2014), or context effects, such as the influence of previous tasks on the rating (Hart, 2006).

Physiological Measurements

Workload can lead to various physiological responses such as a change in respiration, blood pressure or Heart Rate (HR) (Veltman and Gaillard, 1998). The latter is rather easily accessible and can be derived from an Electrocardiogram (ECG). It is usually described as Beats per Minute (BPM). Another metric is the Heart Rate Variability (HRV), which describes the fluctuation of the Interbeat Interval (IBI), i.e. the time between two heartbeats. The HRV usually decreases when a subject is experiencing a higher level of mental demand (Jorna and Hoogeboom, 2004, p. 268). Mental effort is only one of many psychological and physiological factors that influence the HR. Thus, HR measurements for determining mental effort are only valid in time-limited and controlled laboratory experiments (Mulder, De Ward, and Brookhuis, 2004, p. 20-4).

Posture Assessment and Analysis

In ergonomics, a wide range of methods for posture analysis exist. In most of these methods the body posture is recorded and coded into a certain scheme; then the physical exposure is calculated. Some of these methods are pen and paper based whereas others are computer aided or even fully automated. Li and Buckle (1999) provide an extensive overview of methods for posture analysis.

2.3. Flight Simulation

In the early days of aviation, learning how to fly involved a consecutive series of flying tasks in real aircraft. Using an aircraft for training implies that it is not available for its intended military or commercial purpose. Therefore, already in the 1910s, the first mechanical training device for simulating flight characteristics, the Antoinette Simulator, was created (Page, 2000, p. 12-13). Later in 1929, Edwin Albert Link developed the first electromechanical simulator, also known as the "Blue Box" as shown in Figure 2.2 (Link, 1931). In this simulator, World War II pilots trained flying under Instrument Flight Rules (IFR) conditions with no outside visibility (Koonce and Debons, 2009, p. 1-6). In the 1960s, flight simulation became established and accepted in commercial aviation training. This led to a shorter total duration of training, a cut in training costs,

Figure 2.2.: The “Blue Box” Link Trainer photographed at the *Museu do Ar* in Sintra, Portugal.

and an improvement in safety due to a greater flexibility of training scenarios (Page, 2000, p. 12).

A modern flight simulator consists of a mathematical flight model, a system simulation, a cockpit representation, an outside visual, and an optional motion simulation (Rolfe and Staples, 1986, p. 11). In combination, these components resemble the behavior of the real aircraft, its systems, and the environment to a certain degree. This degree of realism is called simulation fidelity. According to Rehmann (1995, p. 22), simulation fidelity is an ill-defined term with no widely accepted definition. It can be divided into several subcategories like physical fidelity, functional fidelity, or psychological-cognitive fidelity (Liu, Macchiarella, and Vincenzi, 2009, p. 64). Although the research community has not agreed upon a single definition nor a consistent set of subcategories of fidelity, it is still possible to assess and compare the fidelity of a simulator with the help of qualitative categorization (Prasad et al., 1991), with subjective questionnaires (A. Robinson, Mania, and Perey, 2004), or by comparing systems based on quantitative performance data (Beard et al., 2013) on a case-by-case basis.

There is an ongoing dispute whether a simulator should be as realistic as possible or whether it should only cover some categories of fidelity depending on the purpose of the simulator (W. F. Moroney and Lilienthal, 2009). The latter opinion leads to a wide range of training simulators with different levels of fidelity, from simple desktop systems to procedure training devices to high fidelity motion-based full-flight simulators (A. Robinson, Mania, and Perey, 2004, p. 261). All these systems target an optimal simulator effectiveness, i.e. teaching the required skills as fast and inexpensively as possible (W. F. Moroney and Lilienthal, 2009, p. 19-7).

Yet, flight simulation is not only limited to training. With the rise of human factors engineering, the need for means of evaluating flight deck designs gained importance. Flight simulation is one way of doing so. Rehmann (1995) stated:

> Simulation provides an early opportunity to bring experienced flight crews into the aviation human factors design process to assess and insure, in particular, proper man/machine interfaces and workload levels (p. 3).

When using flight simulators for HF engineering in the flight deck development process, not only the level of fidelity is important; so are flexibility, cost aspects, and the system architecture. This is why user trials in simulators are an integral part of the flight deck development process.

2.4. The Flight Deck Development Process

Until the 1980s, the cockpit development process was a rigid step-by-step process without iterations or interdisciplinary cooperation. Later approaches included the participation of human factors experts and feedback loops (Reuzeau and Nibbelke, 2004, p. 40). These feedback loops proved to be effective as changes to a design should be applied as early as possible, or, as Reuzeau and Nibbelke (2004) note: "Whatever the product being designed, there is an inverse relationship between the ability to change the product and the stage of the design process" (p. 37). Thus, problems identified at a late point can be costly and often lead to compromises that do not reflect the optimal solution (Adelstein et al., 2006, p. 9). Today, a user-centered approach and the application of human factors engineering techniques are an integral part of the development process (Adelstein et al., 2006, p. 36). This includes rapid prototyping, user involvement, and iterative testing in early stages of the design process (Noyes, Starr, and Frankish, 1996).

According to Kathy H. Abbott (2014, p. 15-10), three levels of evaluation exist: compatibility, understandability, and effectiveness. For every level of evaluation an appropriate, i.e. useful and efficient, evaluation method exists, as illustrated in Figure 2.3.

At a very early stage of the design process, static paper-based sketches or interactive presentations can be the method of choice. Later, so-called part-task evaluations or simulations are used. These methods provide more detailed insights on the usability of a single system but do not consider other cockpit components or operational constraints. Or, as Rehmann (1995) states: "However, what part-task simulation may gain in experimental control it lacks in external validity, i.e. accurate representation of the real world" (p. x).

At some point in the design process, single system prototypes have to be integrated into a holistic cockpit environment. The reason, according to Reuzeau and Nibbelke (2004), being: "From the pilot's point of view, one specific sub-system is not the main focus. He or she has to work in a situation where the constraints are interlinked" (p. 37). These interlinked cockpit environments can be simulated in so-called full-task simulators, which can be engineering mock-ups, research simulators, or even high fidelity full-flight simulators. Obviously, in-service evaluations in the flying cockpit

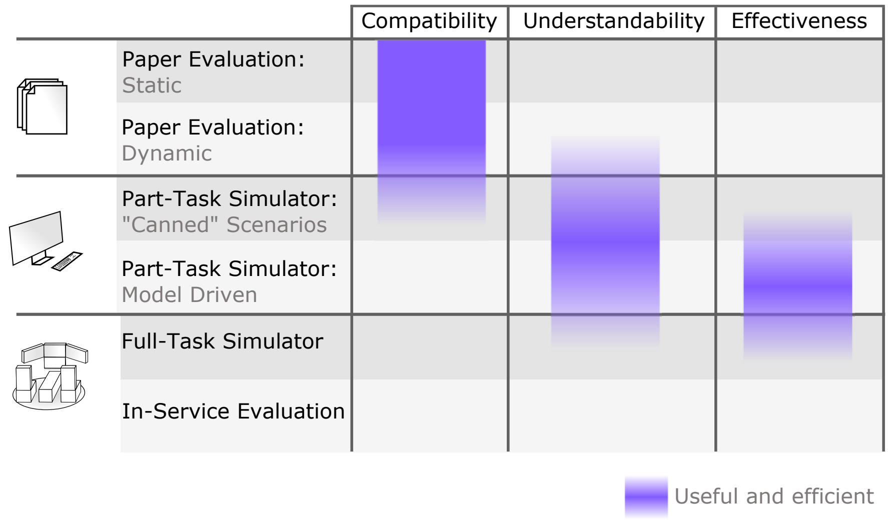


Figure 2.3.: The usefulness and efficiency of simulators for the evaluation of cockpits in the design process adapted from Kathy H. Abbott (2014) (Oberhauser, Dreyer, Convard, et al., 2016).

can be conducted as well. Yet, although this means of evaluation is useful, due to high costs for the evaluation and even higher costs for implementing changes in the cockpit, it is usually not considered as being effective (Kathy H. Abbott, 2014, p. 15-11).

The integration of single system prototypes into full-task simulators can be time consuming and cost intensive. Depending on the HMI component, extensive modifications of the cockpit and simulator equipment might be necessary. This leads to a gap between flexible and affordable part-task simulators and rather inflexible and expensive full-task simulators with limited availability (see Figure 2.4). Immersive virtual worlds can be a way to bridge this gap in the flight deck development process between single system evaluations and full-task simulators (Oberhauser, Dreyer, Mamessier, et al., 2015). With the help of virtual reality, low-cost medium fidelity environments can be created that still offer a relatively high level of flexibility.

2.5. Virtual Reality

The term Virtual Reality (VR) was coined by Damien Broderick in 1982. Since then, various definitions of the term emerged (Krueger, 1991; Steuer, Biocca, and Levy, 1993). For example, Rheingold (1991) defines virtual reality as an experience in which a person is . . .

> . . . surrounded by a three-dimensional computer-generated representation, and is able to move around in the virtual world and see it from different angles, to reach into it, grab it, and reshape it (p. 17).

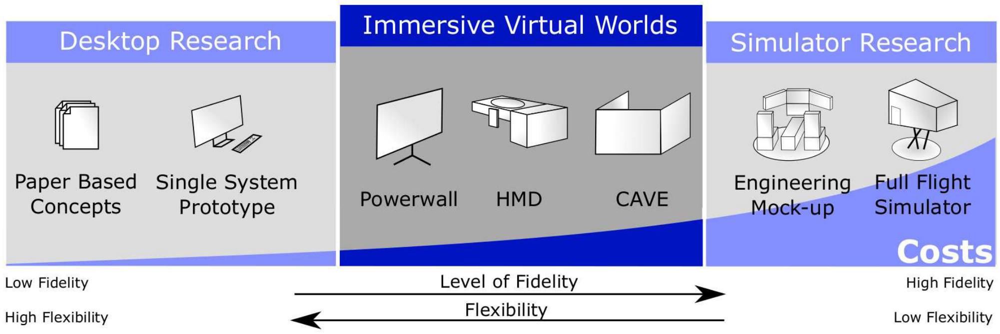
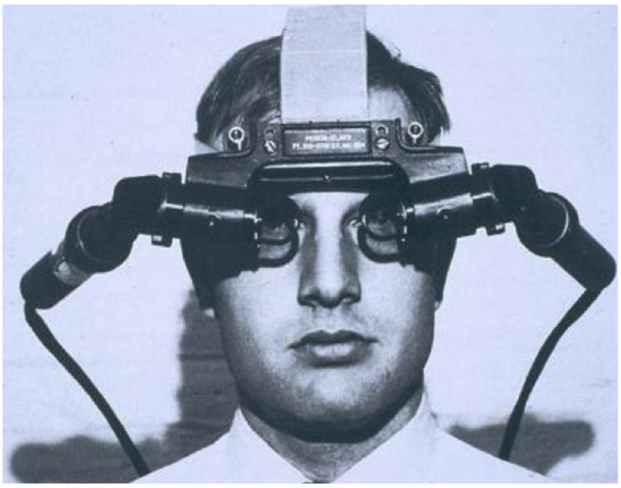


Figure 2.4.: Different types of simulators to visualize cockpit concepts. Virtual reality offers a medium flexibility and medium fidelity environment to test new cockpit concepts. (Oberhauser, Dreyer, Mamessier, et al., 2015)

The origins of virtual reality reach as far back as the 1960s when Heilig (1962) patented the Sensorama Simulator (see Figure 2.5a), a 3D display intended to "... teach and train individuals without actually subjecting the individuals to possible hazards ..." (p. 1). The first 3D HMD was developed by Sutherland (1968). This device (see Figure 2.5b), also known as "The Sword of Damocles", used head tracking based on ultrasound and a dynamic 3D image generator unit. Sutherland's work was the basis for further research (McLellan, 1996). In the early 1990s, the interest in virtual reality technology reached an unexpected climax. Numerous virtual reality companies and technologies emerged, often with an emphasis on the consumer market. However, the high expectations for this new technology could not be met at that time (Robertson and Zelenko, 2014). Thus, this first attempt to establish VR as a consumer technology failed and only a few, yet well established applications for VR in research and the industry persisted (Brooks, 1999). Today, virtual reality is on the verge of entering the consumer market, mainly in the gaming sector, once again with major hard- and software companies currently developing affordable head-mounted displays, controllers, and tracking systems (Avila and M. Bailey, 2014). Although the commercial success has yet to be proven, this recent awareness is leading to a significant technical progress in the field of virtual reality; a technology that had been written off years ago.

A modern VR system consist of a visualization system, a tracking system, and a rendering engine (Brooks, 1999, p. 16). Additional sensory stimuli like vibrations, motion, force feedback or odor can be part of the environment. Depending on the use case, the respective technologies for these three basic components and, if necessary, additional stimuli have to be selected. This combination has to meet the requirements of the individual use case and should minimize the effect of simulator sickness, the Achilles' heel of virtual reality. Simulator sickness is an effect similar to motion sickness. It is caused by a discrepancy in the visual input and the user's vestibular sensation. Causes of simulator sickness in virtual environments are, among others, delays in visualizing the user's head movements or low frame rates (Kolasinski, 1995). Consequently, delays should be minimized and a high frame rate should be targeted. Simulator sickness is an effect also experienced in conventional flight simulators.

(a) The Sensorama CC BY-SA 4.0, by Minecraftpsyco via Wikimedia Commons.



(b) The Sword of Damocles (Sutherland, 1965)

Figure 2.5.: Early virtual reality devices.

Kennedy et al. (1993) developed the standardized Simulator Sickness Questionnaire (SSQ) to quantify this effect.

In the following, selected types of visualization and tracking systems that are particularly relevant to this research will be discussed with an emphasis on their usability and on simulator sickness.

Head-mounted Displays. With the recent advances in consumer technology, head-mounted displays have almost become a synonym for virtual reality. Like the very first device by Sutherland (1968), head-mounted displays usually consist of two independent display units that are placed in front of the user's eyes. These displays can be fed by two identical (monoscopic) or by two individual (stereoscopic) images of a scene. Depending on the used HMD, the perceived image covers a certain area of the user's Field of View (FOV). Low field of view devices cover less than 40 degrees, mid field of view devices cover 40 to 60 degrees, and high field of view devices cover more than 60 degrees diagonal FOV (Cakmakci and Rolland, 2006). Today's modern consumer devices, like the Oculus Rift with its 130 diagonal FOV, still do not cover the complete human FOV of approximately 220 degrees as shown in Figure 2.6. A high field of view leads to a higher immersion into the virtual environment (Prothero and Hoffman, 1995). Yet, with today's technology, there is a trade-off between a high FOV and the display resolution, which is a factor crucial to the usability and, what is even more important, the readability in the virtual environment. Ergonomics are another factor of HMDs: According to Stanney, Mourant, and Kennedy (1998), the weight of a HMD can cause simulator sickness and even spinal injuries, in particular during long exposures.

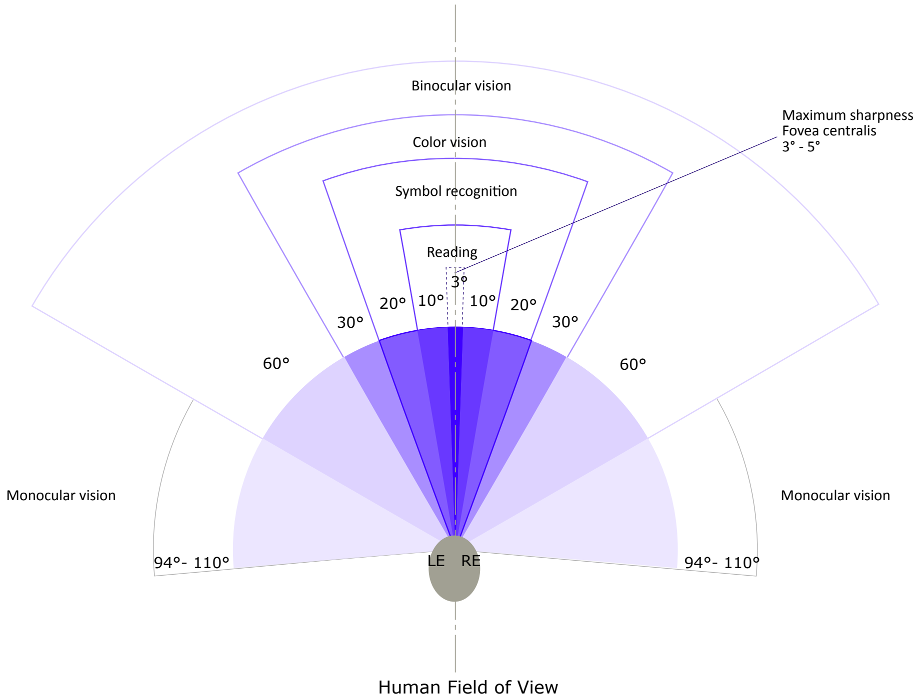


Figure 2.6.: The human field of view. Adapted from the original work by Rheto via Wikimedia Commons CC BY-SA 3.0.

CAVE Automatic Virtual Environment. Another visualization technique presented by Cruz-Neira, Daniel, and Thomas (1993) is the CAVE Automatic Virtual Environment (CAVE). In this system, a minimum of three surrounding screens with back projections are placed as cube-sides around the user. With 3D goggles and head tracking, a stereoscopic, immersive experience can be achieved. This setup can be enhanced by adding more screens up to a complete cubicle. If only one screen is used, the system is referred to as a *Powerwall*. A CAVE environment has a high field of view and creates low rates of simulator sickness. On the other hand, the setup of a CAVE is rather cumbersome, expensive, and inflexible and - if not all sides of the cube are used - these systems have a limited Field of Regard (FOR), i.e. the total area that can be captured when moving one's head (Bowman and McMahan, 2007).

Tracking System. No matter which visualization technique is used, a tracking system is crucial for virtual reality. In order to achieve a feeling of immersion, the displayed image has to be rendered according to the user's head position (Bowman and McMahan, 2007, p. 38). There are various types of tracking systems with different levels of accuracy. Accelerometers, gyroscopes, and magnetometers, commonly known from mobile phones (Lane et al., 2010, p.141) are often used to determine the orientation in consumer HMDs. These sensors have a low latency but are prone to drift errors. For professional VR applications, external tracking systems are the preferred choice due to their high accuracy. For example, optical tracking systems rely on several infrared cameras that capture reflective markers or Light-emitting Diodes (LEDs). These systems can determine the position of a marker in a sub millimeter range.

Depending on each use case, the optimal combination of hardware components has to be found. A wide range of choices with different levels of cost, accuracy, and quality exists: Whereas consumer HMDs with internal tracking systems can be purchased at a price of under 1000 Euros, a CAVE with an accurate tracking system will cost well over 100,000 Euros.

2.6. Related Work

In this section, related work that combines flight and/or cockpit simulations with virtual reality will be presented. These use cases reach from low-cost, low-fidelity consumer systems to high-fidelity virtual environments.

In the design of aircraft cockpits, today virtual reality is well known and common practice. With this technology, ergonomic aspects of a cockpit can be explored and evaluated (Goutal, 2000; Larimer et al., 1991). An additional aspect is visibility: A long running use-case in the automotive domain is the assessment of visibility limitations of different wiper configurations (Brooks, 1999, p. 22). In the aircraft cockpit domain, visibility studies focus on occlusions of displays or cockpit elements and the influence of reflections in various lighting conditions (Dreyer, 2011, p. 155). These ergonomic assessments benefit from the flexibility of the virtual environment as different cockpit configurations can be tested without the need for hardware rearrangements. Yet, static ergonomic assessments are not enough to develop highly dynamic systems like an aircraft cockpit. Cognitive human factors have to be considered as well, in particular with an emphasis on high workload situations. Therefore, a dynamic, functional virtual environment is necessary.

The virtual cockpit by McCarty et al. (1994) appears to be the first attempt to create an immersive virtual flight simulator based on a head-mounted display. For the purpose of training, a virtual cockpit has been connected to a flight simulator at the Technical University of Darmstadt by Dörr, Schiefele, and Kubbat (2001). A similar system, the Enhanced Virtual Environment (EVE), is used at Airbus Helicopters for training purposes (Bauer and Klingauf, 2008) and also for air crash investigations (Bauer, 2009; Da Silva, 2010). These two systems seem to be the most advanced in terms of immersion, functional fidelity, and maturity. Other systems, like the low-cost virtual reality flight simulator presented by Yavrucuk et al. (2009), the virtual reality flight simulator by Persiani, Piancastelli, and Liverani (1997), or the rapidly reconfigurable research cockpit presented by Joyce and S. K. Robinson (2015) are early prototypes or proofs of concepts. This work is based on another proof of concept by Aslandere, Dreyer, Pantkratz, et al. (2014). Here, a Commercial Off-the-shelf (COTS) flight simulator is extended with virtual reality capabilities.

There already exists research on the usability of virtual cockpits: Hüsgen and Klingauf (2005) presented different interaction concepts with fully virtual buttons and conducted a usability study. The effect of virtual button sizes was investigated by Aslandere, Dreyer, and Pankratz (2015). However, these two studies are applicable for a specific system and use case only. More general studies on usability in virtual environments were conducted by Bowman and Hodges (1999).

In contrast to the existing simulators, which mainly focus on training purposes, this research will focus on the rapid integration of HMI components into a virtual cockpit and the evaluation of these components using human factors methods. Hence, a virtual cockpit, which can be redesigned and reconfigured, with a sufficient fidelity has to be provided.

3. Analysis and Design

In the previous chapter, fundamental concepts of the flight deck development process, flight simulation, and virtual reality technology have been discussed. In this chapter, requirements for the VRFS, which reflect the experience from the flight deck development process, will be specified. In order to put these requirements into practice, two components are crucial: the flight simulation software and the virtual reality hardware. Hence, an analysis of current technologies and products in this field will be conducted. The requirements and the conducted analysis will be the basis for the following chapter: Implementation.

3.1. Requirements and Use Cases

In order to gather the requirements for developing the Virtual Reality Flight Simulator, a user centered approach based on use cases has been chosen. With this common approach, the user of the system and his or her intentions, goals, and expectations are the basis for the development (Lee and Xue, 1999). Use cases capture a set of functional and non-functional requirements and are used for the design, implementation, and evaluation of a system (Bittner and Spence, 2003, p. 11). In the latter sections of this thesis, the author will refer to the use cases instead of the underlying set of requirements.

The first step of use case modeling is the identification of relevant stakeholders or actors. For the VRFS, these are:

- The pilot,
- the simulation controller,
- the human factors practitioner, and
- the developer.

For modeling and visualizing the use cases, the Unified Modeling Language (UML) will be used (Booch, Jacobson, and Rumbaugh, 2004). Figure 3.1 shows the modeled use cases and the involved users. In the following, the users and their respective use cases will be presented:

The Pilot. He or she is immersed in the virtual environment. From his or her perspective, the interaction with the cockpit, i.e. the interaction with displays (Use Case 1.5), levers (Use Case 1.4), and buttons (Use Cases 1.1, 1.2 and 1.3) is relevant. There can be a second pilot using the system simultaneously; thus, the interaction with the second pilot should be feasible as well (Use Case 1.6).

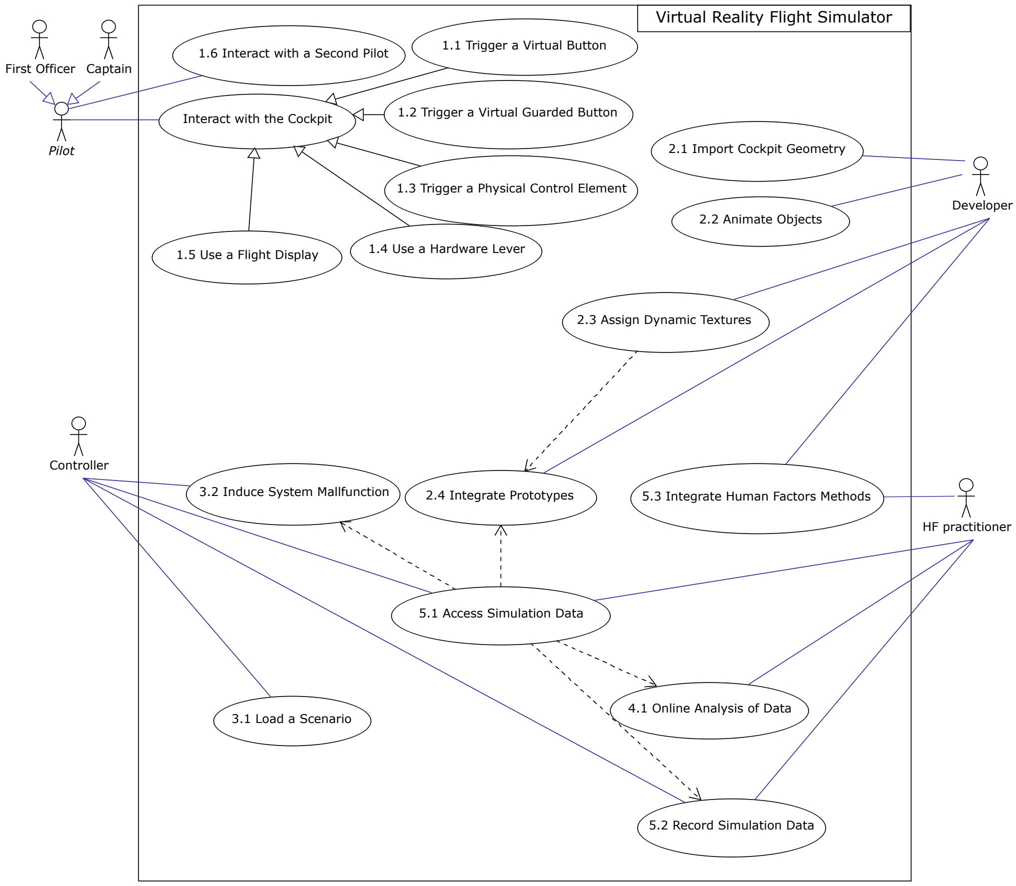


Figure 3.1.: The UML use case diagram for the Virtual Reality Flight Simulator.

The Simulation Controller. He or she is supervising the simulation from a technical and operational point of view. The controller has to set up the aircraft simulation, the surrounding environment, and the virtual cockpit. This means, he or she has to initiate a predefined scenario (Use Case 3.1). During the simulation session the controller has to monitor the progress of the simulated flight. Depending on the scenario, the controller has to influence the environment or the system simulation, for instance by inducing simulated system failures (Use Case 3.2). Thus, the possibility to monitor and manipulate simulation data is a necessary requirement (Use Cases 5.1 and 3.3). For further processing, the controller should be able to trigger the recording of the simulation data (Use Case 5.2).

The Human Factors Practitioner. He or she needs to have access to data, including simulation data, physiological data or data from integrated human factors methods (Use Case 4.1). The data should be recorded for a following evaluation or should be evaluated live during a trial (Use Cases 5.1 and 5.2).

Number	**1.1**
Name	**Trigger a Virtual Button**
Goal	The user shall be able to push a fully virtual button and get visual feedback regarding the state of the button.
Precondition	The simulation system and X-Plane is up and running. The hand is fully tracked.
Postcondition	The button has changed its indication
Actors	Pilot
Scenario	1. The user moves his/her finger to the virtual position of the button. 2. The user's hand turns green while colliding with the button. 3. The button changes its indication to a different color or texture.
Alternative Scenario	02a. While colliding with the button the button geometry is translated. 03a. Eventually the button indication is changed by the flight simulation system to a different indication.
Non-Functional Requirements	The collision with the button geometry should be easy to accomplish for the user. It should be error tolerant. No double executions should be possible.

Figure 3.2.: The detailed description of the use case *Press a Virtual Button*.

The Developer. He or she is responsible for integrating new cockpit designs. Geometry data of these cockpit designs has to be converted and textured (Use Case 2.1); certain elements have to be animated according to system states (Use Case 2.2). The developer might also have to integrate third party prototypes into the VRFS (Use Case 2.4). Thus, an interface with these external components has to be provided. This interface shall include bi-directional data access and the possibility to stream image content into the simulation (Use Case 2.3).

For every use case, a detailed description is created (Bittner and Spence, 2003, p. 225) as shown in Figure 3.2. In this description, the acting user and his or her functional goal is defined. A detailed step by step scenario to reach this goal, including alternative process steps, is part of the description. However, not only the functional goal of a use case is relevant, but also non-functional requirements, such as usability or performance, are important for the success of a technical solution. Table 3.1 shows an overview of all identified use cases and their functional goals. All use case descriptions can be found in the appendix on page 163.

3.2. Review of available Flight Simulation Software

In order to build a flight simulator, no matter whether it is a hardware simulator or a fully virtual one, an appropriate flight simulation software has to be chosen. Today, a variety of flight simulation environments exist. Starting in the 1980s, the first affordable commercial flight simulation software products were developed for home and professional use. Due to years of development, these products have reached a high

Table 3.1.: An overview of the identified use cases of the Virtual Reality Flight Simulator.

No.	Name	Goal/Functional Requirement
Pilot Use Case		
1.1	Trigger a Virtual Button	The user shall be able to push a fully virtual button and get visual feedback regarding the state of the button.
1.2	Trigger a Guarded Virtual Push-button	The user shall be able to push a fully virtual button that is covered by a protective cap and get visual feedback regarding the state of the button.
1.3	Trigger a Physical Push-button	The user shall be able to push a physical button that also has a virtual representation and get visual feedback regarding the state of the button.
1.4	Use a Hardware Lever	The user shall be able to use a lever in the cockpit and see the change of its position or orientation interactively.
1.5	Use a Flight Display	The user shall be able to get information of the flight status from a display interactively.
1.6	Interact with a Second Pilot	The user shall be able to see and interact with a second pilot in the cockpit.
Developer Use Case		
2.1	Import Cockpit Geometry	The developer shall be able to use geometry for the virtual cockpit in commonly used 3D data formats.
2.2	Animate Objects	The developer shall be able to bind a variable to an animated object.
2.3	Assign Dynamic Textures	The developer shall be able to map a dynamic texture to a mesh/object.
2.4	Integrate Prototypes	The developer shall be able to integrate an external component/prototype.
Controller Use Case		
3.1	Load a Scenario	The supervisor shall be able to load a preconfigured scenario.
3.2	Induce a System Malfunction	The controller shall be able to induce a predefined behavior of the aircraft that simulates a system malfunction (e.g. engine fire).
Human Factors Practitioner		
4.1	Online Analysis of Data	The HF practitioner shall be able to create a component that analyzes simulation data or data about human behavior during run-time.
Shared Use Cases		
5.1	Access Simulation Data	The controller and the HF practitioner shall get access (read and/or write) to a variable that is available in the VRFS.
5.2	Record Simulation Data	The controller and HF practitioner shall be able to record all data that is available in the VRFS.
5.3	Integrate Human Factors Methods	The engineer in cooperation with the HF practitioner shall be able to integrate HF methods.

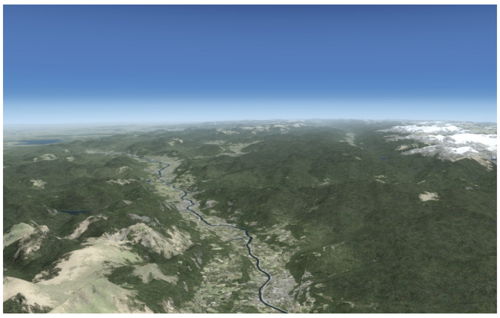
Table 3.2.: Feature comparison of different flight simulation software.

	Prepar3D	X-Plane 10	FlightGear	Outerra	Google earth
Company	Lockheed Martin	Laminar Research	FlightGear	Outerra	Google
Name	Prepar3D	X-Plane	FlightGear	Outerra	Google Earth
Physical Fidelity					
Outside Visual	High	High	Medium	High	Low
3D **Cockpit**	High	High	Medium	High	N/A
Flight Model	High	High	Medium	Medium	Low
Audio	High	High	Medium	Low	Low
Terrain	High	High	High	High	High
Low-altitude LOD	Medium	Medium	Medium	High	Medium
Infrastructure	Medium	Medium	Medium	Low	High
Airports	High	High	High	Low	Medium
Functional Fidelity					
Aircraft Systems	High	High	Medium	Low	Low
Navigation Aids	High	High	High	N/A	N/A
Traffic	Medium	High	Low	N/A	N/A
Development					
Modifiability	Medium	Medium	High	Low	Low
Cost	Medium	High	Low	N/A	Medium

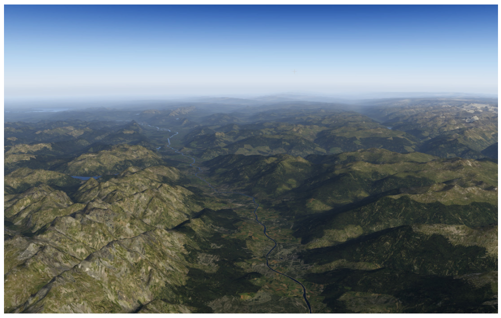
grade of maturity. Craighead, R. Murphy, and Burke (2007) conclude that it is "... no longer necessary to build a new simulator from scratch" (p. 856).

The two most popular commercial flight simulations for home users are X-Plane (XP) and Microsoft Flight Simulator (MSFS), which still has a large and active user base, although the development stopped in 2009. The successor of the MSFS is Prepar3D (P3D) by Lockheed Martin, which acquired the intellectual property from Microsoft. P3D offers all the features of MSFS and is being developed further. Thus, MSFS will not be considered in this review. Other flight simulations, such as the military simulation DCS World, were not considered in this review as they do not feature a global world map.

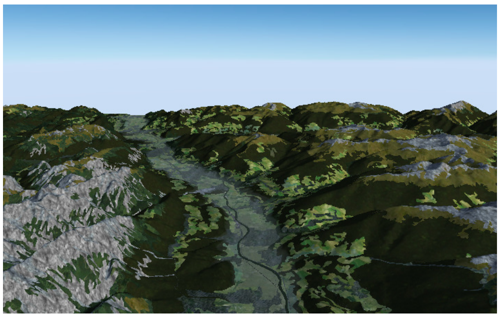
These simulators are popular among home users but are also used for professional applications in training, research, and even aircraft design. Especially for research applications, the open source flight simulator FlightGear is a popular choice. In addition, the open world simulations Outerra and Google Earth are part of this review. Although both are not fully dedicated flight simulations, they provide some specific advantages that will be discussed in more detail in this section.



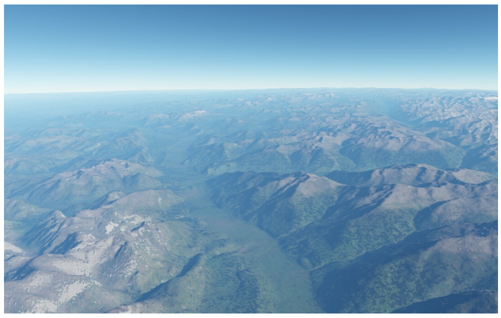
(a) Prepar3D



(b) X-Plane



(c) FlightGear



(d) Outerra

Figure 3.3.: Comparison of the outside visual at cruise altitude.

The fidelity of these simulations as well as other non-functional factors are compared in a qualitative assessment. Table 3.2 shows the overall results.

3.2.1. Physical Fidelity

Physical fidelity is the degree to which a simulation resembles real world phenomenons (Alexander et al., 2005). It includes the visual representation, the accuracy of the flight model as well as other stimuli. For this review, the physical fidelity is divided into subcategories for the representation of the outside visual, the 3D cockpit, the flight model, audio stimuli, infrastructure representation, and airports.

Outside Visual. The basis for an exact representation of the earth's topography are elevation data. These elevation data points are then connected and form a three-dimensional mesh. If a ground texture, like an aerial photograph, is mapped on this mesh, a realistic three-dimensional impression of the topography can be created. Not only the resolution of the elevation data and the ground texture are important for the overall visual quality but also visual effects like atmospheric scattering, a light simulation, and water reflections. Figure 3.3 shows the topography rendering of several flight simulators. Both MSFS/P3D and X-Plane offer a high visual quality at

(a) X-Plane (b) Outerra (c) FlightGear

Figure 3.4.: Comparison of the visualization of 3D cockpits.

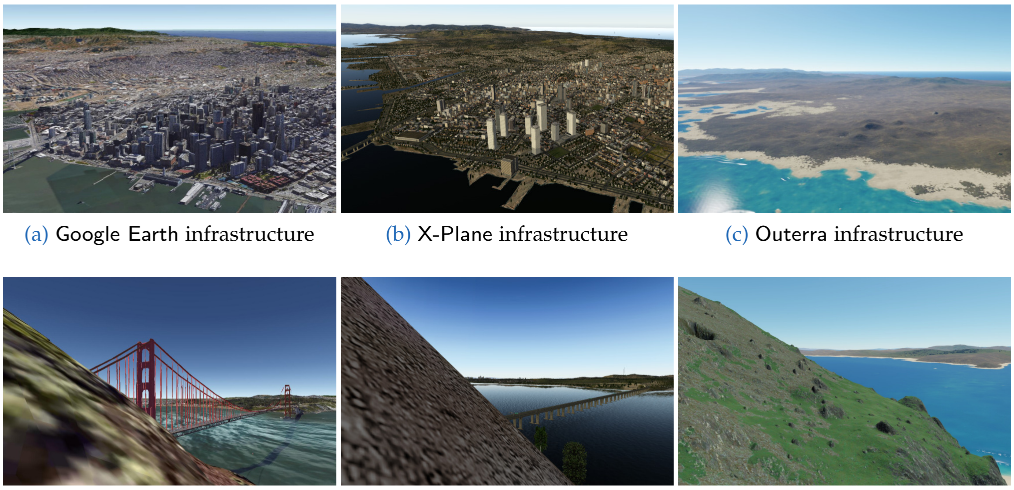
cruise altitude and well-balanced visual effects. FlightGear also offers a high quality mesh but the visual effects are poorer. The visual quality of Outerra is very high as well. Especially the level of detail close to the ground, as shown in Figure 3.5f, is an advantage of this engine (Cozzi and Ring, 2011, p. 369). Nevertheless, topographic features, such as rivers and small lakes, are missing.

Another feature that leads to a realistic representation of the outside visual is cloud simulation. Prepar3D, X-Plane, and FlightGear offer a simulation with extensive configuration options. Outerra has limited cloud simulation capabilities, Google Earth does not offer cloud simulation features at all.

3D Cockpit. Except for Google Earth, all presented flight simulators offer a 3D cockpit. The visual impression of these cockpits highly depends on the used 3D data and textures (see Figure 3.4). Another factor is the cockpit functionality: Levers, instruments, and displays have to be animated according to aircraft system states in order to provide the user with feedback that is needed to operate the simulated aircraft. The simulations Prepar3D and X-Plane offer a variety of high fidelity 3D cockpits. Outerra features only a few high quality 3D cockpits. Compared with the other simulations, the 3D cockpits in FlightGear are rather simplified.

Flight Model. Prepar3D uses stability derivatives and lookup tables to predict the aircraft behavior. X-Plane leverages the blade element theory to calculate force vectors based on the airflow around the aircraft geometry. Thus, it is possible to simulate the flight model without having stability derivatives available. X-Plane's flight model was even used as an engineering tool in the SpaceShipOne development, an experimental sub-orbital spaceship (Reibold, 2011). A comparison of MSFS/P3D and X-Plane to real flight test data was conducted by Babka (2011), who concludes that "...both simulators have extremely accurate models when compared to the actual aircraft" (p. 16).

FlightGear offers multiple flight models with different levels of complexity and fidelity (Perry, 2004); the standard flight model engine used is JSBsim, the simulation that is also being used in Outerra. JSBsim is an engine based on C++. It uses a configuration that allows the definition of aircraft controls, engines, and other systems (Berndt and Marco, 2009). Although numerous applications of JSBsim in the industry and academia exist (Berndt and Marco, 2009), no study about the fidelity of the flight model is



(a) Google Earth infrastructure (b) X-Plane infrastructure (c) Outerra infrastructure

(d) Google Earth at low altitude (e) X-Plane at low altitude (f) Outerra at low altitude

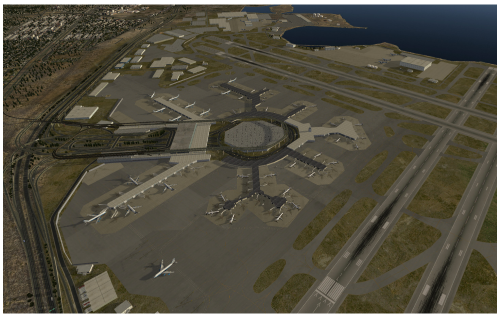
Figure 3.5.: Comparison of the infrastructure visualization and the level of detail at low altitude.

available. The flight model of Google Earth is simplified and only offers the dynamics of two aircraft. Therefore, the fidelity is considered as being low.

Audio. Sound effects are an important factor for increasing the immersion of a simulation. Inaccurate audio stimuli can have a negative influence on the flight performance of a pilot (Jarvis, Lalonde, and Spira, 2008). Prepar3D, X-Plane, and FlightGear offer an accurate sound representation of engines, aircraft systems, and the environment. Outerra only features a very basic sound simulation and Google Earth does not offer sound effects at all.

Infrastructure. The accurate representation of settlements, roads, or power lines is important in a flight simulation, especially when conducting Visual Flight Rules (VFR) navigation or a visual approach. Google Earth has an up-to-date database of streets, rails, and buildings. Thus, the quality of the infrastructure representation is rated as being high. Prepar3D, X-Plane, and FlightGear have means to represent infrastructure features. For example, X-Plane focuses on auto-generated objects based on aerial photographs and data from Openstreetmap. This auto-generated data often is a poor representation of the actual infrastructure, in particular at low altitudes. Figure 3.5 (top) shows a comparison of the visualization of the city of San Francisco in Google Earth, X-Plane, and Outerra. Google Earth shows the city with diverse detailed 3D buildings and streets. X-Plane generates the city infrastructure with generic buildings and roads, Outerra does not inherently come with infrastructure features. FlightGear and Prepar3D use an approach very similar to that of X-Plane. Figure 3.5 (bottom) shows the Golden Gate Bridge from the Battery Spencer Viewpoint: Google Earth shows an accurate representation of the landmark, X-Plane shows a generic bridge and Outerra shows no bridge at all.

(a) Google Earth



(b) X-Plane

Figure 3.6.: Comparison of the airport infrastructure.

Low-altitude Level of Detail. The level of detail close to the ground is of particular interest when flying rotary wing aircraft. Here, a high quality texture and mesh are important as well as visual cues like trees, stones, or bushes. X-Plane, FlightGear, and Prepar3D feature auto-generated trees. Outerra generates a high-fidelity representation of the ground with details like stones, bushes, and even grass that is moved by the wind, as shown in Figure 3.5f. Google Earth relies solely on a mesh and aerial photography textures.

Airports. The airport infrastructure is of particular interest in a flight simulation as it is necessary for simulating a take-off or landing and for taxiing. Hence, not only the runways should be modeled and aligned correctly, but taxiways and gates should be present as well. These elements are important for ground operations, i.e. they are visualized in a close proximity. Thus, a high level of detail is necessary and objects like runway lights, markings, signs, or runway stop bars should be clearly visible for the user. The use of areal photography textures for airports, as in Google Earth, lacks the necessary level of detail to show runway markings and cannot visualize static runway signs nor functional stop bars or airport lights. In contrast, in flight simulations like Prepar3D, X-Plane, and FlightGear the airport infrastructure is remodeled with dedicated tools and with the use of 3D objects. This might decrease the visual accuracy of the airport (see Figure 3.6) but for the purpose of simulation, the functional aspects and the level of detail on ground are more important. Outerra uses the same remodeling approach. Yet, only a small number of airports is available here.

3.2.2. Functional Fidelity

Alexander et al. (2005) define functional fidelity as "...the degree to which the simulation acts like the operational equipment in reacting to the tasks executed" (p. 4). In this review, the relevant aspects of functional fidelity are the aircraft systems, the functionality of navigation aids, and traffic simulation.

The Aircraft System Simulation. The available aircraft system simulation highly depends on the used aircraft model. Prepar3D and X-Plane offer a large number of commercial aircraft models with detailed system simulations. FlightGear depends on its community to offer new flight models with limited possibilities for commercialization. Therefore, only a limited number of high quality aircraft models are available in comparison to Prepar3D and X-Plane. Nevertheless, Prepar3D, X-Plane as well as FlightGear offer the possibility to extend or modify existing and to create new aircraft system simulations. Both, Outerra and Google Earth do not offer system simulations besides the primary flight controls.

Navigation Aids. The realistic functionality of navigation aids like Instrument Landing System (ILS), Global Positioning System (GPS), or Inertial Navigation System (INS) is crucial for the operational aspects of a simulated flight. Outerra and Google Earth do not offer such functionality; all other flight simulations have a realistic representation of navigation aids based on up-to-date databases. Databases that reflect the latest changes are available for ground based navigation aids as well.

Traffic. In order to provide a realistic operational environment, other aircraft have to be present in the virtual airspace. Prepar3D, X-Plane, and FlightGear offer a native traffic functionality. These simulators use Artificial Intelligence (AI) to simulate traffic and the Air Traffic Control (ATC) communication. Outerra and Google Earth do not have such functionality.

3.2.3. Ease of Development

Modifiability. X-Plane and Prepar3D offer tools for scenery and aircraft development and allow including custom code via a plug-in Application Programming Interface (API) and a Software Development Kit (SDK). Extensive modifications are possible in both simulators but the capabilities and features of the respective API or SDK are a limiting factor. FlightGear on the other hand is open source, which means that it can be fully customized. Outerra and Google Earth offer very limited options for modifications.

Cost. Prepar3D and X-Plane are commercial products in a consumer price segment. Google Earth and Outerra offer limited demo version with a low-price update option. FlightGear is free.

3.2.4. Conclusion

Both Google Earth and Outerra are engines mainly for rendering a 3D globe with strengths and weaknesses. Google Earth offers a high level of fidelity when it comes to infrastructure and terrain with up-to-date data and an active user community. The flight simulation module is very basic and some fundamental features like weather simulation, navigation aids, or a 3D cockpit are missing. Outerra has a high visual quality with an infinite level of detail. Yet, it is missing some topographic features and infrastructure details have not been implemented, yet.

The three native flight simulations Prepar3D, X-Plane, and FlightGear have a high functional fidelity. Yet, the physical fidelity, especially the visual quality of FlightGear, has some weaknesses. Prepar3D and X-Plane offer almost the same functional and physical fidelity. Both simulations provide means for modifications to some degree, certainly enough for the proposed Virtual Reality Flight Simulator.

3.3. Review of available Virtual Reality Hardware

In order to put the previously defined use cases into practice, not only the software implementation is necessary. An appropriate virtual reality hardware selection is mandatory, especially in order to fulfill non-functional requirements like usability, visual quality, or the prevention of simulator sickness.

Yet, some limitations apply to the hardware selection: As described in the introduction to this research, the VRFS has to be integrated into an existing virtual reality laboratory. In addition, the budget for purchasing virtual reality hardware is limited. In the following, a review of the existing virtual reality hardware in the laboratory as well as a market survey of HMDs with regard to the applicability to the VRFS will be conducted, considering the mentioned restrictions.

3.3.1. The Tracking System

An optical tracking system, the ARTTRACK 2, from German hardware supplier Advanced Realtime Tracking (ART)[1] is already available in the laboratory. The system consists of eight cameras, two mounted on the floor and six mounted on the ceiling. Although only two cameras are necessary in order to use the system, more cameras help to deal with occlusions and increase the accuracy. The tracking system emits Infrared (IR) light and captures the reflection from passive retro reflective targets. A central controller processes this information and calculates the orientation and position of these targets. It is also possible to use active markers; these markers emit infrared light and do not rely on a minimum reflective surface. Therefore, they can be smaller than passive targets.

The ART system also offers a finger tracking solution. The hand as well as each individual finger are tracked with active markers (Advanced Realtime Tracking GmbH, 2016, p. 137). Based on this data and an initial calibration, the controller software delivers a complete kinematic model of the user's hand.

The tracking system that is used at the virtual reality laboratory has a frame rate of 60 Hz, i.e. the number of emitted IR flashes per second. The time from sending an IR flash until the data is available via Ethernet is 17 to 19 milliseconds. With the eight cameras, an effective tracking area of 3.5 to 5 meters is covered. For a rather

[1] http://www.ar-tracking.com

Table 3.3.: Market survey of virtual reality head-mounted displays, as of mid-2017.

Company	Sensics	nVision	Vrgineers	Starbreeze	Fove	HTC	Oculus	Sony	Avegant
Name	dSight	nVisor SX60	VRHero 5K Plus	StarVR	Fove	Vive	Rift	HMZ T-2	Glyph
Display Type	TFT	LCOS	OLED	N/A	OLED	OLED	OLED	N/A	LED
Resolution (per Eye)	1920x1080	1280x1024	2560x1440	2560x1440	1280x1440	1080x1200	1080x1200	1280x720	1280x800
FOV Diagonal	143	60	170	247	100	110	110	45	45
Pixel Density (per Degree)	15.4	27.3	17.28	11.89	19.26	14.67	14.67	32.63	33.54
Geometrical Distortion	No	No	Yes	Yes	Yes	Yes	No	No	
Video Input	Dual HDMI, Single HDMI	Dual DVI, Single DVI	DisplayPort and USB	HDMI and USB	HDMI and USB	HDMI and USB	HDMI and USB	HDMI	N/A
Weight (g)	570	1000	N/A	850*	520	470*	555*	330	850*
Eye Tracking	Optional: Dikablis integration	Custom Airbus Group Innovations integration	No	Tobii Eye Tracking integration	2x infrared eye trackers	No	Optional: SMI Eye Tracking integration	No	
Price (in Euro)	12,000	10,000	N/A	N/A	500	700	450	500	550
Availability	In production	Out of production	Prototype available	Mid 2018	In Production	In Production	In Production	In Production	In Production

* Retrieved from http://www.vrnerds.de/vr-brillen-vergleich

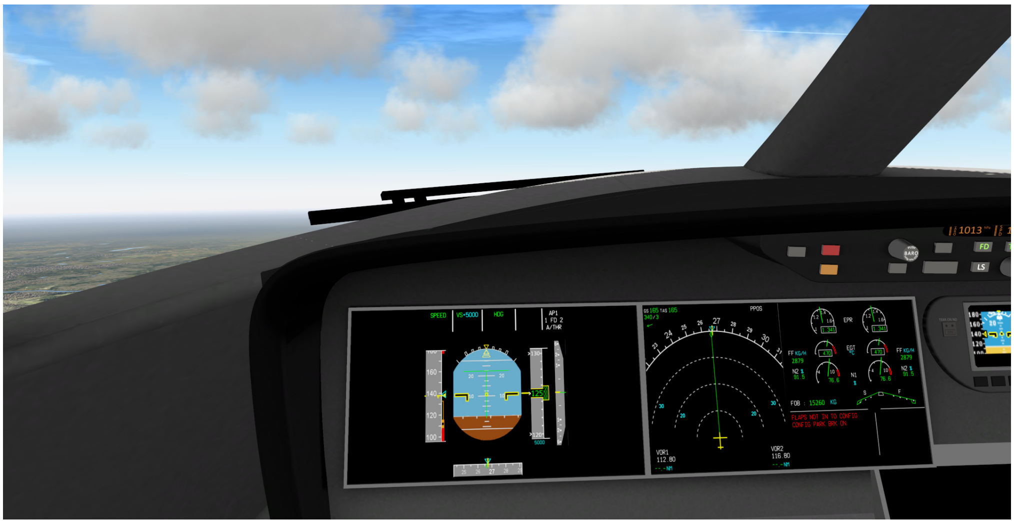
small virtual commercial aircraft cockpit, this is more than sufficient. Nevertheless, if hardware elements are added to the cockpit and two pilots interact in the VRFS, the line of sight of some cameras might be occluded. The extra cameras help maintain the tracking in these situations.

3.3.2. Powerwall

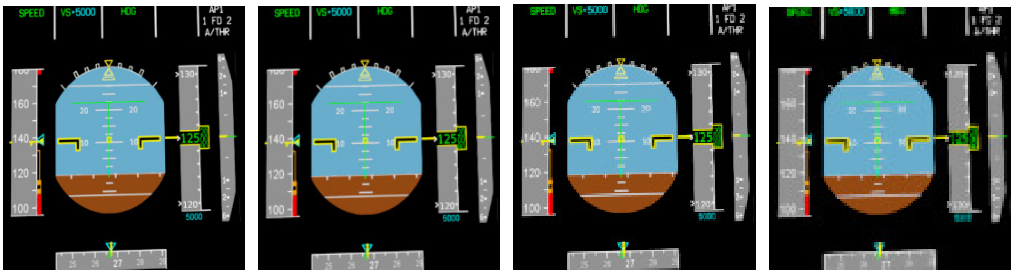
A high resolution stereoscopic projector, a so-called powerwall, is available at the virtual reality laboratory. The Barco F35 projector has a resolution of 2560x1600 pixels. The image is projected to a screen of 2.10 meters height and 2.80 meters width. With active 3D shutter glasses, a three-dimensional impression can be achieved (Barco, 2016). A multi-sided CAVE, which would offer the advantage of a higher field of regard, is not available in the laboratory and cannot be integrated due to spatial and budget constraints.

3.3.3. Head-mounted Displays

Due to the recent interest in virtual reality technology, various new devices have been developed or are currently under development. Rude (2014) presented an excessive



(a) View with 100 degrees field of view and a pixel density of 30 px/°



(b) 25 px/° (c) 20 px/° (d) 15 px/° (e) 10 px/°

Figure 3.7.: The effect of the pixel density on the readability of display content.

overview of head-mounted displays and devices. Due to the dynamics of this technology, the overview conducted by Rude (2014) as well as the overview presented in this study will soon be outdated. A continually updated list of head-mounted displays is maintained by VRNerds (2016) and Data Reality (2016).

Table 3.3 shows selected professional and consumer head-mounted displays. The models Fove, VR Hero 5K, and Vive leverage Organic Light-emitting Diode (OLED) technology, which was already used in the first Rift prototype. All these devices rely on a distortion of the input image in shape and color to counter distortion and light refraction effects from the optics. The dSight head-mounted display is similar to these devices but uses a Thin-film Transistor (TFT) display and more sophisticated optics, without the need for image distortion. HMZ T-2 and Glyph use a similar technology but offer a lower FOV. A professional device, the Nvisor SX60 uses a Liquid Crystal on Silicon (LCOS) display. For this device, a custom eye tracking solution was developed at Airbus Group Innovations (Liesecke, 2013). It is worth mentioning that some other devices like dSight, Fove, Rift, and StarVR already have native support for eye tracking

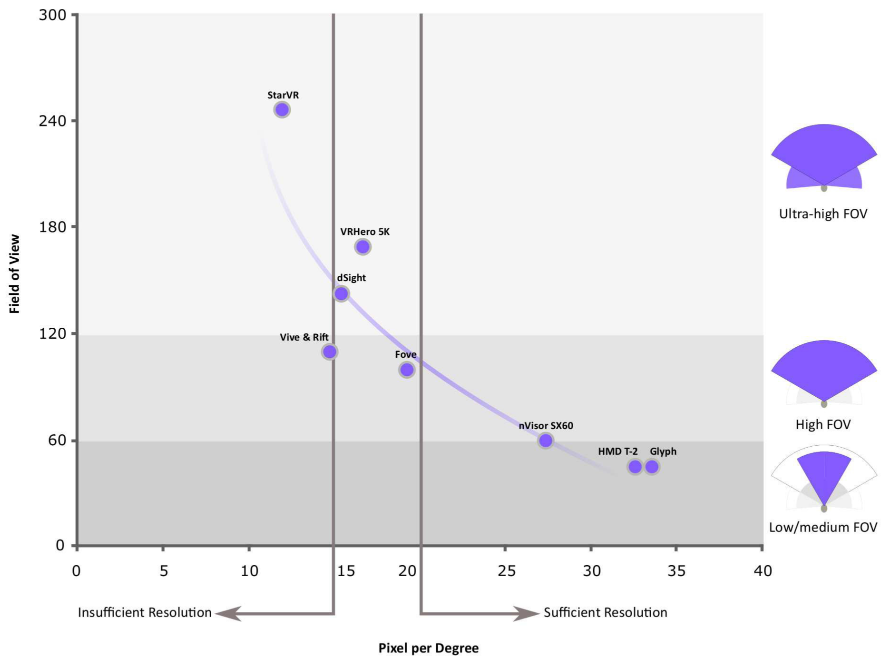


Figure 3.8.: The correlation of field of view and pixel density.

or are cooperating with manufacturers for eye tracking devices. This fact underlines the relevance of this technology for professional and consumer applications.

When selecting a head-mounted display for a virtual reality flight simulation application, special attention has to be paid to the field of view and the pixel density, i.e. the pixels per degree field of view. The peripheral vision is an important stimulus for the vestibular system as it induces an illusion of self-motion and increases the level of immersion (Brandt, Dichgans, and Koenig, 1973). The pixel density is relevant for the readability of displays in the virtual cockpit. Figure 3.7a shows a virtual cockpit from the pilot's Design Eye Reference Point (DERP) with a field of view of 100°. In this first image, a resolution, which leads to a diagonal pixel density of 30 px/°, is used. The image resolution is gradually degraded and the effect on the readability of the primary flight display is visualized. Based on this test, it can be assumed that a pixel density of 15 px/° to 20 px/° is the minimum necessary for using the virtual cockpit.

There is a correlation between the field of view of a device and the pixel density, as shown in Figure 3.8. The resolution of display panels is limited by technological constraints, spatial constraints, and computing power. Hence, a limited amount of pixels has to be spread across a certain field of view. This leads to a trade-off between low field of view devices with high pixel density and high field of view devices with low pixel density. As 15 px/° are used as an absolute minimum, the HMDs StarVR, Vive, and Rift have to be excluded for the VRFS. Due to their low field of view, the models HMZ T-2 and Glyph have to be excluded as well. Fove, VRHero, and dSight seem to offer a borderline pixel density that requires further investigation. Only the nVisor SX60 meets the requirements with regard to field of view and pixel density.

4. Implementation

Based on the analysis of virtual reality hardware, flight simulation software, and tracking systems in Section 3, several hard- and software choices were made. With regard to the flight simulation software, X-Plane was chosen as it offers, along with Prepar3D, the highest fidelity. The optical tracking system from ART that was already available in the facilities is used for head and finger tracking; for visualization, the nVisor SX60 HMD offers the ideal pixel density. Figure 4.1 shows the architecture of this typical setup of the VRFS.

All components communicate by means of the Robot Operating System (ROS), a network framework initially developed for distributed robotic applications (Quigley et al., 2009). Due to this middleware and its modular implementation, existing components of the VRFS are interchangeable and new components can be added anytime if necessary. Thus, the VRFS can be be easily updated as hard- and software evolves and it can be adapted to particular use cases. In this section, the implementation and the architecture of the Virtual Reality Flight Simulator with its standard components will be presented. Besides this base system, the integration of additional hard- and software components will be described. This setup, integrated prototypes, and human factors methods are the basis for the use cases presented in Section 6.

The implementation has to fulfill the requirements that have been identified in the use cases in Section 3.1 (page 19).

4.1. The Robot Operating System

The Robot Operating System is used as a network framework for the VRFS and is one of its core components. At its basic level, ROS is a communication infrastructure that allows a program (node) to send (publish) or receive (subscribe) messages via a network connection. A central server, the so-called roscore, acts as a coordinating instance. New nodes register at the roscore and a peer to peer connection is established between these two nodes. The communication is established either using the Transmission Control Protocol/Internet Protocol (TC/IP) or the User Datagram Protocol (UDP).

4.1.1. Reasons for Using the Robot Operating System

In the following paragraphs, some key features will be described that are the reason for choosing ROS as a network middleware for the VRFS.

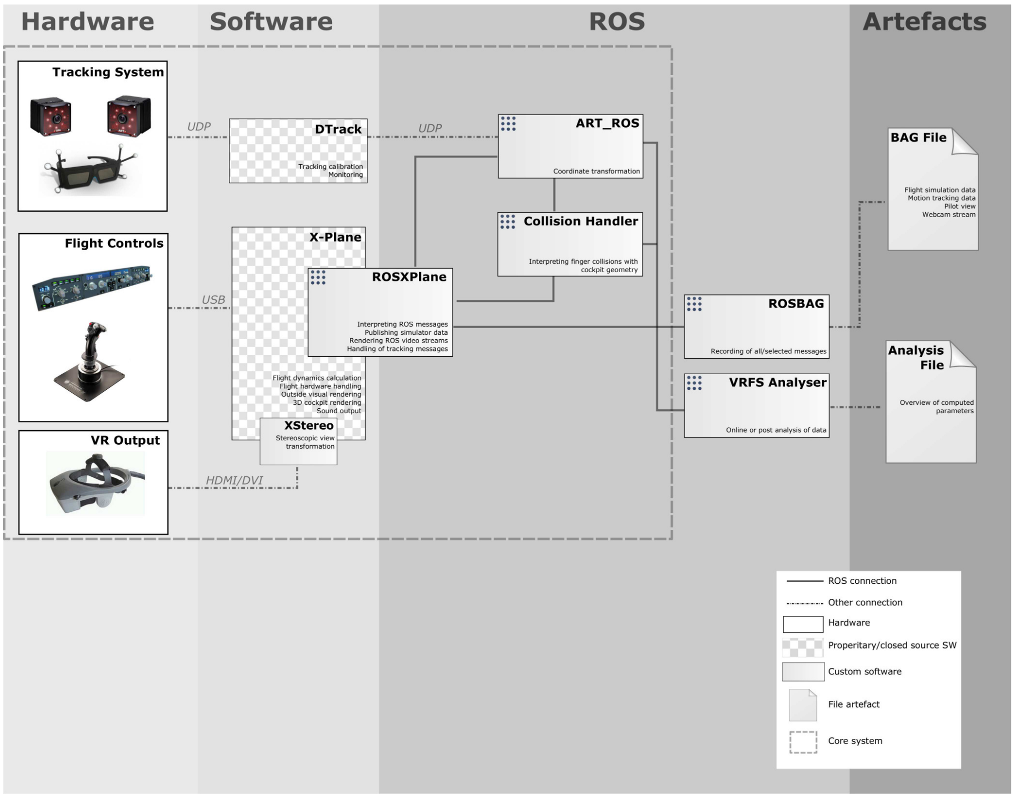


Figure 4.1.: The architecture of the VRFS in its typical configuration. The core system consists of X-Plane, an optical tracking system, hardware elements, and a head-mounted display.

Listing 4.1: A simple ROS publisher written in Python.

```
import rospy
from std_msgs.msg import String

# Initiate Topic and node
pub = rospy.Publisher('myFirstTopic', String, queue_size=10)
rospy.init_node('publisher', anonymous=True)

# Publish Message
pub.publish('Hello World')   # Publish Message
```

Cross-Platform

The main client side libraries of the ROS middleware are written in C++, Python, and Lisp. These libraries compile on multiple operating systems. More experimental client side libraries exist, which cover even more programming languages like C#, Java, or Lua. Another way to write a ROS node is to use rosbridge, an extension to the roscore that provides a websocket server for communication. The websocket protocol is a standardized web technology that enables a bi-directional communication between a server and a client, usually a web browser, which has been widely adopted (Fette and Melnikov, 2011). Thus, more platforms without a dedicated ROS client side library can communicate via ROS (Crick et al., 2011). This cross-platform and multi-language support is important for the VRFS, as connecting components, independent from the used programming language or operating system, is a key feature of the system.

Ease of Development

The use of the client side libraries does not require low-level knowledge of network protocols. Writing a publisher (see Listing 4.1) or a subscriber (see Listing 4.2) in Python does only require a few lines of code. In order to connect to a ROS network, no knowledge of the network's topology is necessary. Only a common set of topics and message types has to be available. This makes ROS ideal to connect external components without a huge development effort.

Predefined Message Types

ROS has a set of message types that include common primitive data types like integer, double, or string and more complex message types specific for robotic applications such as messages for position and orientation (pose), image data, or audio data. Some of these specific messages will be used in the VRFS, for example for the transformation of coordinate frames, for the tracking system, or for video and audio streaming. In addition, the definition of custom message types is possible as well.

Listing 4.2: A simple ROS subscriber written in Python.

```
import rospy
from std_msgs.msg import String

# Define a callback function
def callback(data):
    print(data.data)

# Init node and subscribe to a topic
rospy.init_node('listener', anonymous=True)
rospy.Subscriber("myFirstTopic", String, callback)

# Run Node and check for incoming messages
rospy.spin()
```

Recording

All communication within the ROS framework can be recorded and saved in so-called rosbags. These rosbag files can be replayed, i.e. all recorded information is published in the ROS network. This is helpful as it allows to save simulated flights and all other recorded data and analyze the data after a simulator session. In addition, this functionality is helpful for developers: With the simulated data output of a simulator session, which is available in one file, new components can be developed and tested without having access to the simulator itself.

Recording data is a use case that is described in Table 3.1 (on page 22) as 5.2 *Record Simulation Data*.

Documentation

The Robot Operating System has a big, active, and constantly growing user community. This user base contributes to an extensive documentation, including code examples and a "question and answer" section (Cousins et al., 2010). In addition, the source code of the core ROS system is available and published under the standard three-clause Berkeley Software Distribution (BSD) license[1]. This makes the software rather future-proof and likely to be further developed as it is independent from proprietary software companies. The source code can even be altered to add specific features that are not included in ROS.

Graphical User Interface

A specific framework for developing graphical user interfaces for ROS is called RQT. Although ROS can be fully controlled with command line tools, these graphical interfaces are helpful for debugging and monitoring a ROS network. Many tools are

[1]See https://opensource.org/licenses/BSD-3-Clause

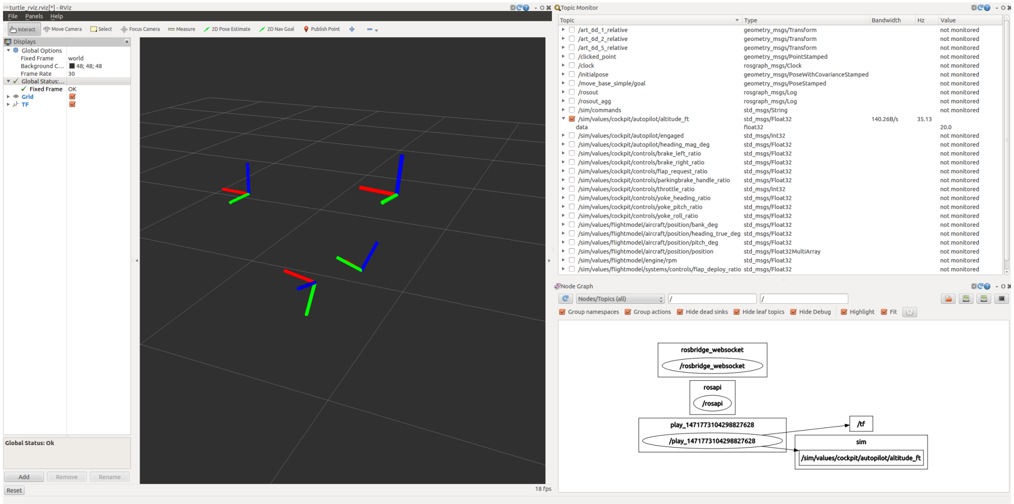


Figure 4.2.: Monitoring the ROS network with RQT tools. The figure shows the ROS Visualization Tool (left), the topic monitor (top right), and a graph representation of the network topology (bottom right).

already included like a simple topic monitor, the ROS Visualization Tool (RVIZ) that can visualize spatial data like coordinate systems or point clouds, and rqt_graph for visualizing the network topology as a graph. These tools can be used as standalone applications or as plug-ins in the RQT application. Figure 4.2 shows the RQT application with RVIZ, the topic monitor and rqt_graph.

4.1.2. ROS and the Architecture of the VRFS

Most components in the VRFS communicate via ROS, i.e. they subscribe and publish topics. In order to establish this communication, a common nomenclature of topic names has to be introduced. For this research, a hierarchical approach to name each topic was chosen. A topic name is classified by the node that is most relevant for the topic and then followed by more detailed categorization, separated by a slash (/) sign (Mamessier, Dreyer, and Oberhauser, 2015).

Figure 4.3 shows the categories and some examples of topic names that are used in the VRFS. Most topics stem from the the flight simulation node (sim/), as over 400 different types of telemetry data and topics related to the cockpit are available. Other topics relate to the tracking system (tracking/), to texture streaming and video cameras (texture/), audio messages (audio/), as well as physiological data to monitor the pilot (physio/).

In the following sections, the purpose of each part of the VRFS will be described including the connection to ROS.

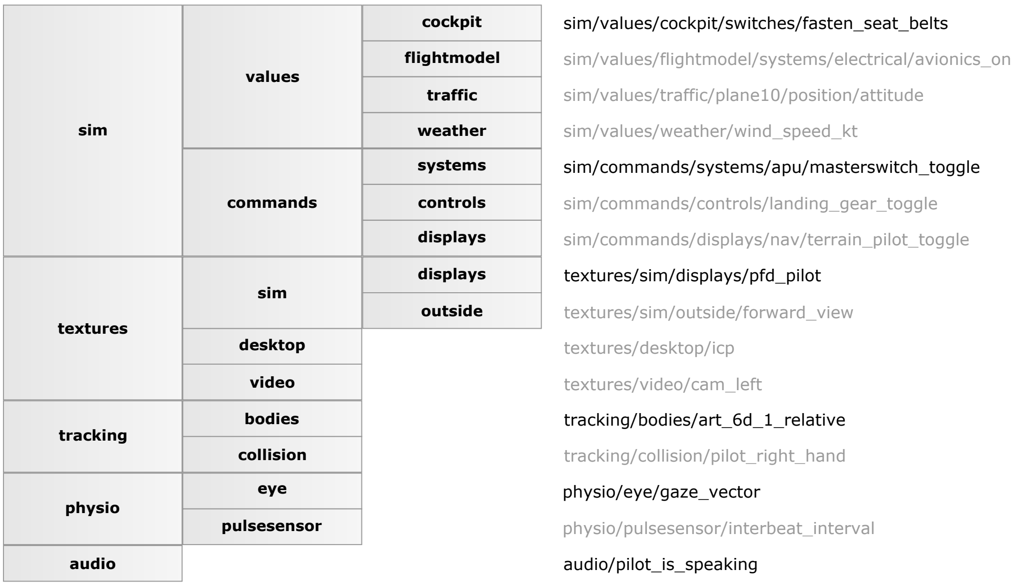


Figure 4.3.: Nomenclature of the ROS topics in the Virtual Reality Flight Simulator. The figure includes example topics for every presented high level ROS topic.

4.2. The Flight Simulation Software

X-Plane was chosen as the flight simulation software for the VRFS. As shown in Section 3.2 (page 21), it offers a high level of physical and functional fidelity while it still can be heavily modified with the help of plug-ins. X-Plane renders an outside visual according to the position of the simulated aircraft, time of day, and weather parameters. The position of the aircraft is re-calculated in each simulation loop based on an atmospheric simulation, the aircraft model, and the user input. On top of this outside visual, either a Two-dimensional (2D) panel with gauges, switches, and displays or a three-dimensional cockpit can be visualized. This three-dimensional cockpit can contain animated buttons and a dynamic texture, the so-called *panel texture*. Similar to the two-dimensional panel, this panel texture contains displays, gauges, or indicators. The panel texture is not visible but is mapped to the cockpit's 3D geometry. Figure 4.4 shows the main components of the X-Plane simulation as typically used in the VRFS, including the used aircraft model and additional plug-ins.

A core concept for X-Plane plug-in developers are the so-called *datarefs*. With these data references, a plug-in developer can gain read and write access to all major variables that are used by the flight simulation. This enables manipulating the behavior of many aspects of the simulation during run-time[2]. Datarefs can point to variables with primitive data types like float, integer or string; and can point to arrays with these data types. Some are read only whereas others offer bi-directional access, in other words read and write. Another way to control the simulation are commands. In contrast to

[2] A list of all available datarefs can be found here: http://www.xsquawkbox.net/xpsdk/docs/DataRefs.html

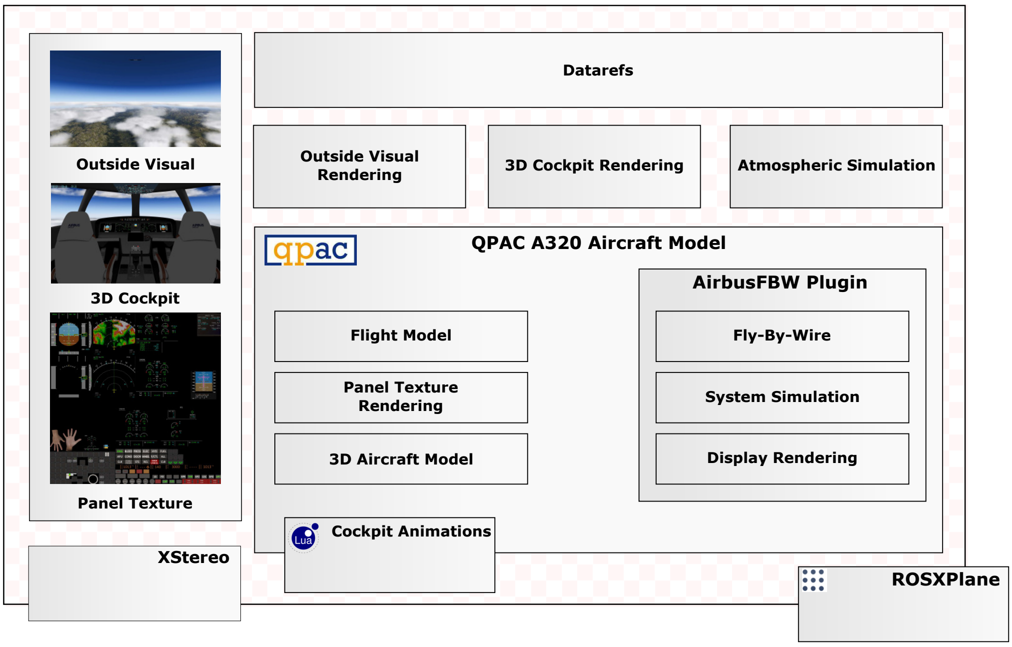


Figure 4.4.: The components and plug-ins used in X-Plane. The Figure shows the accessible OpenGL frame buffers (left top), functionality provided by X-Plane (top), and the main components of the QPAC aircraft model with attached Lua scripts (left center). Two components connect X-Plane to other components: XStereo for visualization and ROSXPlane for connecting to the Robot Operating System.

datarefs, commands are single actions that trigger additional changes in the flight simulation. In a plug-in, available commands can be triggered and custom commands and their specific actions can be created.

With the help of plug-ins, it is possible to control certain aspects of the rendering of the simulation. For this purpose, different frame buffers, like the outside visual or the panel texture, can be accessed. In this way, 3D objects can be dynamically drawn into the outside virtual world or 2D content can be added to the panel texture.

4.2.1. Aircraft Model

X-Plane already includes some aircraft models like general aviation aircraft, military jets or commercial airliners. Besides this selection, many further aircraft can be purchased or downloaded for free from several sources. An aircraft model usually includes a flight model, a system simulation, a 3D model of the aircraft, and a 2D or 3D cockpit. X-Plane features software that can be used to create custom aircraft. With a software called Plane-Maker, the aerodynamic and system characteristics of an aircraft can be modeled. Plane-Maker also includes a tool to create 2D cockpits with gauges, displays or control elements. Another tool that is included in X-Plane is the Airfoil-Maker, a tool to create airfoils. Yet, a large number of predefined airfoils is already included in X-Plane.

The quality of the available aircraft models varies. Some models only have a simplified flight model with a 2D cockpit, whereas others feature a high-fidelity flight model with detailed 3D cockpits. Some of these high-fidelity models, usually commercial software, expand the built-in X-Plane features by using custom plug-ins. With these custom plug-ins, additional systems that are not simulated by the standard X-Plane simulation loop can be modeled as well as custom flight displays that are not available in Plane-Maker.

For this research, the QualityPark AviationCenter (QPAC) A320 aircraft model is used for the standard setup. This aircraft resembles an Airbus A320, one of the most common aircraft models in commercial aviation. The strengths of this flight model are a sophisticated representation of the so-called Fly-by-wire (FBW) system, a custom Flight Management System (FMS), and accurate high resolution displays. All these components are managed by a custom plug-in, the *AirbusFBW* plug-in. Additional datarefs are created by this plug-in in order to reflect the custom systems that come with the aircraft model. The high resolution displays are rendered directly to the panel texture using OpenGL calls.

4.2.2. The Virtual Cockpit

The Cockpit Geometry

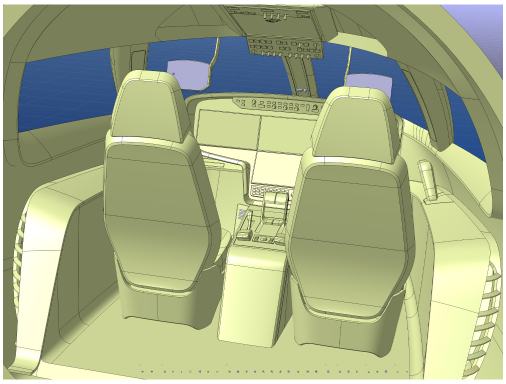
A generic cockpit, originally designed by Mühlbauer (2010) (see Figure 4.5b), is used in the standard setup of the VRFS. In contrast to an official Airbus cockpit, a generic cockpit enables the easy integration and testing of new cockpit components without being restricted by specific legacy cockpit aspects like screen formats or panel positions. In addition, no confidentiality restrictions apply for this generic cockpit, which is important in particular when sharing the data with academic or industrial partners and publishing research.

The cockpit was initially designed for a small business jet as part of the All Condition Operations and Innovative Cockpit Infrastructure (ALICIA) project (ALICIA, 2014). For the use in combination with the QPAC aircraft model, the cockpit had to be adapted to an Airbus A320 aircraft. Therefore, the Computer-aided Design (CAD) data of the cockpit, which is available in the CATIA V5 file format (see Figure 4.5a), was scaled and a new hull was designed. The central pedestal was modified to comply with a hardware thrust lever that was purchased for the VRFS. Due to the increased space, another wide screen display was added, as illustrated in Figure 4.5c.

The Geometry Import Pipeline

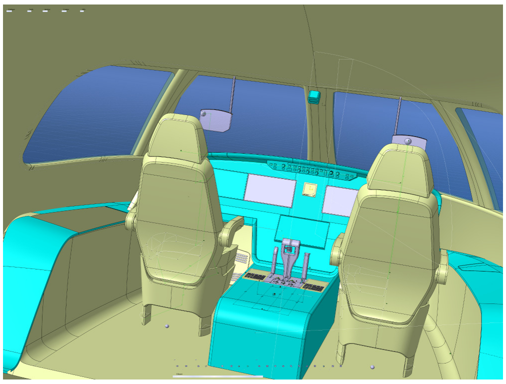
The cockpit geometry that was designed in CATIA V5 had to be converted into an X-Plane specific file format - the *OBJ8* format[3]. These text files describe the 3D geometry with the position of each vertex, the faces, the texture mapping, and the normals

[3]See `http://developer.x-plane.com/?article=obj8-file-format-specification` for a specification of the OBJ8 format.



(a) ALICIA CAD model

(b) ALICIA rendering (Mühlbauer, 2010)



(c) ACROSS CAD model

(d) ACROSS rendering

Figure 4.5.: The CAD models and the rendering of the ALICIA (top) and the ACROSS (bottom) cockpit. The ACROSS cockpit was enlarged to match an Airbus A320 aircraft. This makes an additional flight display in the ACROSS cockpit possible, compared with the ALICIA cockpit.

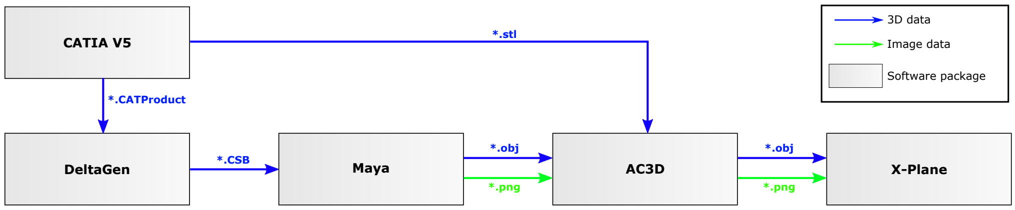


Figure 4.6.: The pipeline for integrating cockpit geometry available in CATIA V5 into X-Plane.

of each face. In addition, X-Plane specific information like animations, lighting, and manipulators, i.e. 3D objects that can be manipulated with mouse inputs, are stored in this file. OBJ8 files are used by X-Plane for scenery objects, cockpit objects, and other aircraft objects. With a plug-in for the 3D design application AC3D OBJ8 files for X-Plane can be created.

CATIA V5 geometry can be imported into AC3D by using the STereoLithography (STL) file format. For this file format, the parametric geometry data of CATIA has to be tessellated into a set of triangles. The quality of the tessellation of the CATIA exporter is limited, the model does not feature any materials or textures, and the product structure is lost. This is why two programs are added to the pipeline for tessellating and texturing the model, as shown in Figure 4.6. First, DeltaGEN, a software for high quality real-time rendering of 3D models, is used for tessellation as it features a sophisticated CATIA V5 importer. After the tessellation, the model is transferred to Autodesk Maya, a 3D computer graphics software. In Maya, textures, materials, and lights are added to the model and rendered to static textures. The textured model, including the respective graphic files, is then exported to AC3D (Oberhauser, Dreyer, and Kovačič, 2015). This import pipeline fulfills use case 2.1 *Import Cockpit Geometry*, as shown in Table 3.1 (page 22).

Adding Animations

The X-Plane plug-in for AC3D has a graphical user interface to add animations, which are connected to X-Plane *datarefs*. Four different types of basic animations exist:

- Translation,
- rotation,
- show and,
- hide.

With this set of primitive animations and their combinations, all animations in the cockpit can be implemented. For translations, a line with a specific length has to be added to the model. Along this line, the object will translate. For rotation animations, an axis has to be set about which the object rotates. For every animation, key frames have to be set that connect a dataref value to the progress of an animation. Between the key frames, a linear interpolation will be calculated. An unlimited number of key frames can be added. Figure 4.7 shows an object that is animated in AC3D. A rotation axis is placed in the center of a rotary knob. This rotation animation is connected to an

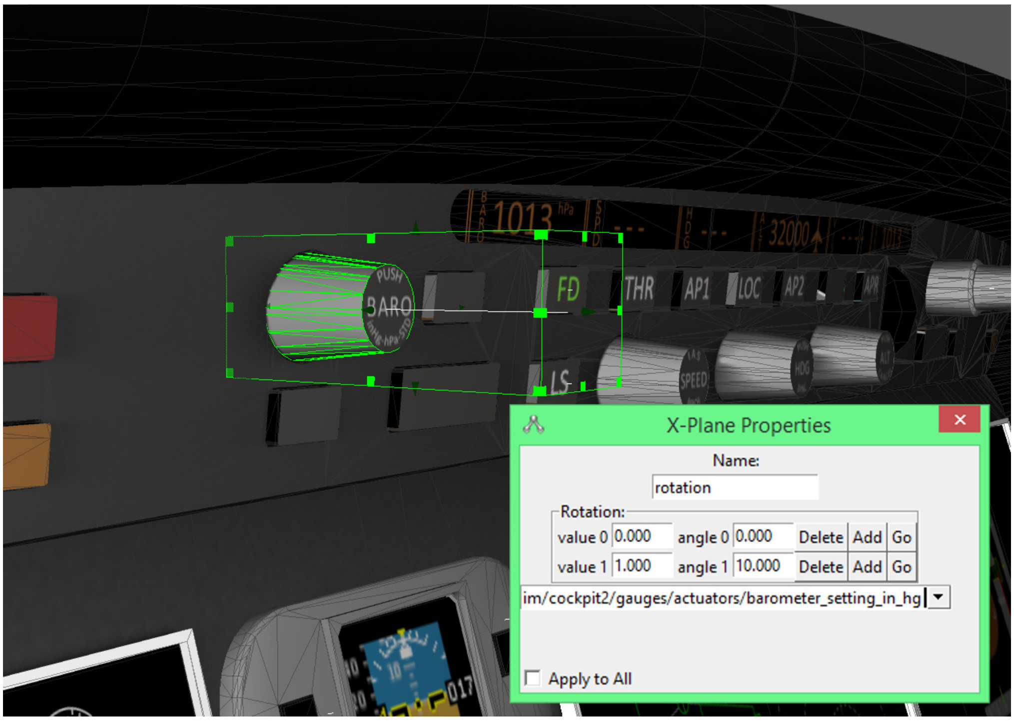


Figure 4.7.: Adding animations to 3D objects in AC3D. The figure shows a rotation animation of a rotary knob. The X-Plane Properties window shows the key frames for this animation and the connected dataref.

X-Plane dataref, the key frame setup shown in Figure 4.7 results in a rotation of ten degrees if the connected dataref value is changed by one.

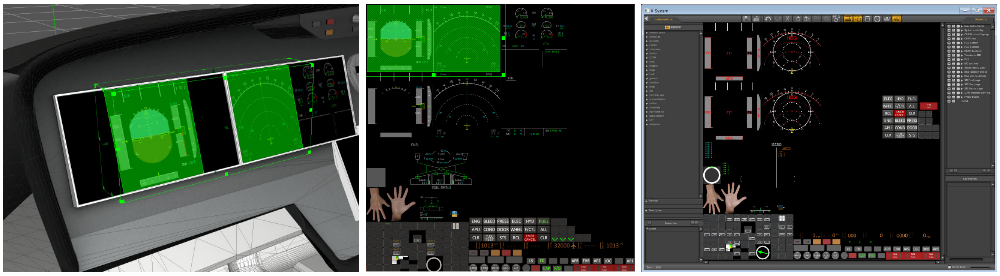
With this approach, animations can be connected to every movable object in the cockpit, a use case (2.2 *Animate Objects*) described in Table 3.1 (page 22). Rotations have been added to rotary knobs and levers like the throttle levers, the gear lever, the flaps lever, and the speed brakes lever. Two rotations have been attached to the flight stick in order to reflect the joystick's pitch and roll movement. For every push-button in the cockpit, a small translation animation (push and release) has been added as well.

All animations of 3D geometry in the cockpit have been implemented with this method. All other animations like indicator lights or displays are implemented with the help of the panel texture.

The Panel Texture

The panel texture is an integral part of every 3D cockpit. X-Plane already comes with a tool to create and edit the panel texture - the Plane Maker[4]. This tool (see Figure

[4]See `http://developer.x-plane.com/docs/aircraft/plane-maker-manual/` for the official manual of Plane Maker including editing the panel texture.



(a) Texture mapping in AC3D (b) The panel texture (c) X-Plane panel editor

Figure 4.8.: The panel texture and the 3D mapping.

4.8c) features a set of predefined instruments that can be used to create a cockpit panel. Besides, in the current X-Plane version twelve generic instruments are available. With these generic instruments, custom images can be used to create gauges, map displays, and indicators or to show numbers. For example the indicators on a push-button—as shown in Figure 4.9—can be implemented by providing images for all button indications and linking the images to specific dataref values. In particular the Overhead Panel (OHP) of the proposed generic cockpit has numerous push-buttons. The QPAC flight model already comes with some overhead panel indicators in the 2D panel that can be used for the 3D cockpit; all others have to be created using the described implementation.

The panel texture, including the displays that are drawn by the QPAC A320 plug-in (see Figure 4.8b), can be mapped to the 3D geometry in AC3D. For this purpose, the surfaces are selected in AC3D (see Figure 4.8a) and mapped to a part of the texture. This process is called UV-mapping and is common for adding textures to 3D objects. In the cockpit, the displays, button indications, and warning lights are mapped to the 3D cockpit from the panel texture.

Mapping the dynamic panel texture to the 3D geometry is part of the use case 2.3 *Assign Dynamic Texture* described in Table 3.1 on page 22.

4.2.3. ROS Integration

For connecting X-Plane to the ROS network, a plug-in called ROSXPlane was developed in the scope of this research. This plug-in, written in C++, uses *roscpp*, a ROS client side library for C++. This plug-in allows access to X-Plane datarefs and commands through

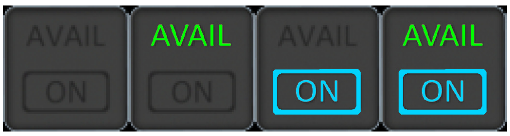


Figure 4.9.: Indications on a push-button.

Listing 4.3: Example of the datarefs.ini file.

```
"s","xref","rtype","rperm"
"sim/values/weather/metar","sim/weather/metar","std_msgs/String","both"
"(...)/aircraft/position/pitch_deg","(...)/position/theta","std_msgs/Float32","both"
"(...)/icing/engine_anti_ice","(...)/switches/anti_ice_engine_air","std_msgs/
    Float32MultiArray","read"
```

the common VRFS nomenclature presented in Section 4.1.2. Accessing simulation data as provided by this plug-in is a use case described in Table 3.1 (page 22) as 5.1 *Access Simulation Data*. In addition, the plug-in makes bi-directional streaming of textures possible and is essential for many virtual reality aspects of the VRFS.

In the following sections, the implementation of the ROSXPlane plug-in will be described with an emphasis on telemetry data access, commands, and texture streaming.

Telemetry Data

The X-Plane datarefs offer comprehensive access to internal variables of the simulation. Yet, it is not beneficial to provide direct access to these datarefs via ROS as only a fraction of these datarefs is relevant for integrating HMI components or for monitoring a simulated flight. Another issue arises from the list of datarefs itself: The number of datarefs and their nomenclature have grown throughout time. This resulted in inconsistencies, duplications, and legacy entries. Therefore, only access to selected datarefs with a new and consistent nomenclature that is inspired by X-Plane will be provided (Mamessier, Dreyer, and Oberhauser, 2015). The nomenclature was presented in Section 4.1.2 (page 39).

For a selection of relevant datarefs in this research, the specification of the *CANAerospace* protocol (Stock and Deas, 2009) is used as a baseline. The basic 124 data points of this protocol cover relevant telemetry data like airspeed, attitude, and autopilot states. This list was further extended and data points for cockpit elements, traffic information, and weather information were added. This resulted in 480 datarefs that are accessible via

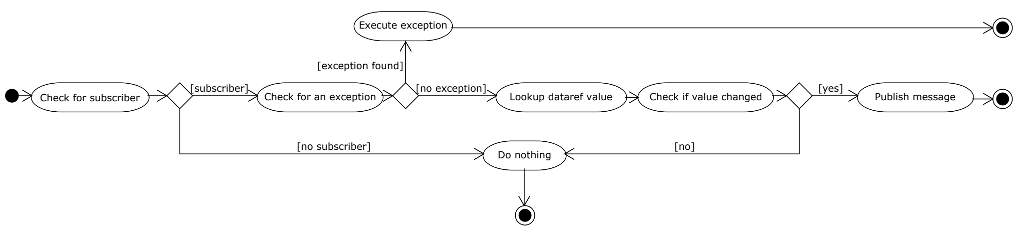


Figure 4.10.: The UML activity diagram for outgoing ROS messages. Several checks are performed before a message is published: (1) Is there a subscriber? (2) Is there an exception function? (3) Did the value change since the last publication?

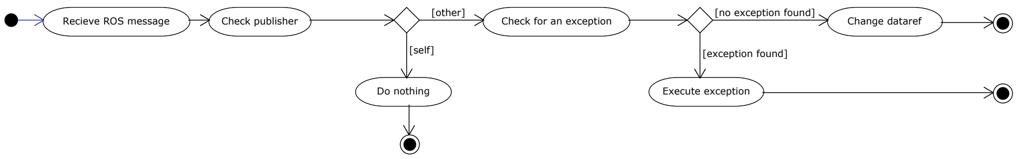


Figure 4.11.: The UML activity diagram for incoming ROS messages. Two checks are performed before an incoming message is applied. First, a check for a self-subscription and second, a check for an exception function.

ROS so far. This number is constantly growing in order to reflect technological advances and project needs.

A Comma-separated Values (CSV) file (see Listing 4.3) with the list of datarefs, the respective ROS topic, the data type, and the permission is created and centrally maintained. This text file is read by the ROSXPlane plug-in and stored in a lookup table for read access, write access or both. The topics that have the permission `read` or `both`, i.e. `read` and `write`, are published in the ROS network. Yet, as shown in the activity diagram in Figure 4.10, two checks are performed before a topic is published. First, the application checks if there is a subscriber for this topic in the ROS network; second, the topic is only published if the value has changed. With the help of these two mechanisms unnecessary network traffic and computing time can be avoided.

Topics that have the permission `both` or `write` have a subscriber callback that handles incoming messages on a specific topic. In case the permission is `both`, it is possible that the incoming message was published by ROSXPlane itself. This would lead to a self-subscription loop, which can be problematic. Therefore, a check is performed in which the name of the publishing node is compared with the node name of ROSXPlane. This is illustrated in the UML activity diagram in Figure 4.11.

In most cases, receiving ROS messages and publishing ROS messages from X-Plane requires a simple lookup operation and reading or writing one dataref value. In some cases, additional operations are necessary. For example, if a new aircraft position is published by an external node, multiple steps including disabling the flight model, repositioning the aircraft, loading new scenery, and re-enabling the flight model have to be performed in X-Plane to place the aircraft according to the new coordinates[5]. Thus, `read` and `write` exceptions have been introduced. These exceptions are C++ functions that are hard coded in ROSXPlane and executed if a topic or dataref is accessed that needs special attention.

Commands

Commands in X-Plane are single events, which trigger one or multiple actions, e.g. dataref changes. Over 1100 commands are available in X-Plane 10. Similar to the list of datarefs, this list contains legacy entries, duplicates, and irrelevant commands. For the VRFS, the list was condensed to 350 commands, mainly for operating the

[5]A complete explanation of the necessary steps is provided here: http://www.xsquawkbox.net/xpsdk/mediawiki/MovingThePlane

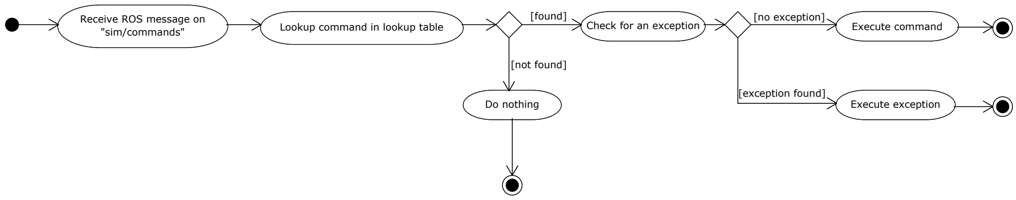


Figure 4.12.: The UML activity diagram for incoming ROS command messages. If the command is present in the lookup table and if there is no exception, the corresponding command is triggered in X-Plane.

cockpit elements such as push-buttons, rotary knobs, and levers. In contrast to the telemetry data, for these 350 commands a single topic is introduced. This topic named *sim/commands/* is of data type string. This string holds the rest of the name that was defined in the nomenclature presented in Section 4.1.2 (page 39).

For instance, *sim/commands/systems/signs/nosmoking_switch* becomes the message *systems/signs/nosmoking_switch* in the *sim/commands/* topic. As commands are not issued continuously, this approach is feasible and limits the number of topics that are available in the system. In addition, a ROS node can subscribe to the single *sim/commands/* topic and still receive all 350 possible commands.

ROSXPlane does only receive commands, no commands are issued from this node. Figure 4.12 shows an activity diagram for an incoming message on the *sim/commands* topic. First, the plug-in checks whether the command name is present in the lookup table, which is populated from a CSV file as shown in Listing 4.4. If yes, another check is performed whether there are any exceptions for this command. If not, the command is issued according to the X-Plane command name in the lookup table.

Texture Streaming

The Robot Operating System offers a package to transport image data. With the *image_transport*[6] and the *compressed_image_transport* packages, images can be published and subscribed either in a compressed or in a raw format. The ROSXPlane plug-in uses the *image_transport* functionality to stream textures from the simulation to external nodes and for streaming textures into the 3D cockpit.

Listing 4.4: Example of the commands.ini file.

```
"s","x"
"sim/commands/autopilot/altmode_toggle","Airbus/FCU/Autopilot/AltMode_toggle"
"http://vrfs.airbus.com/fm#sim/commands/controls/autoland_toggle","Airbus/FCU/Copilot/
    Autoland_toggle"
"http://vrfs.airbus.com/fm#sim/commands/systems/legacy/calls/fwd_toggle","Airbus/OHP/
    Calls/Fwd_toggle"
```

[6]See http://wiki.ros.org/image_transport for more details.

Table 4.1.: The X-Plane drawing phases (Barbour and Supnik, 2005).

Drawing phase	Number	Description
xplm_Phase_FirstScene	0	This is the earliest point at which you can draw in 3-d.
xplm_Phase_Terrain	5	Drawing of land and water.
xplm_Phase_Airports	10	Drawing runways and other airport detail.
xplm_Phase_Vectors	15	Drawing roads, trails, trains, etc.
xplm_Phase_Objects	20	3-d objects (houses, smokestacks, etc.
xplm_Phase_Airplanes	25	3-d objects (houses, smokestacks, etc.
xplm_Phase_LastScene	30	This is the last point at which you can draw in 3-d.
xplm_Phase_FirstCockpit	35	This is the first phase where you can draw in 2-d.
xplm_Phase_Panel	40	The non-moving parts of the aircraft panel.
xplm_Phase_Gauges	45	The moving parts of the aircraft panel.
xplm_Phase_Window	50	Floating windows from plug-ins.
xplm_Phase_LastCockpit	55	The last chance to draw in 2d.

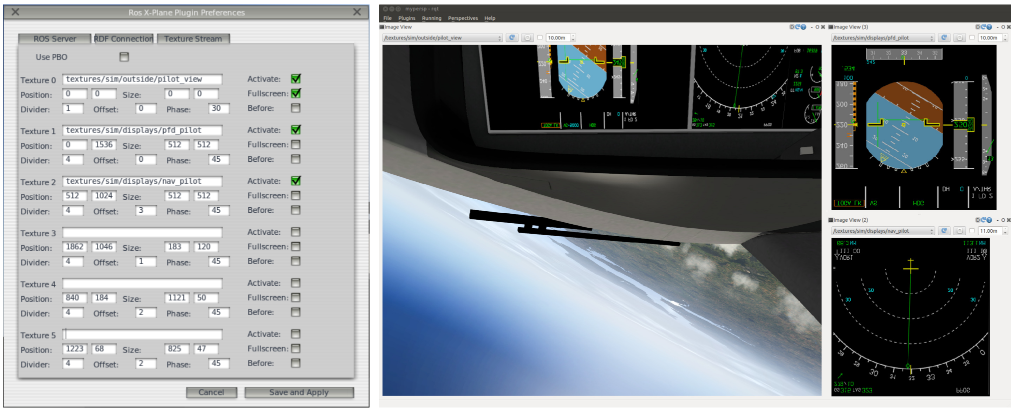
X-Plane renders a scene step by step: First, the outside visual is rendered, then the 3D cockpit, and finally Graphical User Interface (GUI) elements. After each rendering step, the results of the rendering are stored in a so-called frame buffer, a rectangular array of pixels, which can be displayed on a computer screen (Woo et al., 1999, p. 144). The X-Plane plug-in SDK offers the possibility to access these OpenGL frame buffers at different points in time during this drawing cycle. Table 4.1 shows the drawing phases for which frame buffer access is possible (Barbour and Supnik, 2005).

The X-Plane OpenGL frame buffers that are of interest for external nodes are the outside visual without the cockpit (phase 30), the panel texture (phase 45), and the cockpit view (phase 55). In order to stream the content of these frame buffers, the buffer has to be read and transferred from the Graphics Processing Unit (GPU) controlled memory to Central Processing Unit (CPU) controlled memory. This can be achieved with the OpenGL call `glReadPixels()`. This operation has performance restrictions in particular when reading a large area of the frame buffer as the data has to be transferred to CPU controlled memory. By copying the current frame buffer into a so-called Pixel Buffer Object (PBO) and then calling `glReadPixels()`, this operation can, to some extent, be parallelized[7]. As soon as the pixel data is available in the CPU memory, the data can be published via ROS.

For configuring the texture streaming, ROSXPlane features a GUI for publishing image topics as shown in Figure 4.13a. Up to five different image topics can be published from within X-Plane. The drawing phase, the topic name, and the image clipping can be set in this GUI. A part of the panel texture can be selected and published via ROS in a dedicated topic, e.g. only the Primary Flight Display (PFD) can be published. In addition, a divider can be set to limit the publishing frame rate of the image topics to increase the performance of X-Plane. Figure 4.13b shows the output of the texture stream in RQT. Due to the mismatch of the OpenGL coordinate system and the coordinate system used in RQT, the textures from the simulation are displayed upside-down.

Up to three image topics can be streamed into the 3D cockpit. In a configuration file, the topic name, the position, and the size of the section on the panel texture can be

[7]See `http://www.songho.ca/opengl/gl_pbo.html` for details



(a) The texture streaming GUI (b) RQT windows of the texture streams

Figure 4.13.: Texture streaming of the flight simulation.

set. An image topic is subscribed via ROS and written to the frame buffer at the gauges drawing phase (number 45) using the OpenGL call `glTexImage2D()`. Figure 4.14a shows the panel texture with two external image streams. As the panel texture is mapped to the 3D cockpit geometry as described in section 4.2.2, the textures appear in the 3D view of the cockpit as shown in Figure 4.14b .

4.2.4. Scenarios

The Virtual Reality Flight Simulator will be used as an instrument for human factors studies. In order to be reliable, a psychological test has to be consistent, i.e. every measurement should take place in the same controlled environment (G. Domino and M. L. Domino, 2006, p. 17). For a flight simulator, this implies the ability to recreate the same scenario for different pilots. X-Plane offers the possibility to set-up, save, and load scenarios in so-called situation files (Williams, 2012, p. 115). With this built-in functionality, only a limited set of datarefs like the aircraft's position, the weather settings, and the time of day can be be saved and restored. Yet, the amount of information on the aircraft systems and the state of cockpit elements, which is stored in these situation files, is insufficient for some scenarios.

Hence, an application that stores more telemetry data values was created in the scope of this research: The VRFS Scenario Manager. This program, written in Java, is connected to ROS and subscribes to all available telemetry data. Either all, or only a subset of telemetry data points can be stored in a JavaScript Object Notation (JSON) file. JSON is a human readable file format to store structured data (Crockford, 2006). If these files are loaded with the Scenario Manager, all saved values are published once and the initial state of the aircraft is set up by the ROSXPlane plug-in. The Scenario Manager can be used together with the built-in X-Plane situations.

A scenario can also include system malfunctions. In X-Plane, a feature to induce system malfunctions already exists. Yet, the available system malfunctions are limited and

(a) The panel texture (b) The 3D cockpit view

Figure 4.14.: Texture streaming of a webcam image and a remote desktop into the cockpit.

rather generic. Hence, the Scenario Manager offers the possibility to induce commands or to change telemetry data interactively through a graphical user interface via keyboard shortcuts. Compared with the X-Plane built-in functionality, more specific systems can be manipulated or deactivated with the ROS interface.

In this section, the implementation of two use cases has been presented. These use cases have been described in Table 3.1 on page 22: 3.1 *Load a Scenario* and 3.2 *Induce System Malfunction*.

4.3. Virtual Reality

Three distinct features are needed to extend the flight simulation with the 3D cockpit by VR capabilities: A tracking system to determine the user's head and hand position, a visualization of the user's viewpoint and his or her hands as well as possibilities to interact with the 3D cockpit (Dreyer, Oberhauser, and Bandow, 2014). In the following subsections, the implementation of these features, which create the virtual reality environment of the VRFS, will be described.

4.3.1. The Tracking System

With the help of the tracking system, the position and orientation, i.e. pose of the user's head, hand, fingers, and other objects has to be captured. This data is then transformed into a common coordinate system frame and published via ROS.

With the *ARTTrack 2* tracking system, which is typically used in combination with the VRFS, the pose of several bodies (x_{bn}, y_{bn}, z_{bn}), active or passive, can be monitored. The poses of these bodies refer to a coordinate system (see Figure 4.15) that is defined by

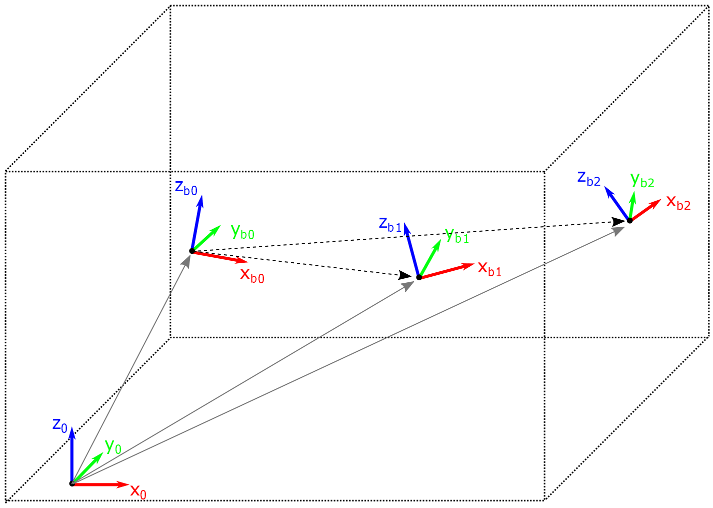


Figure 4.15.: Coordinate frames in the ART tracking system.

an initial calibration of the tracking system. During this calibration, a calibration target, which defines the coordinate origin, also called world coordinate system (x_0, y_0, z_0), is placed inside the tracking area. As this calibration has to be repeated from time to time, the coordinate frame of the tracking targets may change. Thus, a so-called reference target (x_{b0}, y_{b0}, z_{b0}) is placed inside the tracking area. The pose of all other ART targets is based on this reference target instead of the world coordinate system. This makes it independent of world coordinate system changes as a result of recalibration.

The transformation of the ART world coordinate frame to the relative coordinate frame is conducted in a ROS node called ROSART. This node was developed for the VRFS and other use cases and is written in C++. It receives data from the ART tracking system, transforms it into a common coordinate frame, and provides the data via ROS to other nodes.

In DTrack, the control software of the ART tracking system, tracking data can be sent via UDP. For standard Six Degrees of Freedom (6DOF) bodies, this output is structured as shown in 4.1:

$$[id\, qu][s_x\, s_y\, s_z\, \eta\, \theta\, \phi][b_0\, b_1\, b_2\, b_3\, b_4\, b_5\, b_6\, b_7\, b_8] \tag{4.1}$$

id is a unique identification number of the target that is set in DTrack. s_x, s_y, and s_z define the translation and η, θ, and ϕ the rotation of the bodies relative to the world coordinate frame (x_0, y_0, z_0). Alternatively, a rotation matrix R consisting of $[b_0...b_8]$, as shown in Equation 4.2 is provided. Due to problems that can stem from different definitions of axes and angles, it is recommended to use the rotation matrix for further computations (Advanced Realtime Tracking GmbH, 2016).

$$R = \begin{pmatrix} b_0 & b_1 & b_3 \\ b_3 & b_4 & b_5 \\ b_6 & b_7 & b_8 \end{pmatrix} \tag{4.2}$$

ROS includes the ROS Transform Library (TF), which was developed to manage coordinate frames and transformation of spatial data (Foote, 2013). All coordinate frames and transformations are represented in a graph with the coordinate frames as the nodes and the transformations as the edges. Transformations between all connected coordinate frames can be calculated with this library.

The ROSART node receives the UDP messages from DTrack and converts the pose information of these UDP datagrams to the TF graph, which is also published via ROS. This enables visualization in RVIZ and any transformation operation that might be necessary. TF provides a transform message type which contains information about translation $[x, y, z]$ and rotation, given in quaternions $[x, y, z, w]$; in other words, a pose message. Two topics are published for each ART body: A transform message based on the ART world coordinate frame (x_0, y_0, z_0) and a transform message relative to the reference coordinate frame (x_{b0}, y_{b0}, z_{b0}). The pose based on the reference coordinate system is calculated using the `lookupTransform()` method provided by the TF library. ROSART considers the ART body with the ID 0 as the reference target.

4.3.2. Pilot's View Visualization

The head pose relative to the reference coordinate frame (x_{b0}, y_{b0}, z_{b0}) is subscribed by the ROSXPlane node. An event is triggered when a new message arrives and the information is used to control the camera of X-Plane. All incoming transform messages are converted to the X-Plane coordinate system frame by the ROSXPlane node. For the rotation, the incoming quaternions, which are defined as shown in Equation 4.3, are converted to roll, pitch, and yaw values. Quaternions can be transformed to roll, pitch, and yaw values using Equation 4.4 (Diebel, 2006).

$$q = [q_0\, q_1\, q_2\, q_3]^T \tag{4.3}$$

$$\begin{pmatrix} \text{roll} \\ \text{pitch} \\ \text{yaw} \end{pmatrix} = \begin{pmatrix} \operatorname{atan2}\left(2(q_0q_1 + q_2q_3), 1 - 2(q_1^2 + q_2^2)\right) \\ \arcsin\left(2(q_0q_2 - q_3q_1)\right) \\ \operatorname{atan2}\left(2(q_0q_3 + q_1q_2)\right), \left(1 - 2(q_2^2 + q_3^2)\right) \end{pmatrix} \tag{4.4}$$

The incoming translation has to be transformed as well. The main camera and all tracked objects in the cockpit (e.g. hands, fingers, or checklists) have their origin in the center of gravity (x_{cg}, y_{cg}, z_{cg}) of the simulated aircraft. The incoming transform message from ROSART has a translation relative to the reference coordinate frame (x_{b0}, y_{b0}, z_{b0}). Hence, a linear translation as shown in Equation 4.5 has to be applied to the incoming translation data. This translation reflects the position of the reference

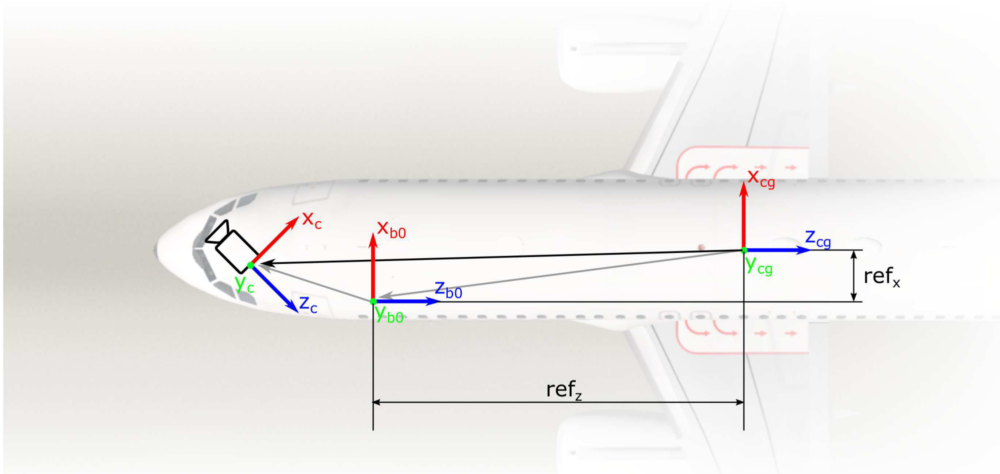


Figure 4.16.: The coordinate system frames of X-Plane.

target (x_{b0}, y_{b0}, z_{b0}) relative to the center of gravity (x_{cg}, y_{cg}, z_{cg}) of the simulated aircraft as shown in Figure 4.16. The center of gravity of the simulated aircraft is a static value. It does not change during flight, e.g. due to changes in the fuel system. Yet, the translation has to be readjusted if a different X-Plane flight model is used.

Listing 4.5 shows the implementation of the coordinate transformation, both rotation and translation, in ROSXPlane. This function is called for every incoming transform message. It includes a transformation from the incoming quaternion rotation to roll, pitch, and yaw and it adds the necessary translation from the center of gravity.

$$\begin{pmatrix} {}^{cg}x_c \\ {}^{cg}y_c \\ {}^{cg}z_c \end{pmatrix} = \begin{pmatrix} {}^{b0}x_c \\ {}^{b0}y_c \\ {}^{b0}z_c \end{pmatrix} + \begin{pmatrix} ref_x \\ ref_y \\ ref_z \end{pmatrix} \tag{4.5}$$

The mono image that is provided by X-Plane can be cloned by the SX60's controller for the output to the used device NVIS SX60 HMD. This results in a monoscopic visualization. An important aspect of every virtual reality environment is the perception of depth. For monoscopic or flat images, so-called pictorial cues like shadows, known object sizes, or linear perspective help to determine the position of an object in three-dimensional space. In addition, parallax effects, i.e. relative movements of objects due to head translations, are another depth cue in monoscopic visualizations (Reichelt et al., 2010). Sutherland (1968) describes this as the *kinetic depth effect*: "Psychologists have long known that moving perspective images appear strikingly three-dimensional even without stereo presentation" (p. 757).

For a better 3D depth perception, in particular for close objects, stereoscopic vision is important. In order to achieve this, the scene has to be rendered from two different positions for each eye (Southard, 1993). Aslandere (2013) presented a method for adding stereoscopic vision to X-Plane, based on two instances of the simulation on two individual computers. Yet, this implementation suffers from delays and synchronization

Listing 4.5: Quaternion transformation in ROSXPlane

```
geometry_msgs::Twist ROSNetWorkManager::transformToXPlane(const geometry_msgs::
    Transform data)
{
  geometry_msgs::Twist coordX;
  XPLMDataRef gx,gy, gz;

  // Init Quaternion and transform to matrix
  tf::Quaternion q (data.rotation.x,data.rotation.y,data.rotation.z, data.rotation.w);
  tf::Matrix3x3 m (q);

  // Rotation Matrix to Quaternion
  m.getRPY (coordX.angular.x,coordX.angular.y,coordX.angular.z);

  // Get reference datarefs
  gx = XPLMFindDataRef ("sim/graphics/view/referenceX");
  gy = XPLMFindDataRef ("sim/graphics/view/referenceY");
  gz = XPLMFindDataRef ("sim/graphics/view/referenceZ");

  // Coordinates rad to deg
  coordX.angular.x =      coordX.angular.x * 180/M_PI;
  coordX.angular.y = -1.0*coordX.angular.y * 180/M_PI;
  coordX.angular.z = -1.0*coordX.angular.z * 180/M_PI;

  // Apply linear transformation
  coordX.linear.x = -1.0*(data.translation.y+XPLMGetDataf(gx));
  coordX.linear.y =      data.translation.z+XPLMGetDataf(gy);
  coordX.linear.z = -1.0*(data.translation.x+XPLMGetDataf(gz));

  return coordX; //return X-Plane rotation and translation
}
```

issues between the two eyes. Thus, a stereoscopy implementation on a single X-Plane instance is necessary and was implemented: A plug-in for X-Plane called XStereo was developed in C++ as part of this research. This plug-in uses the ability to access X-Plane's OpenGL frame buffers. The scene is rendered from one camera position and saved to a Pixel Buffer Object (PBO). Then, the camera perspective is changed slightly to reflect the human eye distance. The scene is then rendered again and saved to another PBO. Both PBOs are rendered side-by-side as shown in Figure 4.17b. This output has to be allocated to two different screen outputs and can be sent to the NVIS SX60 HMD. All other head-mounted displays that support side-by-side stereoscopic input or separate dual head input can make use of this plug-in.

The presented implementation has a negative impact on the frame rate, as the complete scene has to be rendered twice. A more low-level access to X-Plane would be necessary to improve this. Yet, with the level of access that is offered by the X-Plane plug-in SDK, this is not possible at the moment. Another issue is simulator sickness. Häkinnen et al. (2006) found that, although the level of immersion is higher, stereoscopic vision abets simulator sickness symptoms.

With the presented coordinate transformations and the processing in the ROSXPlane plug-in, the user wearing the head-mounted display is placed inside the aircraft. He or she can look and move around in a virtual environment. As the virtual cockpit is functional, the user is able to monitor indicators and displays as described in use case 1.5 *Use a Flight Display*, presented in Table 3.1 on page 22. Yet, at this point, no

(a) Head tracking in X-Plane (b) Stereo rendering in X-Plane with XStereo

Figure 4.17.: The visualization in X-Plane.

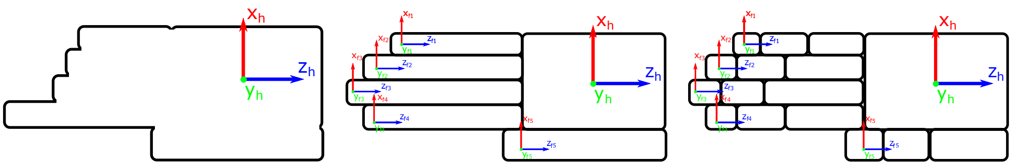
interaction with the cockpit is possible—a feature that will be described in the next section.

4.3.3. Interaction

In the last sections, the visualization of the 3D cockpit in a head-mounted display was described. Yet, in order to be able to fly the aircraft, interaction with this cockpit, e.g. with buttons, levers, or rotary switches has to be possible. Different types of interaction have been implemented to use the virtual cockpit. Some of these involve additional hardware components whereas others are fully virtual. All of these techniques have one thing in common: They rely on a finger or hand tracking system.

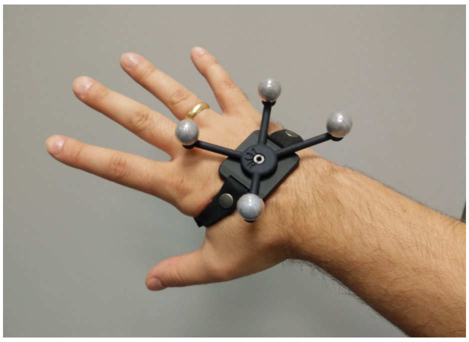
Using finger or hand tracking is an obvious and natural way to interact in a three-dimensional environment (Bowman, Kruijff, et al., 2005, p.106). In the ART Tracking system, either a passive hand tracking target (see Figure 4.19a) or an active finger tracking system with a sensor for each finger tip (see Figure 4.20a) can be used. The data gathered from these targets has to be fed into the virtual reality environment resulting in a 3D hand representation.

Three levels of fidelity for the hand representation have been implemented in the VRFS, as shown in Figure 4.18. If the interaction device is a single hand tracking target, a static hand geometry is used. The pose of the coordinate system frame of the hand (x_h, y_h, z_h)



(a) Static hand (b) Simplified hand (c) Complete hand

Figure 4.18.: The different hand representations in the VRFS. The figure shows the movable objects and joints as well as the pose that is provided by the tracking system.


(a) ART hand target


(b) Static virtual hand

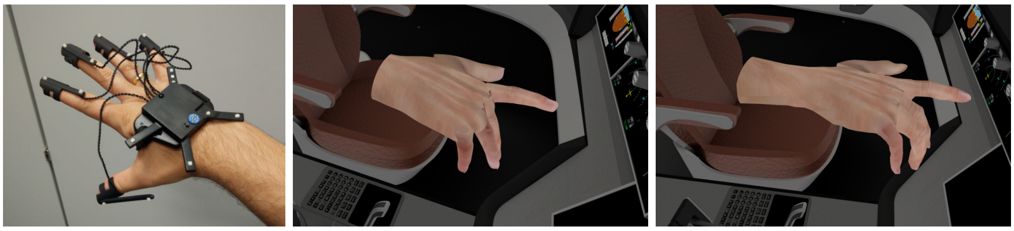
Figure 4.19.: The hand tracking target and the static 3D representation of the hand in the cockpit.

is given in world coordinate system by the ART tracking system and is transformed to the reference coordinate frame (x_{b0}, y_{b0}, z_{b0}) by ROSART. In the ROSXPlane plug-in, a transform message from the hand target will be processed in an exception function. In this exception, the incoming pose is transformed to the center of gravity coordinate frame (x_{cg}, y_{cg}, z_{cg}) of the simulated aircraft with the function presented in Listing 4.5 on page 56.

It is not possible to draw 3D objects inside the cockpit as there is no 3D drawing phase available after the cockpit is rendered (see Table 4.1 on page 50). Thus, the 3D hand representation cannot be drawn directly with the help of OpenGL calls. As a workaround for this limitation, the ability to animate objects can be used (Aslandere, 2013). In the X-Plane flight model, a 3D hand geometry including the hand and all fingers, usually already with a pointing index finger, has to be present at the center of gravity (x_{cg}, y_{cg}, z_{cg}). To this 3D object, three rotation animations, i.e. roll, pitch, yaw, and three translation animations, i.e. x, y, z, are attached in AC3D. This results in a static hand representation that translates and rotates according to the hand tracking target. As shown in Figure 4.19, this hand geometry does not reflect movements of single fingers. Yet, it is sufficient to interact with the cockpit to some degree, e.g. operating push-buttons or hardware elements.

If a finger tracking system is available, a more sophisticated hand representation and a more precise virtual interaction can be provided. The ART finger tracking device features an active hand target and five finger targets that are placed on the user's finger tips as shown in 4.20a. Similar to the previously described hand target, the pose of the hand coordinate frame (x_h, y_h, z_h) refers to the world coordinate system and is transformed to the reference frame (x_{b0}, y_{b0}, z_{b0}) by ROSART. The finger targets (x_{fn}, y_{fn}, z_{fn}) refer to the local hand coordinate frame (x_h, y_h, z_h) and have to be transformed to refer to the coordinate frame (x_{b0}, y_{b0}, z_{b0}) as well.

A simplified hand model as shown in Figure 4.18b was presented by Aslandere (2013). It uses the finger tip pose to visualize a simplified hand representation. As shown in Figure 4.20b, the bending of the finger joints cannot be visualized with this hand representation as all the finger phalanxes are joined into one geometry.



(a) ART finger tracking (b) Simplified hand model (c) Complete hand model

Figure 4.20.: The three types of hand and finger tracking representations.

A more sophisticated hand model is introduced in this research. The ART tracking system delivers the deflection of the finger joints (Advanced Realtime Tracking GmbH, 2016, p. 209). This data is used to animate the hand model, which is split into single phalanxes, as illustrated in Figure 4.18c. With this additional data, which is published via ROS, the flexing of the finger phalanxes can be visualized. This results in a more natural hand representation, as shown in Figure 4.20c.

The implemented hand representations are the basis for different types of interaction with the cockpit, which will be presented in the following sections.

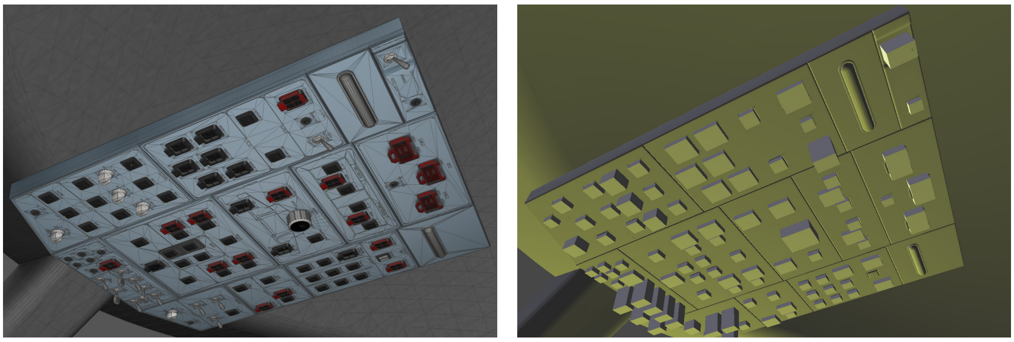
Fully Virtual Interaction

In a cockpit, different types of control elements are present. According to Schmidt (2011) these are:

- Push-buttons,
- guarded push-buttons,
- toggle switches,
- rotary knobs,
- levers, and
- touchscreens (in future cockpits).

All these elements can be controlled fully virtual in the VRFS. Implementing virtual interactions in a virtual reality environment requires a collision detection system and an interpreter. This module, the Collision Handler, has to monitor the position of the virtual hand and fingers and trigger an action if a collision with the virtual hand and a virtual control element occurs. Aslandere (2013) demonstrated the use of Unity3D, a widely used game engine (Creighton, 2010, p. 8), for this purpose. In this research, this approach will be further developed by integrating additional interaction methods and leveraging the capabilities of the Robot Operating System.

In Unity3D, the 3D geometry of the cockpit has to be imported. Based on this geometry, so-called box colliders, which are cube-shaped collision primitives (Unity Technologies, 2016), are used to mark the position of control elements. Thus, boxes are placed at the spatial position of the virtual control elements. For example, Figure 4.21b shows the box colliders of the overhead panel. The Unity3D component is connected to ROS using



(a) The overhead panel in AC3D

(b) The overhead panel in Unity3D

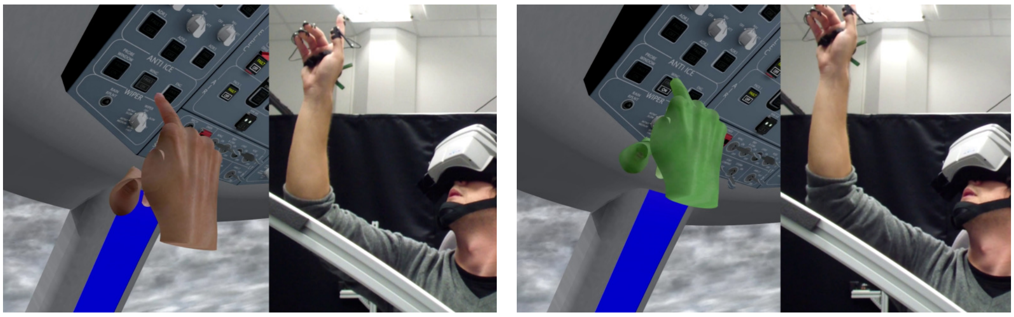
Figure 4.21.: The overhead panel in AC3D (left) and in Unity3D with the box colliders (right).

the websocket server that is provided by rosbridge (Crick et al., 2011). It subscribes to the available hand or finger tracking pose, which is mapped to sphere objects. If these spheres collide with a box, a certain action is triggered. In order to avoid double activations, a debounce filter is implemented for each collision box individually. This means that the collision detection is deactivated for this specific control element for a certain amount of time, after a first collision has been detected.

Push-buttons. In the case of push-buttons, upon collision a command is sent via ROS. This command is received by the ROSXPlane plug-in and translated to an X-Plane internal command. In order to show the user in the virtual environment that a button has been activated, some type of feedback is necessary. Usually, pressing a button leads to a system change and a new indication of the button or an indication close to it. Yet, in some situations, a system cannot be triggered and consequently pressing a button has no effect. In such cases, the user cannot distinguish if this is a system or an interaction problem. Hence, additional indicators for pressing a button in the virtual environment have to be implemented. A Lua script, based on the X-Plane plug-in Gizmo64 (Russel, 2016), checks if a button is being pressed in X-Plane and consequently triggers an animation. This push animation is one indicator for the user. Another indicator, also triggered by the Lua script, is a click sound. The third indicator that is used in the VRFS to indicate collisions is a change of color of the hand representation as shown in Figure 4.22b.

The ability to interact with virtual push-buttons is described in the use case 1.1 *Trigger a Virtual Button* presented in Table 3.1 on page 22.

Guarded Push-buttons. In order to activate guarded push-buttons, two individual actions are necessary: The user has to press the guarded button in order to open the guarding mechanism. This first interaction triggers an animation that shows the opening of the guard. When the guard is open, a second push activates the button and the underlying system. This logic is also implemented with the help of a Lua script.



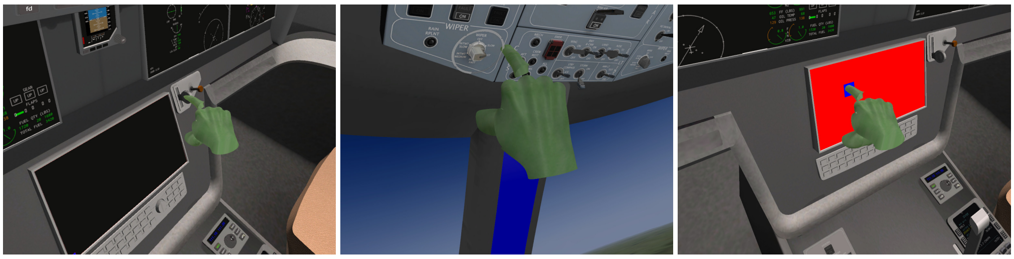
(a) The virtual button (b) Collision with the virtual button

Figure 4.22.: The interaction with a virtual push-button.

Operating these guarded buttons is part of the use case 1.2 *Trigger a Virtual Guarded Button* described in Table 3.1 on page 22.

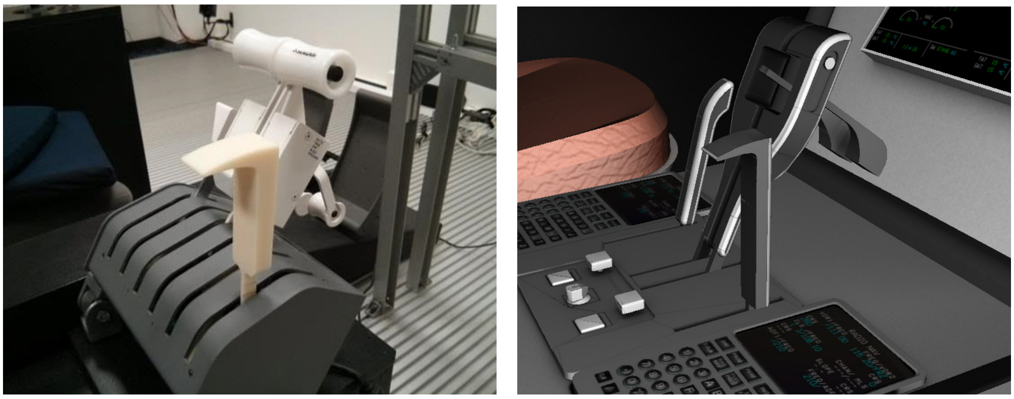
Switches. The behavior of switches is similar to that of push-buttons. A simple collision triggers the next switch position. A Lua script processes the incoming command and changes the switch to the next position. If the switch is at the last position, it will be reset to the first available position.

Rotary Knobs. There are two types of rotary knobs: Knobs with discrete positions, i.e. rotary switches and knobs that can be adjusted continuously, i.e. rotary encoders. If the rotary knob is discrete, the interaction is implemented similar to the switch interaction. Every activation rotates the knob to the next position. Continuous rotary knobs however need a more complex interaction. First, a finger has to enter the knob's collision volume. After that, the position of the finger is bound to the rotation of the knob. By cycling the finger around the knob, it can be turned. A circle visualization around the knob, as shown in Figure 4.23b, helps the user to perform this interaction (Oberhauser, Dreyer, and Kovačič, 2015). If the user retreats his or her finger from the collision volume, the interaction stops.



(a) Virtual lever (b) Virtual rotary knob (c) Virtual touchscreen

Figure 4.23.: Complex interactions with virtual control elements that require the use of gestures.



(a) Real throttle quadrant (b) The throttle quadrant in the virtual cockpit

Figure 4.24.: The throttle quadrant in a mixed mock-up setup.

Levers. The use of levers in the cockpit is similar to continuous rotary knobs. The user has to touch the lever, which connects the lever position to the finger position. This way, for example the gear lever as shown in Figure 4.23a can be manipulated in a natural way. If the hand is retreated, the position of the lever rests at the last position and the interaction stops.

Touchscreens. The user can also interact with fully virtual touchscreens. A collision box covers the entire virtual screen. In Unity3D, the x and y coordinates of the entry point into the collision box can be determined and can be published via ROS. This information can be used by X-Plane internally or by an external component to emulate a touchscreen. Figure 4.23c shows a virtual touchscreen that is being pressed, implemented in the VRFS.

Haptic Feedback

The absence of haptic feedback makes the interaction with fully virtual control elements with a hand or finger challenging, as the user has to aim for buttons in mid air. Adding haptic feedback to virtual reality is a long standing challenge that so far has been addressed with special gloves or robotic arms (Stone, 2000). A simple but effective way to add haptic feedback is emulating it by adding wooden or acrylic plates to the simulation environment. The position of these plates has to match exactly the virtual counterpart. A more advanced solution are non-functional cockpit elements, which can be 3D printed and represent certain cockpit elements. In the VRFS, non-functional 3D printed button panels and wooden plates are used in the pedestal for providing haptic feedback.

(a) Mixed mock-up Interaction

(b) Interaction with a checklist

Figure 4.25.: The interaction with real life objects in the VRFS.

Mixed Mock-up Interaction

If functional hardware elements are added to the system and placed at the exact same spatial position as in the virtual model, a so-called mixed mock-up (also known as mixed reality) is created (see Figure 4.24). Usually, these functional hardware elements are directly connected to the flight simulation via a Universal Serial Bus (USB) connection. In the virtual environment, the user interacts with a virtual representation of the hardware component, while actually operating the real hardware device as illustrated in Figure 4.25a. As the hardware component is connected to the simulation, audiovisual feedback is provided in the virtual environment, e.g. the position of a lever changes accordingly or a value changes as the result of a rotary knob input.

Mixed mock-up elements are used in the VRFS for frequently used and time critical elements as well as for control elements that require continuous input. These are throttle, flaps, and speed brake levers, the flight stick, rudder pedals as well as the Flight Control Unit (FCU). These elements are hard to control fully virtually and they rely on haptic feedback. Hence, using the VRFS without these components compromises the usability of the system.

The described mixed mock-up concept satisfies two use cases that have been described in Table 3.1 on page 22: 1.3 *Trigger a Physical Control Element* and 1.4 *Use a Hardware Lever*.

Additional Objects

For some scenarios, it might be necessary for a pilot to use additional objects that are detached from the cockpit, like a Flight Crew Operating Manual (FCOM), an Electronic Flight Bag (EFB) on a tablet computer, or an approach chart. To visualize these objects in the VRFS, simple mock-ups can be equipped with an ART target. The pose information of these markers are processed by ROSART and the ROSXPlane plug-in. In the X-Plane aircraft, a 3D representation of these objects has to be present at the center of gravity (x_{cg}, y_{cg}, z_{cg}). These 3D objects can be connected to the transformed pose information from the ART target similar to the hand representation. This leads to an object with

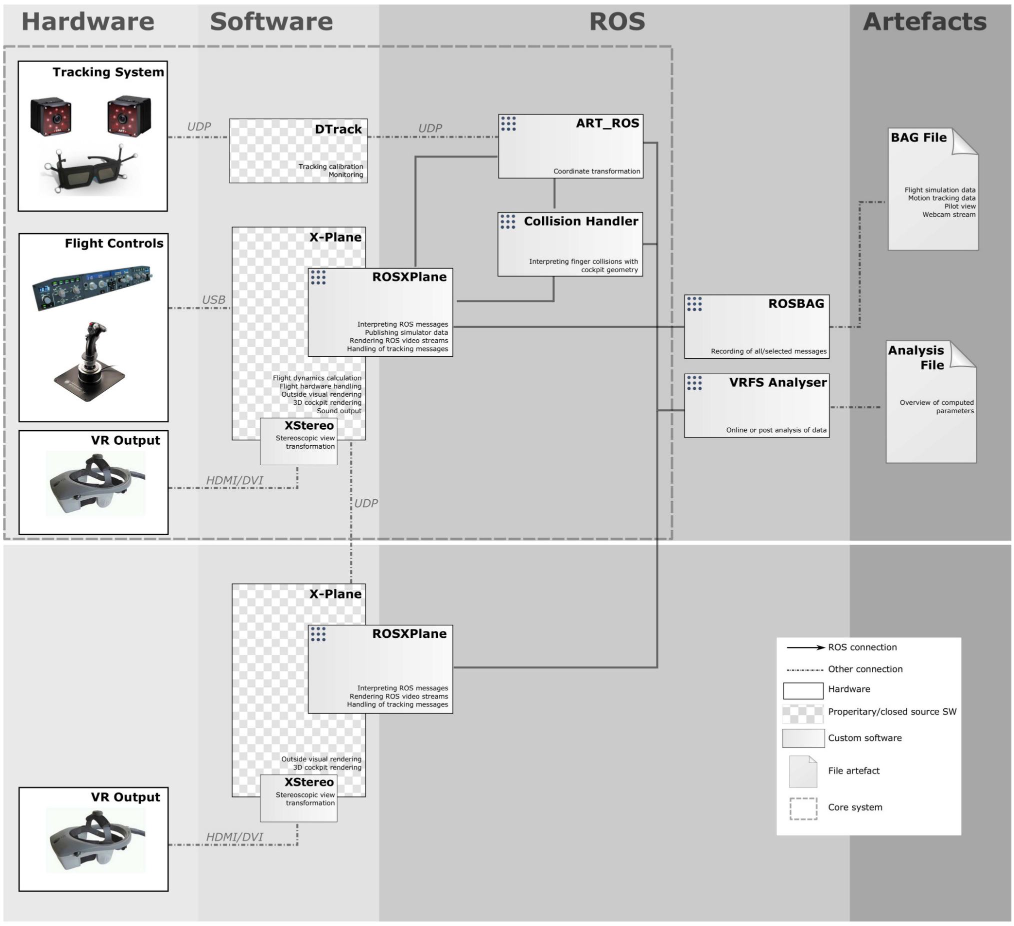
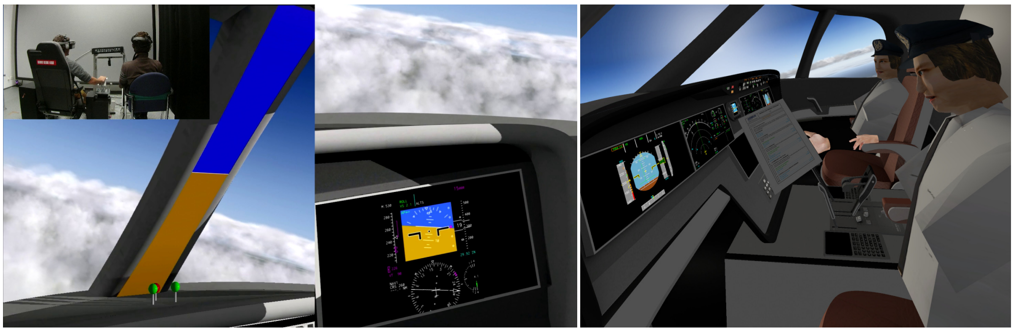


Figure 4.26.: The architecture of the VRFS with two pilots. For this setup, a second instance of X-Plane is added to the base system alongside a second head-mounted display. The second X-Plane instance is passive and does not publish messages via ROS.

haptic feedback that can be picked up and used as shown in Figure 4.25b. In this case, the FCOM pages can be changed by the operator. For other use cases, more interactive content might be desirable. Here, a user would be able to change the content by pressing a virtual button on the object.

4.3.4. Multi-user Setup

Today's cockpit is a collaborative two person workspace. All operational procedures are based on a Pilot Flying (PF) and a Pilot Not Flying (PNF). Both pilots are allocated specific tasks to ensure a safe operation of the aircraft. This so-called Crew Resource Management (CRM) is an essential part of operating a commercial aircraft (Helmreich and Foushee, 1993). In consequence, the possibility to interact with a second pilot also has to be implemented in the VRFS.



(a) Two viewpoints (b) Avatars in the cockpit

Figure 4.27.: Virtual collaboration in the Virtual Reality Flight Simulator: Each pilot has a head-mounted display with an independent head tracking target and a corresponding visualization (left). Both pilots can interact with each other as they see simplified avatars of the other pilot (right).

In order to immerse another user into the virtual environment, a second head-mounted display with a distinct ART tracking target and a second instance of X-Plane to render the other user's viewpoint are necessary, as shown in Figure 4.26. This second X-Plane instance does not calculate the flight loop or atmospheric effects but receives the data through the X-Plane built-in master slave UDP interface, which synchronizes selected system variables. This ensures that the virtual cockpit of the second user is at the exact same position with the same outside visual as the virtual cockpit of the first user. This UDP interface is less than perfect as it only synchronizes some of the available system variables or datarefs. In particular cockpit related information like the status of switches, levers, or display information are not at all, or only insufficiently covered by this built-in interface. Thus, the missing information should be provided by the master X-Plane instance via ROS. In addition, the display content of the master instance can be streamed into the slave virtual cockpit. All inputs actions by both pilots (hardware USB inputs or virtual touch inputs) are processed by the master X-Plane instance. All these measures ensure that the aircraft systems of both instances are conformal and cannot diverge.

The positional tracking of the second pilot's head is provided by ART and the ROSART node. In the ROSXPlane configuration file of the slave instance, the pose of the second pilot has to be attached to the in-game camera. The pose of the first pilot has to be assigned to a 3D manikin, which is present in the X-Plane aircraft at the center of gravity (x_{cg}, y_{cg}, z_{cg}). In the master instance, this configuration has to be adjusted vice-versa: The 3D manikin has to be connected to the second user's head tracking and the X-Plane camera position has to be attached to the first pilot's head tracking pose. This results in two individual viewpoints in a shared virtual environment, as shown in Figure 4.27a, and the ability for each pilot to see and interact with each other in the form of a virtual 3D manikin as shown in Figure 4.27b.

The described multi-user setup matches the use case that has been described in Table 3.1 on page 22: 1.6 *Interact with a Second Pilot*.

4.4. Prototype Integration

One of the main purposes of the VRFS is to enable the quick evaluation of novel HMI concepts. In many cases, at some point in the development process these concepts, like displays, touchscreens, head-up displays or hardware panels are implemented as a single system software or hardware prototype. In contrast to the actual flight hardware, there are no standards or common toolkits for developing these kind of prototypes. Thus, for integrating these inhomogeneous components into a simulation environment, flexible and minimum effort interfaces have to be provided. The Robot Operating System with its multi-platform and cross-programming language support and an implementation that only requires a few lines of code, meets these requirements (Oberhauser, Dreyer, Convard, et al., 2016).

Prototypes that are developed for the flight deck can either be system prototypes, i.e. new systems with no or minimal human-machine interfaces, or hardware prototypes like panels with buttons and switches, as well as display prototypes. In the following sections, the integration of these kinds of prototypes will be presented, a use case described in Table 3.1 on page 22 as 2.4 *Integrate Prototypes*.

4.4.1. System Prototypes

Prototypes of system simulations like new fly-by-wire systems, autopilots or fuel systems can have a huge impact on a simulated aircraft on a system level but may have no implications on the cockpit HMI. These pure system prototypes receive data from the simulation, compute a new system state, and propagate this new system state to the simulation. By adding a ROS publisher and subscriber to a system simulation (see Figure 4.28) and by exploiting the flight simulation topics and commands provided by the ROSXPlane node, system prototypes can be integrated into the VRFS with minimum effort.

Depending on the system that is going to be simulated, there might be overlaps with X-Plane or aircraft specific system simulations. Thus, X-Plane offers datarefs for overriding or disabling certain system simulations. With these override datarefs, systems like the autopilot, the flight loop, or the joystick input can be deactivated. Yet, this override functionality is not excessive. This means that some system simulations have to be implemented directly by using an X-Plane plug-in or even need changes in the aircraft model. This has to be addressed on a case-by-case basis depending on the system prototype that will be integrated.

4.4.2. Display Prototypes

In modern so-called glass cockpits, flight, system and navigation information is presented to the pilots on multiple Cathode Ray Tubes (CRTs) or Liquid Crystal Displays (LCDs) (Wiener, 1989, p. 2). With every new aircraft generation, the emphasis moves from manual hardware controls and indicators to more information on displays or even touch displays (Barbé, Wolff, and Mollard, 2013). To a considerable extend,

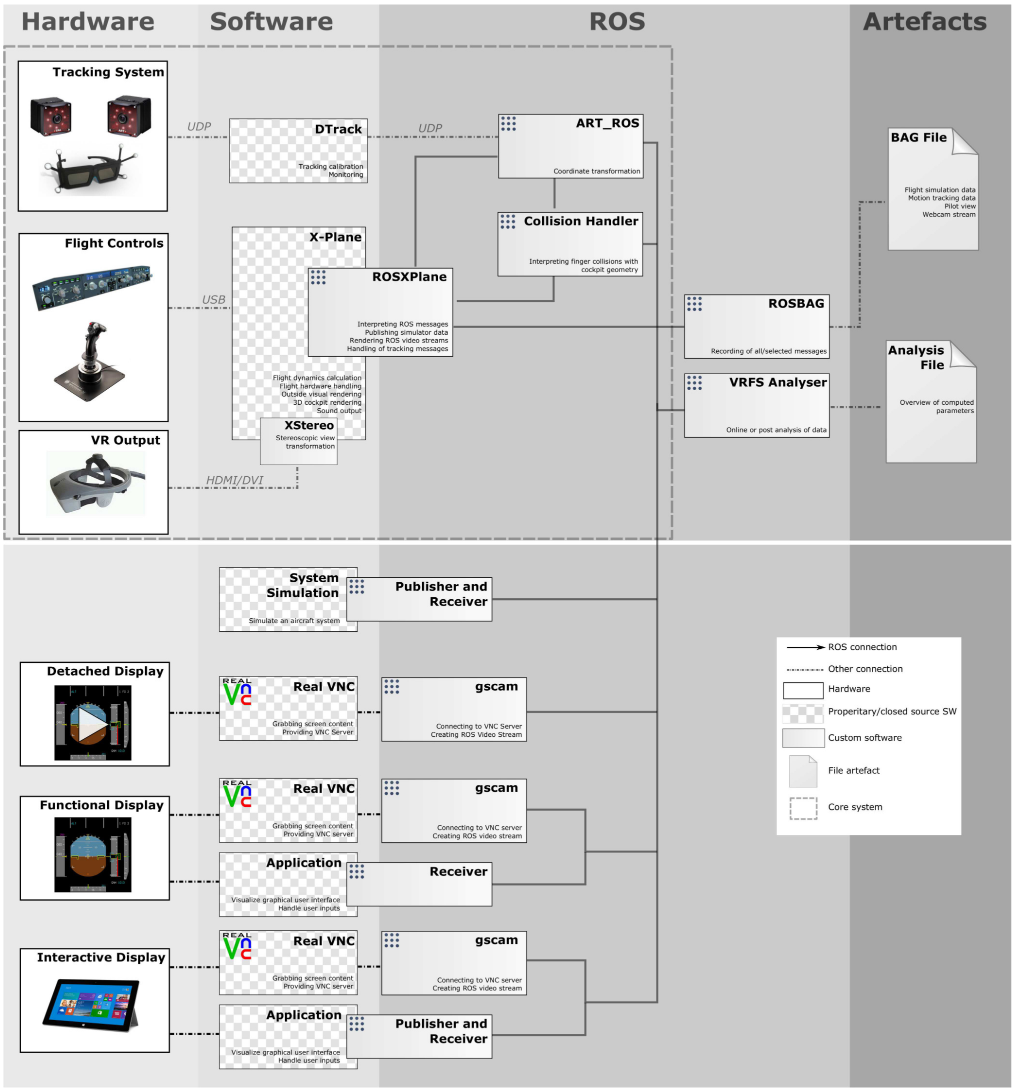


Figure 4.28.: Integration of external system simulations and displays into the VRFS. All these prototypes are connected via ROS. Display prototypes use gscam in combination with RealVNC to stream the display content into the virtual cockpit.

today's research on Human–computer Interaction (HCI) in the cockpit domain focuses on new displays and display content, i.e. display symbology. Thus, the ability to integrate new types of displays and symbology into the VRFS for evaluation purposes is crucial.

One way of integrating display prototypes into the VRFS is to use the built-in capabilities of X-Plane. With Plane Maker, the animation capabilities of AC3D, and with OpenGL functions in custom plug-ins, several methods for implementing display prototypes exist.

External display prototypes, i.e. prototypes that are not specifically made for X-Plane, can be integrated by leveraging capabilities of the Robot Operating System of image streaming and the ability to access flight simulation telemetry data. Different levels of display prototype integration exist: (a) The streaming of a static image, a video, or an independent simulation of a flight display, i.e. a detached display; (b) a display that is driven by telemetry data; and (c) an interactive display that has bi-directional access to the flight simulation's telemetry data (Oberhauser, Dreyer, Convard, et al., 2016). The integration of these display prototypes into ROS is illustrated in Figure 4.28 and will be explained in the following subsections.

Detached Displays

Detached displays need a uni-directional connection to the VRFS. The content of the display has to be published via ROS in order to visualize it in the cockpit. With the implemented texture streaming capability of the ROSXPlane plug-in described in Section 4.2.3 on page 49, this display stream can be mapped to a display or any other geometry in the virtual cockpit. Gathering and publishing the display content in the prototype can be implemented by extending the display simulation with ROS image streaming capabilities. Another non-intrusive way is the use of a Virtual Network Computing (VNC) server, a technique to access the screen content of remote computers (Richardson, Stafford-Fraser, et al., 1998).

A VNC server like RealVNC, which uses the Remote Frame Buffer (RFB) protocol, can be installed on the system that is running the display application (Richardson and Wood, 1998). Gscam[8] is a package for ROS, which uses GStreamer, a framework for displaying, streaming or converting video and audio data (Taymans et al., 2013, p. 1). Gscam uses gst-launch, a GStreamer command line tool to create video and audio transformation and streaming pipelines. With gst-launch, videos can be gathered from multiple sources, processed, and sent to a destination or, in other words, a sink. Gscam is started with a ROS launch file, a configuration file for ROS nodes as presented in Listing 4.6. This launch file, which uses the Extensible Markup Language (XML) syntax, defines the GStreamer pipeline. It uses a RFB source (`rfpsrc`) with the respective login information for the VNC server and publishes this image stream under a topic name such as `textures/desktop/nav_display`.

[8] See http://wiki.ros.org/gscam

Listing 4.6: The launch file for publishing a VNC connection.

```
  <launch>
  <!-- The camera name -->
  <arg name="cam_name" value="textures/desktop/nav_display" />

  <!-- Set up the gst-launch pipeline -->
  <env name="GSCAM_CONFIG" value="rfbsrc host=192.168.1.19
                                  password=Bingo
                                  width=640,height=240
                                  view-only=true !
                                  videoscale ! videorate ! autoconvert!
                                  video/x-raw-rgb,
                                  width=640,height=240,
                                  framerate=25/1,bpp=24,
                                  depth=24 ! ffmpegcolorspace"  />

  <!-- Start the ROS node -->
  <node pkg="gscam" type="gscam" name="$(arg cam_name)">
  <param name="camera_name" value="$(arg cam_name)" />
  <param name="camera_info_url" value="package://localcam/calibrations/${NAME}.yaml"/>
  <remap from="camera/image_raw" to="/$(arg cam_name)/image_raw" />
  </node>

</launch>
```

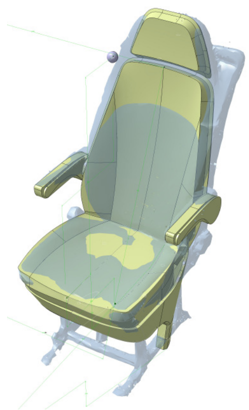
Functional Displays

Functional displays receive data from the flight simulation and return the image stream to the virtual cockpit. In contrast to detached displays, the content of functional displays represent actual system states of the simulated aircraft. Hence, a pilot can monitor system states with this type of display integration. However, some development effort has to be undertaken to connect the display prototype: It needs to be a ROS subscriber implemented with subscriptions to the topics that are required for the particular display. As shown before, the streaming to the simulation can be realized using a VNC connection via gscam; or the streaming functionality is implemented in the display software directly.

Interactive Displays

Interactive displays can use the same image streaming approach as detached and functional displays but have a bi-directional data connection to the flight simulator. This means that the display component, like a touchscreen or a display with line selection keys, can be used to manipulate system states of the flight simulation by publishing commands and values via ROS. For an immersed user, this behavior is only useful if he or she can interact with the display in the virtual environment. Therefore, the components have to be integrated in a way that they create a mixed mock-up, i.e. the display should have an exact representation in the virtual environment and it should be placed at a matching position in the hardware setup.

(a) Seat scan



(b) Remodeling

Figure 4.29.: The 3D scan of a real hardware component and the object remodeled in CATIA. This technique can be used to easily and accurately recreate hardware elements without CAD data available.

4.4.3. Hardware Prototypes

The integration of hardware prototypes use the same principle as mixed mock-up components described in Section 4.3.3 on page 63. A hardware component has to be placed at the exact same position as in the virtual cockpit in order to be usable for a person that is immersed in the system. The component has to have a connection to the flight simulation software. This can be realized via USB or via ROS.

If there is no 3D data for a hardware prototype, it has to be remodeled in a CAD application. To aid this process, a 3D scan can be performed with a low-cost sensor like the Microsoft Kinect and a reconstruction software like Skanect (Occipital, 2016). Based on this 3D scan, the component can be remodeled as shown in Figure 4.29.

4.5. Integration of Human Factors Methods

In order to evaluate novel cockpit concepts and integrated HMI prototypes, several human factors methods have been integrated into the VRFS. Similar to the integration of prototypes, most of the integrated human factors methods that are presented in this section leverage the capabilities provided by the ROS middleware. A central requisite for many of these methods is the ability to record, replay, and analyze the data traffic in the ROS network into rosbags. Figure 4.30 shows the architecture of the VRFS and integrated human factors methods.

The integration of human factors methods is a use case that was described in Table 3.1 on page 22 as 5.3 *Integrate Human Factors Methods*.

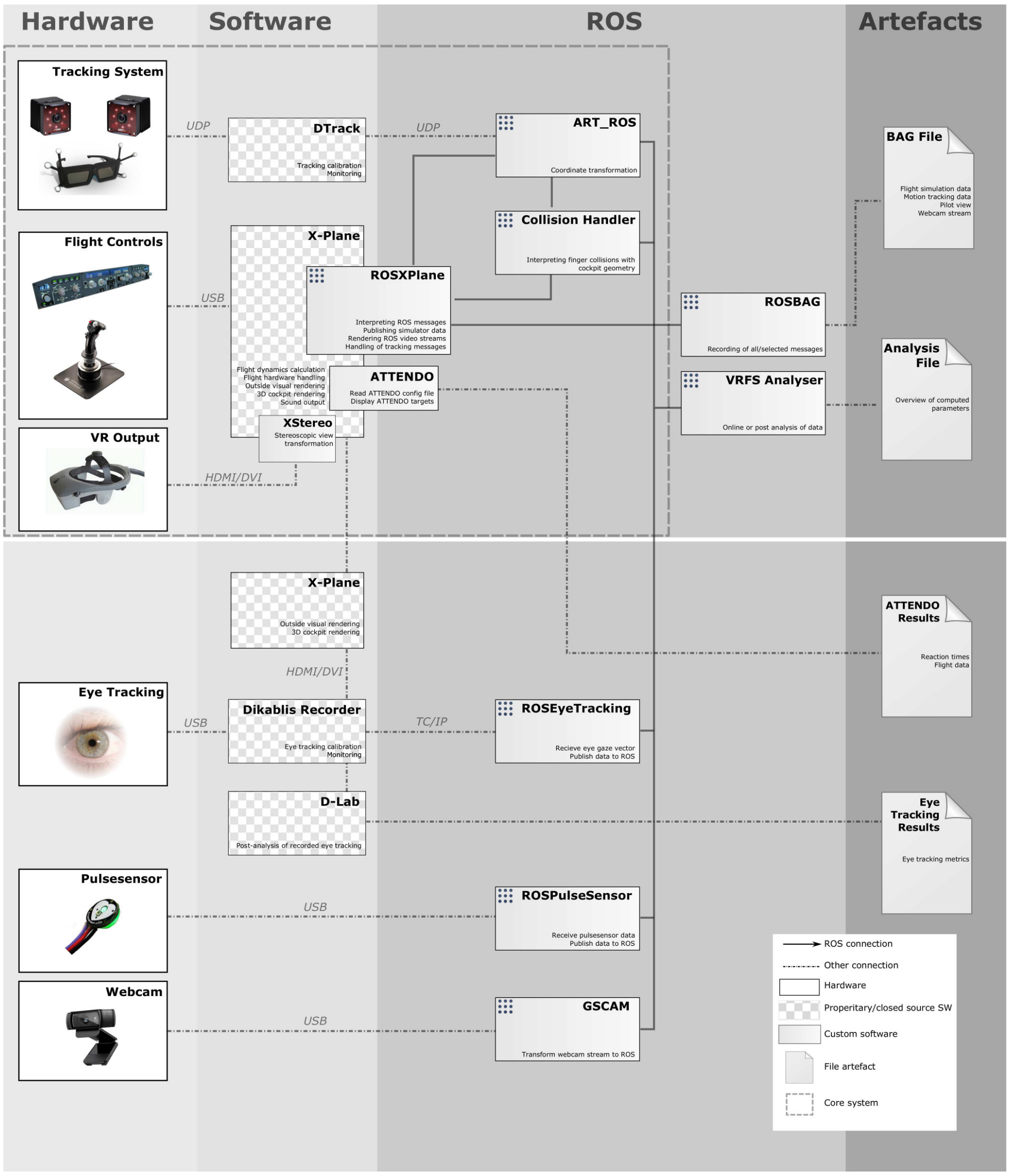
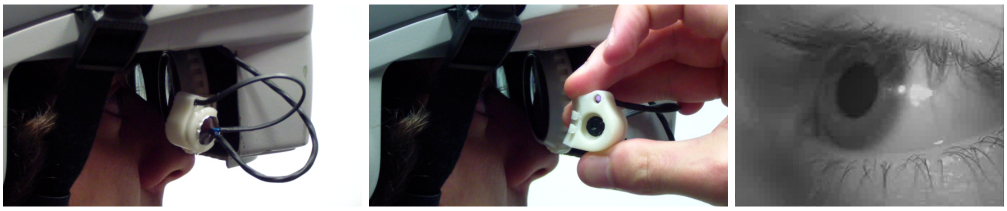


Figure 4.30.: The architecture of the VRFS with integrated human factors methods. The eye tracking, the pulse sensor, and the webcam are connected via ROS. ATTENDO is implemented as a plug-in for X-Plane.



(a) Eye tracking integration (b) Eye tracking camera (c) Eye camera output

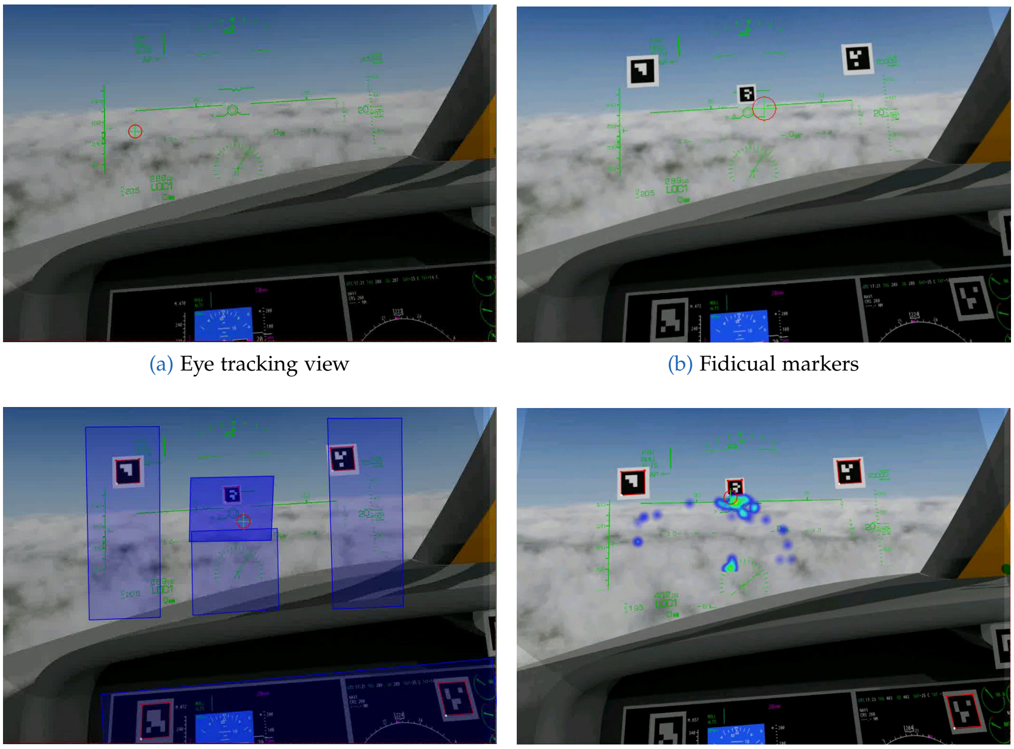
Figure 4.31.: Integration of eye tracking into the head-mounted display (left). The eye tracking camera and an infrared LED are attached to a ring around the eye piece (middle). This results in a clear and level view on the user's eye (right).

4.5.1. Eye Tracking

Liesecke (2013) presented the integration of an eye tracking system into a NVIS SX60 head-mounted display. This system consists of a 3D printed ring that is placed at the optics of the HMD as shown in Figure 4.31a. This ring is used to attach a bracket that holds an eye tracking camera and an LED as shown in Figure 4.31b (Liesecke, Dreyer, and Kocvara, 2015). This bracket can be attached at different positions around the eye to ensure an unobstructed view for the eye camera and a central and leveled position of the pupil (see Figure 4.31c) (Liesecke, 2013, p. 49).

The eye tracking system that is used in this research is the Dikablis Eye Tracking from Ergoneers, which was originally developed by Lange (2005). The system consists of a forward facing field camera to capture the view of the wearer, an eye camera to monitor the eye movements, and a rim into which these two cameras are integrated. Both camera feeds are processed in a software called Dikablis Recorder. The center of the pupil is determined and after a one time calibration, the eye gaze is superimposed on the image of the field camera (Ergoneers GmbH, 2011b). For the eye tracking system that is integrated into the HMD, the field camera is replaced by the visualization from X-Plane and the eye tracking camera is attached to the 3D printed bracket that was developed by Liesecke, Dreyer, and Kocvara (2015).

For further analysis, a software called *D-Lab* can be used (Ergoneers GmbH, 2011a). With this tool, eye tracking metrics like fixations, dwell time or the number of saccades can automatically be calculated from the recorded session. For this purpose, Areas of Interest (AOIs) are defined in the video feed of the field camera (Poole and Ball, 2006). These AOIs can be attached to fiducial markers in order to compensate the head movement. In the virtual cockpit, markers have been added as 3D object as shown in Figure 4.32b. As these marker objects obstruct the pilot's view, the field camera video feed is taken from a second synchronized X-Plane instance shown in Figure 4.30. In consequence, the user in the virtual environment, with the master X-Plane instance, works with a cockpit without markers. Figure 4.32c shows the areas of interest that are bound to the position of the markers and that are the basis of most of the calculated eye tracking metrics. Figure 4.32d shows a heatmap that can be used to visualize the eye gaze through out time.



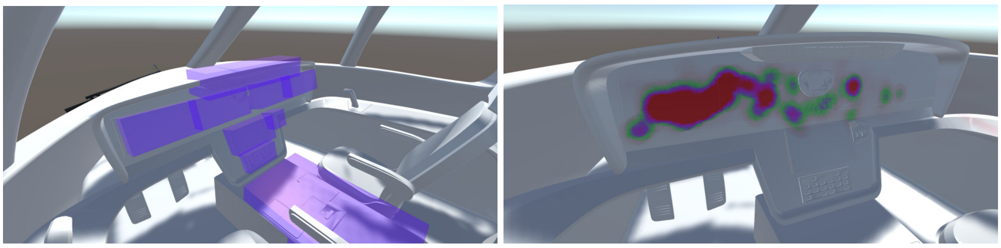
(a) Eye tracking view
(b) Fiducial markers
(c) Areas of interest
(d) Eye gaze heatmap

Figure 4.32.: The eye tracking view and post processing analysis techniques. The fiducial markers (top right) are not visible for the pilot.

The integration of the Dikablis eye tracking as developed by Liesecke (2013) only allows offline use and post-analysis of the eye tracking data. For this research, the eye tracking data should be made available through the ROS network during run time of the simulation. For this purpose, the network interface of the Dikablis Recorder was used (Ergoneers GmbH, 2011b, p. 71). The eye tracking vector can be published and attached to the pilot's head using the ROS TF library. With this data available via ROS, online analysis and visualizations of the eye-gaze can be used. For example three-dimensional areas of interest or a three-dimensional heatmap as shown in Figure 4.33 can be implemented.

4.5.2. ATTENDO

The ATTENDO method relies on displaying targets in the field of view of the pilot. In the VRFS, these targets are implemented as sphere objects. The placement of these objects relies on the ability to add and place additional objects to the virtual environment similar to the additional objects described in Section 4.3.3 on page 63.



(a) 3D areas of interest (b) 3D heatmap

Figure 4.33.: Implementation of a 3D analysis of eye tracking data. This analysis tool is based on Unity3D and analyzes areas of interest (left) and a three-dimensional heatmap (right).

ATTENDO is implemented as a plug-in for X-Plane (Hillebrand, Oberhauser, et al., 2014). This plug-in reads a configuration file line by line which holds information on the position and the display time of each sphere. The user has to react to the targets by pressing a button. The time between appearance and reaction is saved in an output file after each session. The implementation as a plug-in ensures that the method can also be used in small scale experiments on single desktop based instances of X-Plane without the need of a virtual environment or the ROS middleware.

4.5.3. Heart Rate

In order to measure the user's heart rate, multiple options exist. First, a device like a sports watch with a heart beat monitor or a medical heart rate monitoring device can be used. These external devices use their own specific timestamps to save the heart rate data. Thus, the data that is collected has to be synchronized and post-processed after a simulator session.

An open source and open hardware heart rate monitor was developed by J. Murphy and Gitman (2016), the *Pulse Sensor Amped*. The small sensor can be attached to the user's ear or to a finger tip. It is controlled by an Arduino micro-controller, which provides the BPM, the IBI, and the raw heart pulse signal. A simple ROS node was designed, which publishes these three parameters via ROS as presented in Listing 4.7. Thus, the messages from the Pulse Sensor Amped can be recorded to a rosbag temporarily synchronized with other data; or they can be used during run time.

4.5.4. Performance

Depending on the given task and the research question, the performance indicators can differ. In some cases, the flight performance is of interest, e.g. how exactly a given trajectory is being followed. In other cases, the performance of a system management task is subject to research, e.g. how fast a technical malfunction can be resolved.

For all cases, it is necessary to access telemetry data. In the VRFS, all telemetry data can be saved in a rosbag for post-analysis. For this purpose, a Python application can

Listing 4.7: The ROSPulseSensor node.

```
#!/usr/bin/python

import math;
import rospy
from std_msgs.msg import Int32

import serial

bpm = rospy.Publisher('/physio/heartbeat/bpm', Int32, queue_size=10)
rr = rospy.Publisher('/physio/heartbeat/interbeat_interval', Int32, queue_size=10)
raw = rospy.Publisher('/physio/heartbeat/raw', Int32, queue_size=10)

rospy.init_node('PulseSensor', anonymous=True)

ser = serial.Serial('/dev/ttyACM0', 115200)
while True:
    data = ser.readline()
    if data[0] == 'Q': #Interbeat intervall
       rr.publish(int(data[1:]))
    elif data[0] == 'S': #Raw signal
       raw.publish(int(data[1:]))
    elif data[0] == 'B': #Beats per Minute
       bpm.publish(int(data[1:]))
```

be designed to read certain values from this rosbag file, analyze the data, and save the results into a file. This analysis can also be conducted during run time. Yet, for post-analysis and during run time, this script has to be designed on a case by case basis depending on the research question.

4.5.5. Questionnaires

Questionnaires (e.g The NASA-TLX for evaluating workload or the SART for evaluating situation awareness) can be handed to a user as printouts after a virtual reality simulator trial (Oberhauser, Dreyer, Mamessier, et al., 2015). As suggested by Dreyer and Hillebrand (2010), these questionnaires can also be displayed inside the virtual environment in order not to interrupt the immersion. With the mixed mock-up presented in Section 4.3.3 and the ability to stream external display content into the VR as presented in Section 4.2.3, this functionality can be implemented. Therefore, the display of an electronic version of a questionnaire is streamed to a mixed mock-up object like a clipboard. The questionnaire can either be filled out by an operator, as demonstrated in the research of Dreyer and Hillebrand (2010) or by the immersed user him- or herself by providing a virtual interface that can be controlled with finger tracking.

4.5.6. Posture Assessment and Analysis

All methods for posture analysis rely on the direct input of the user's posture. Video data can be one source for posture assessment and analysis. Thus, in the VRFS, provision is made for including external video cameras to the experiment setup. As shown in

Table 4.2.: The verification of the use cases and their requirements.

No.	Name	Implementation
Pilot Use Case		
1.1	Trigger a Virtual Button	Fully Virtual Interaction on Page 59
1.2	Trigger a Guarded Virtual Push-button	Fully Virtual Interaction on Page 59
1.3	Trigger a Physical Push-button	Mixed Mock-up Interaction on Page 63
1.4	Use a Hardware Lever	Mixed Mock-up Interaction on Page 63
1.5	Use a Flight Display	Pilot's View Visualization on Page 54
1.6	Interact with a Second Pilot	Multi-user Setup on Page 64
Developer Use Case		
2.1	Import Cockpit Geometry	The Geometry Import Pipeline on Page 42
2.2	Animate Objects	Adding Animations on Page 44
2.3	Assign Dynamic Textures	The Panel Texture on Page 45
2.4	Integrate Prototypes	Prototype Integration on Page 66
Controller Use Case		
3.1	Load a Scenario	Scenarios on Page 51
3.2	Induce a System Malfunction	Scenarios on Page 51
Human Factors Practitioner		
4.1	Online Analysis of Data	Integration of Human Factors Methods on Page 70
Shared Use Cases		
5.1	Access Simulation Data	ROS Integration on Page 46
5.2	Record Simulation Data	Recording on Page 38
5.3	Integrate Human Factors Methods	Integration of Human Factors Methods on Page 70

Figure 4.30 on page 71, these video cameras are included by leveraging gscam, an approach similar to the texture streaming presented in Section 4.2.3. The video streams of these cameras can, already time synchronized, be stored in a rosbag for further analysis.

A direct approach for assessing a posture is the use of external tracking sensors or accelerometer-based devices (Li and Buckle, 1999). The optical tracking system of the VRFS already delivers 6DOF accurate and drift free information on the pilot's head and hands which makes it ideal for posture analysis. The system can be extended by adding further tracking targets to the limbs of the user.

4.6. Verification

Verification of a software is the process of reviewing wether the requirements, set at the beginning of a project, have been met (IEEE Computer Society, 2005). The implementation of the use cases that have been identified in Section 3.1 on page 19 was presented throughout the current chapter. Table 4.2 lists the use cases and points out the specific sections, in which each of the use cases has been implemented.

In summary, all use cases and the underlying requirements have been met, either by the used flight simulation, the functionality of the Robot Operating System or the software that has been implemented for this purpose.

5. Evaluation

In this chapter, both the usability and fidelity of the Virtual Reality Flight Simulator will be evaluated. Two categories of simulation fidelity that are related to the behavior of the pilot are of special interest: First, the psychological-cognitive fidelity, i.e. the level of realism of the experience, and second, the task fidelity, i.e. the user's ability to interact and recreate tasks in the simulator. As mentioned previously in Chapter 2.3, according to Liu, Macchiarella, and Vincenzi (2009) two approaches for quantifying fidelity exist. First, with a set of metrics, a qualitative assessment of fidelity can be conducted. Yet, there are some issues with this method: There are no standardized metrics; various definitions of fidelity exist; and the results are based on subjective ratings. Thus, no consistent and comparable results can be compiled using this method. A second, more objective way of measuring fidelity is comparing the simulator with its real counterpart (Perfect et al., 2013). In this research, the latter approach is chosen and the Virtual Reality Flight Simulator is compared to a conventional hardware simulator. Due to the level of complexity of a simulated aircraft cockpit, it is not possible to compare and quantify every aspect of the simulator (Liu, Macchiarella, and Vincenzi, 2009). Hence, only selected metrics will be considered as part of this evaluation: The interaction and flying performance (task fidelity) on the one hand and the situation awareness and workload (psychological-cognitive fidelity) on the other. In addition, the effect of simulator sickness in both environments will be evaluated.

Research on the feasibility and the usability of flight simulators in virtual environments already exists (Aslandere, Dreyer, and Pankratz, 2015; Bauer and Klingauf, 2008; Dörr, 2004). Yet, these studies only cover single aspects of usability, like the interaction with cockpit elements, and have not taken into account high workload induced by an operational context, i.e. they have not been performed during a simulated flight. Furthermore, no comparative studies on the usability of virtual flight simulation environments versus conventional simulators or even in-flight cockpits are available, so far.

The studies on simulation fidelity that are conducted as part of this research use a conventional hardware cockpit as a baseline. As the generic conceptual cockpit that is used in the typical setup of the VRFS has no physical counterpart, a general aviation flight simulator, located at the Graz University of Technology, is used for the trials. This simulator, as shown in Figure 5.1a features a 190 degree outside visual and has a glass cockpit with two PFDs and one Multi-function Display (MFD). It features a central flight stick, engine controls in the central pedestal, a Flight Control Unit, and an overhead panel.

For this evaluation, the cockpit of the conventional flight simulator has to be integrated into the VRFS. As no 3D model of the cockpit is available, the flight deck and its components have to be remodeled. Therefore, the cockpit is 3D scanned using a

(a) The conventional simulator.

(b) The cockpit in the virtual environment.

Figure 5.1.: The general aviation flight simulator at the Graz University of Technology with the outside vision system (top) and the cockpit in virtual reality (bottom).

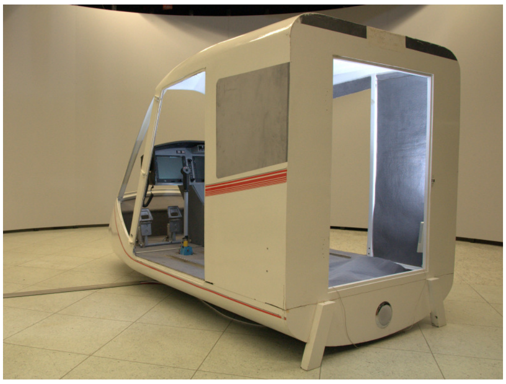
Table 5.1.: An overview of the evaluation trials, the type of fidelity, and the use case that is connected to the metric. Some of the metrics are not directly part of a use case but are important for the evaluation of the system.

Metric	Trial	Type of Fidelity	Use Case
Cockpit Interaction			
Virtual Buttons	Interaction Test	Task Fidelity	1.1
Hardware Controls	Simulator Trial	Task Fidelity	1.4, 1.3
External Display	Simulator Trial	Task Fidelity	2.4
Situation Awareness			
Head Movements	Simulator Trial	Cognitive-Psychological	-
Spatial Position Estimation	Simulator Trial	Cognitive-Psychological	-
Performance			
Traffic Circuit	Simulator Trial	Task Fidelity	1.5
Workload			
Physical Workload	Simulator Trial	-	-
Mental Workload	Simulator Trial	Cognitive-Psychological	-

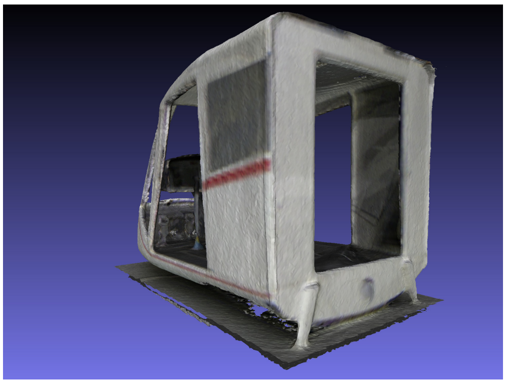
Microsoft Kinect device and the software Skanect (see Figure 5.2b). This 3D scan was used as a basis for remodeling the cockpit in CATIA V5 as shown in Figure 5.2. With the import pipeline presented in Section 4.2.2 and textures that have been generated by photographs of the cockpit, the cockpit geometry is integrated into the VRFS. In a standalone version, this cockpit geometry is being attached to a standard X-Plane flight model, a Cessna 400. In order to make the virtual flight deck usable, the functionality of buttons, displays, and primary flight controls have been established as well. This results in a virtual cockpit that is very close to the conventional hardware simulator as shown in Figure 5.1b.

The evaluation presented in this chapter consists of two user trials: In one study, an operational scenario is conducted in which interaction, workload, performance, situation awareness, and simulator sickness are compared to the same metrics in the conventional simulator. In another trial, the interaction with one particular cockpit element, the virtual overhead panel, is evaluated in a non-operational scenario in a laboratory environment. Table 5.1 shows the aspects that are analyzed in both user trials and the type of fidelity that is being evaluated. Some of the analyzed metrics refer to requirements that have been identified in Section 3.1. In the following section, the methodology of the two user trials will be described; in the proceeding sections, the results of the user trials will be presented. At the end of this chapter, the implications on the fidelity and usability of the VRFS will be discussed.

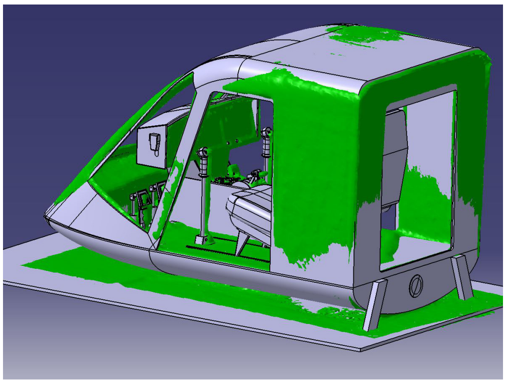
The results of the first trial have been presented at the *32nd Conference of the European Association of Aviation Psychology* (Oberhauser, Dreyer, Braunstingl, et al., 2017a,b) and a preliminary version of this chapter has been accepted for publication by the Journal *Aviation Psychology and Applied Human Factors* (Oberhauser, Dreyer, Braunstingl, et al., 2018).



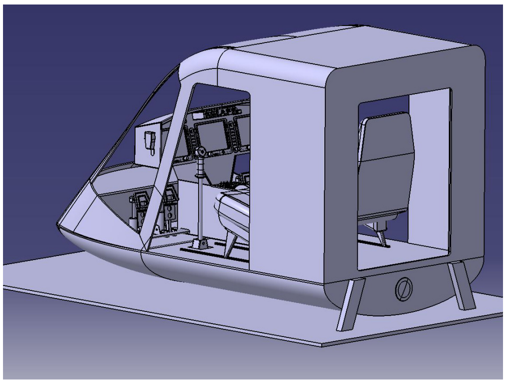
(a) The conventional simulator.



(b) 3D Scan of the simulator



(c) The superimposed 3D scan in CATIA V5



(d) The final CATIA V5 model

(e) The cockpit in X-Plane

Figure 5.2.: The 3D modeling process of the general aviation cockpit: Scanning the cockpit (top), remodeling in CATIA V5 (middle) and the final cockpit in X-Plane (bottom).

(a) Real (b) Virtual

Figure 5.3.: The conventional hardware simulator (left) and the cockpit with an immersed pilot (right) (Oberhauser, Dreyer, Braunstingl, et al., 2017a).

5.1. Methodology

5.1.1. Simulator Trial

Apparatus

Two almost identical hardware setups are used for this trial: The conventional environment and the virtual environment. In both setups, the user sits inside the conventional hardware cockpit. Yet, in the virtual reality environment, the user experiences the cockpit and the outside visual through the head-mounted display as shown in Figure 5.3. The user is equipped with two hand tracking targets in order to be able to see his or her own hands and enable interaction in the virtual environment. The roof of the conventional simulator is removed to allow the optical tracking system an unobstructed view on the pilot's head and hands. With this mixed mock-up approach conformability between the two simulation environments, conventional and virtual reality, can be ensured and the influence of confounding factors can be minimized as the pilot experiences the same haptic feedback in both environments. For the participant, this means that if he or she touches a control element in the virtual environment, he or she simultaneously touches this control element in the real hardware mock-up, which leads to the respective haptic sensation. In order to not only enable a visual but also a functional mixed mock-up, several aspects of the VRFS have to be connected to the simulation environment of the conventional flight simulator.

The simulator at the Graz University of Technology has a distributed architecture with various modules that communicate via UDP broadcasts. All information about the state of the simulation is available in this network, e.g. the status of hardware elements, the aircraft position, environmental information or aircraft telemetry data. This UDP interface has been used to connect the virtual cockpit to the conventional simulator. Therefore, a ROS node called TUG2VRFS was designed (see Figure 5.4). This application

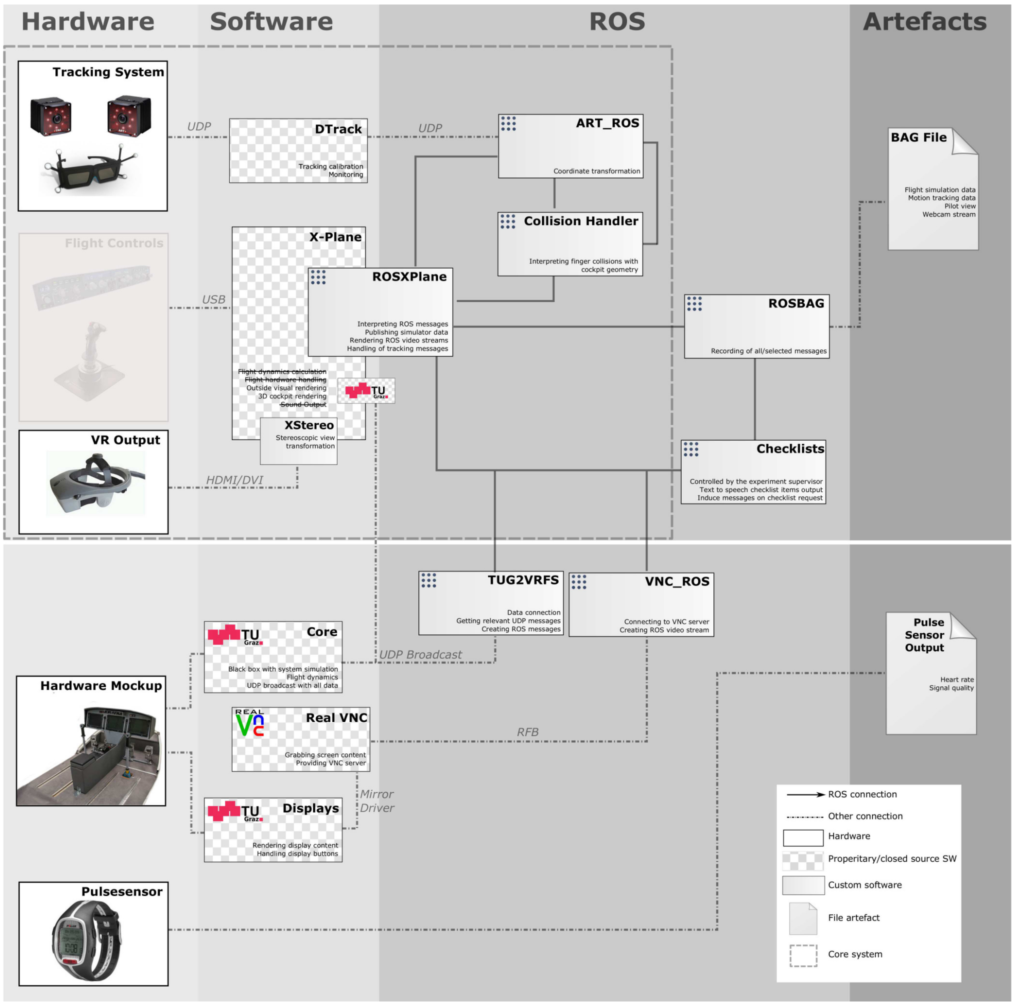


Figure 5.4.: The architecture of the VRFS connected to the general aviation simulator at the Graz University of Technology. The flight loop, the displays, and the position of cockpit elements are provided by the components of the conventional hardware simulator (Oberhauser, Dreyer, Braunstingl, et al., 2017a,b).

(a) Interaction (b) The mixed mock-up

Figure 5.5.: Interaction with hardware elements of the cockpit (left) and an illustration of the mixed mock-up (right) (Oberhauser, Dreyer, Braunstingl, et al., 2017b).

receives UDP messages and translates these messages to the respective ROS topics. With this interface application, the state of the cockpit elements, i.e. switches, levers, buttons, is connected to the state of the hardware elements of the conventional simulator; thus, a functional mixed mock-up is created.

The conventional simulator at the Graz University of Technology also uses X-Plane as a rendering engine for the outside visual. A custom plug-in controls weather, time, and the aircraft position. This plug-in was installed on the VRFS X-Plane instance, which ensures the same outside visual rendering in the conventional simulator as in the virtual environment.

The flight displays of the virtual environment have been connected as well. Therefore, a VNC application was installed on the computer that runs the software for the hardware displays. The captured screen content is streamed into the virtual environment. An average delay of 92 milliseconds at a constant frame rate of 25 frames per second was measured in this particular setup. This satisfies the recommendation of 100 milliseconds and 15 frames per second for primary flight information from the SAE standard Aerospace Recommended Practices ARP 5288 (Society of Automotive Engineers International, 2001). Yet, R. E. Bailey et al. (2005) suggest that this standard is overly generous, i.e. a lower delay and higher frame rate is recommended for real life applications.

All these synchronization efforts create a mixed mock-up environment with the same flight characteristics, the same haptic feedback, the same display content as well as the same outside visual as in the conventional hardware simulator without HMD, as shown in Figure 5.5.

Independent Variable

During the user trial, the simulation environment, virtual reality and conventional, is varied. Thus, the experiment has one within-subject variable, i.e. the simulation environment.

Scenario

A traffic circuit, in no wind conditions, at the airport of Graz (LOWG) starting from a parking position is used as the scenario. After leaving the parking position and take-off from runway 17C[1], two left-hand traffic circuits at an altitude of 2000 feet have to be performed. A touch-and-go, a touchdown without a full stop followed by another take-off has to be conducted in between. The pilots receive pre-recorded audio commands during the scenario. These audio messages (see Figure 5.6a) command the use of the flight control unit (heading bug, altitude bug) and controls in the central pedestal (flaps and parking brake lever). The pilots are asked to conduct these commands rather quickly but not in a rush. Besides these commands, the pilots have to control the attitude, altitude, and heading during the traffic circuit autonomously according to the settings in the flight control unit.

The pilots are asked to report “Abeam Threshold”, the position perpendicular to the runway threshold. After this report, a countdown of 30 seconds is triggered by the experiment supervisor followed by a command to turn left to enter the crosswind leg.

The order of the simulation environment is counterbalanced to avoid artifacts. This means that half of the group starts the traffic circuit in the virtual environment and half of the group starts with the conventional simulator. A familiarization traffic circuit is conducted in every simulation environment before the start of the actual scenario.

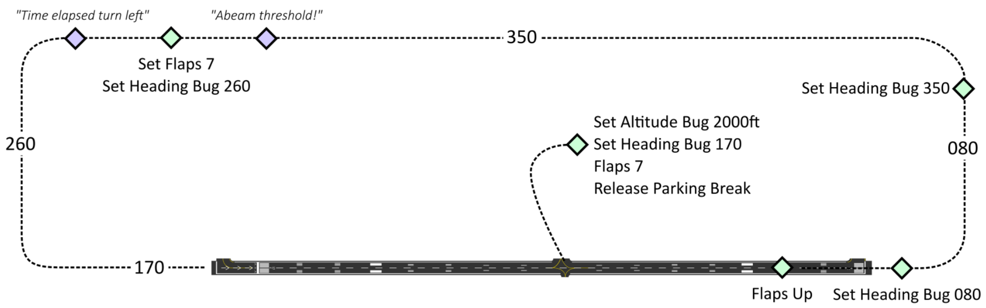
Dependent Measures

Traffic Circuit Performance. In the given scenario, manual flying skills are important to perform the task successfully. The used environment influences the ability to perform the traffic circuit. Hence, the quality, i.e. the performance, of the flying task will differ within the subjects. According to Salmon et al. (2006), the performance of a task is an indicator of the level of situation awareness. The performance is measured with the help of various parameters. Similar to a study by Le Ngoc and Kalawsky (2013), the traffic circuit is divided into flight segments. Several dependent measures are used to quantify the pilot’s performance. Figure 5.6b shows the traffic circuit and the dependent measures that have been used.

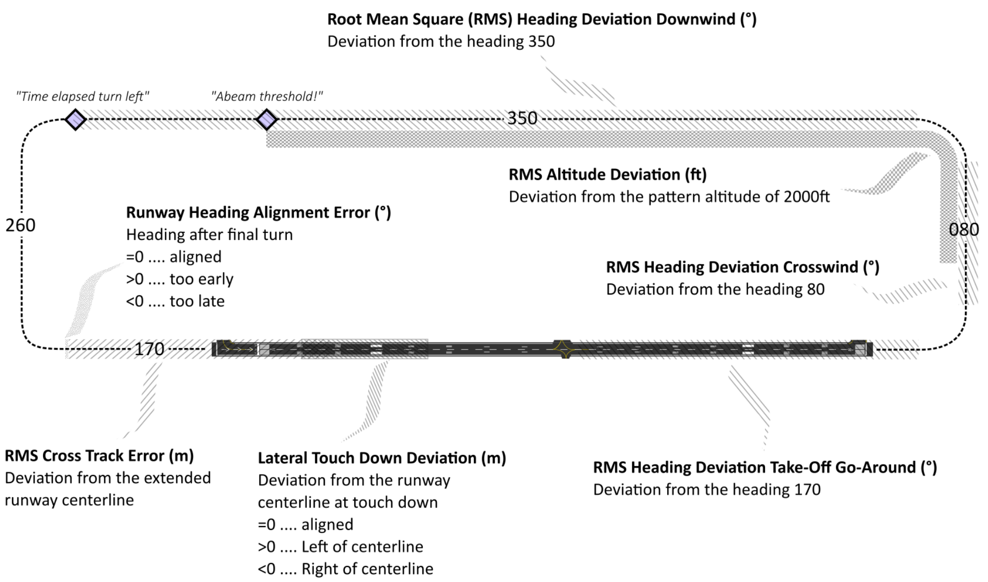
The *Root Mean Square (RMS) Heading Deviation* is measured in three flight segments: After take-off and go-around until the first turn, in the crosswind leg and in the downwind leg. The metric quantifies the deviation from the selected heading during a flight segment. The base leg has not been included, as it is left to the pilots how they fly the approach, i.e. a standard, a short or an extended final approach.

For quantifying the vertical performance, the *Root Mean Square Altitude Deviation* is measured. This metric describes the mean deviation from the selected altitude of 2000 feet, the traffic circuit altitude. The segment starts from reaching the target altitude and ends just before starting the descent.

[1]The number indicates the magnetic heading of the runway. C points to the center runway of the three available runways in Graz.



(a) The traffic circuit with the audio commands (Oberhauser, Dreyer, Braunstingl, et al., 2017b).



(b) The traffic circuit with the dependent measures (Oberhauser, Dreyer, Braunstingl, et al., 2017a).

Figure 5.6.: The traffic circuit is used as a scenario for the simulator trial. The illustration shows the commands that are given to the pilots (top) and the dependent performance measures (bottom).

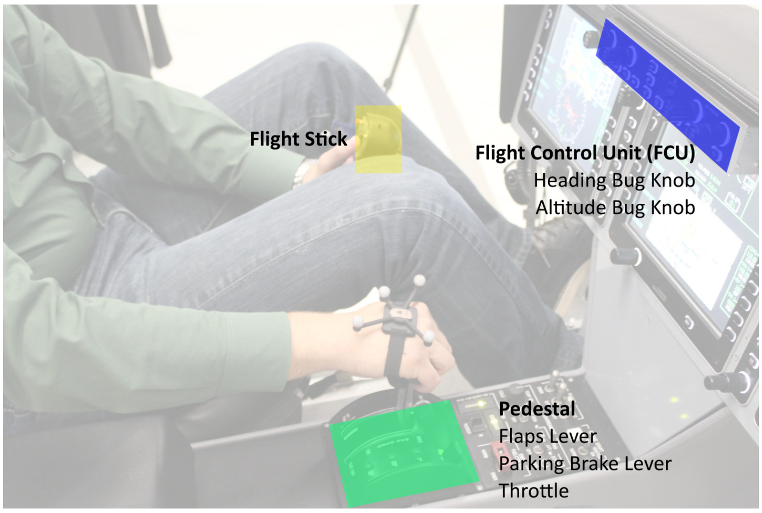


Figure 5.7.: Cockpit elements that are used for the simulator trial (Oberhauser, Dreyer, Braunstingl, et al., 2017b).

After the base leg, pilots have to make a final turn to enter the final approach. This final turn can be too early, right on time or too late. Depending on the timing of the turn, the pilot has to correct the heading to align with the runway. This heading deviation, the *Runway Heading Alignment Error*, is used as a measure of the precision of the final turn.

The ideal flight path in the final approach follows the extended runway centerline. If the pilot deviates from this course, the approach can become unstable. Hence, the *Root Mean Square of the Cross Track Error* (in meters) is a measure of quality of the final approach.

At touch-down, the *Lateral Touchdown Deviation* is measured and analyzed. This value is zero if the pilot lands exactly on the centerline, which is desirable. Deviation from the centerline is considered as a poor performance.

Interaction Performance. In the scenario, audio commands are given to the pilots during flight. These audio commands concern actions in the flight control unit, i.e. the heading and the altitude bug knob as well as the central pedestal, i.e. flaps lever and parking brake lever, as shown in Figure 5.7. The audio commands are brief and unambiguous, for example: “Flaps 7” or “Heading Bug 350”. There are no audio commands for the flight stick and the throttle, as these elements have to be operated continuously.

Several measures are taken to evaluate the interaction performance. As shown in Figure 5.8, it takes a certain time until the action is executed after an audio command. The reaction time of the pilot after the audio command is one part of the total time. It is determined by observing the movement of the hand: The reaction time describes the time span until the hand starts moving towards the control element. The hand

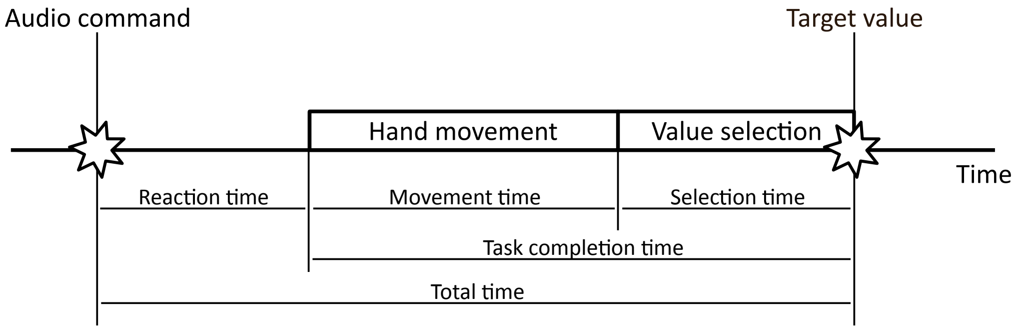


Figure 5.8.: Illustration of the measured times from the initial command to the selected target value (Oberhauser, Dreyer, Braunstingl, et al., 2017b).

movement time is the time span it takes the participant to move his or her hand towards the control element. The value indicates how the position of a control element is perceived and how hard it is to reach.

If the element has continuous input/output, the time for selecting the target value is measured. The selection time is particularly influenced by the interplay of hardware controls and the visualization on the flight display. The Task Completion Time (TCT) takes into account the hand movement and the value selection time. It excludes the reaction time, as this value highly varies between different subjects and depends on the workload level of the current situation.

All these measures are derived automatically from a rosbag that contains data on the initial audio command, the hand position, and the value that is manipulated by the control element. A Python script reads the rosbag after the trials and stores the results in CSV files.

Workload. For measuring the workload, a NASA-TLX questionnaire is handed to the subjects after the trial in each environment as a printout. An objective measure of workload is the heart rate. The heart rate is observed in every trial and is compared between both environments.

Situation Awareness. The traffic circuit performance and the pilot's situation awareness are closely related. If a given flight parameter like heading or altitude cannot be met, the situation awareness, in particular the spatial awareness, is poor. Another metric to measure the spatial awareness was introduced by letting the pilot estimate the "Abeam Threshold" position, i.e. the position perpendicular to the runway threshold. The accuracy of this estimation is used to compare the spatial awareness in this situation between the two environments.

The pilot's head movements are also an indicator for situation awareness. If a pilot is not looking outside, or has extended scanning patterns, the first level of situation awareness, the perception, can suffer. The differences in the scanning behavior in both environments are therefore assessed.

(a) The hardware cockpit (b) The apparatus in the lab environment

Figure 5.9.: The experiment setup used for the virtual interaction trial. The overhead panel and a chair (right) recreate the spatial position of the original cockpit (left).

Simulator Sickness. The Simulator Sickness Questionnaire is used to quantify the simulator sickness in the conventional and the virtual environment. This subjective questionnaire measures simulator sickness by asking participants to rate the intensity of 27 symptoms associated with simulator sickness like fatigue, eye strain or nausea after the scenario (Kennedy et al., 1993).

Participants

28 private and commercial pilots with a mean age of 42.5 years and a minimum of 25 and a maximum of 72 years participated in this trial. Two pilots did not complete the trials due to vision problems and simulator sickness. All participants signed an informed consent form.

Data Analysis

A repeated-measures analysis of variance with two within-subjects factors is conducted: First, the simulation environment (virtual reality versus conventional); second, the trial (first versus second traffic circuit). The analysis of the flight circuit performance data uses absolute values. Alpha is set at .05.

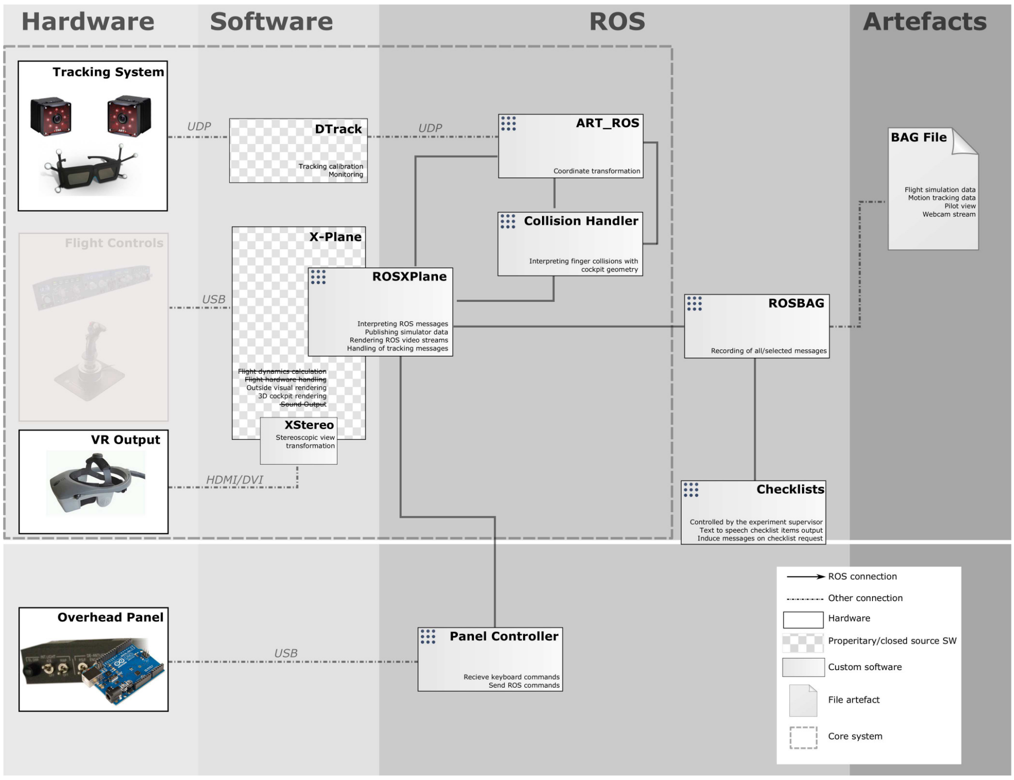


Figure 5.10.: The architecture of the VRFS for the virtual interaction trials. The overhead panel replica is connected via USB and a Python script sends ROS commands to the simulation.

5.1.2. Virtual Interaction Trial

Apparatus

This trial uses a rather simplified setup compared with the simulator trial presented in the previous section. Again, the general aviation cockpit is used for this trial, yet, the hardware simulator located at the Graz University of Technology is not used. Instead, a hardware setup is recreated in the laboratory at Airbus Group Innovations in Munich with only an overhead panel and a seat (see Figure 5.9).

The overhead panel was rebuilt based on the CAD model of the cockpit and is placed at a similar spatial position relative to the pilot. The panel consists of rocker switches that are connected to an Arduino Leonardo, a modular micro-controller platform (Kushner, 2011). The panel is connected via USB and sends keyboard commands. These commands are translated by a Python script to the respective ROS commands, which are then received by X-Plane and manipulate the switches in the virtual cockpit (see Figure 5.10).

Independent Variable

Each pilot has to conduct the trial twice: Once with the overhead panel replica without entering the virtual reality environment and once immersed in the virtual environment without the hardware panel. Thus, the trial has one within-subject variable, i.e. the simulation environment.

Scenario

No operational scenario is used for this trial. The subjects receive pre-recorded audio commands, which are triggered by the experiment supervisor. These audio commands describe buttons in the overhead panel. After receiving a command, the subject has to trigger the respective toggle switch.

Dependent Measures

The interaction performance is measured with the help of the same metrics as shown in Figure 5.8. Yet, only the reaction time and the movement time (i.e. the task completion time) will be distinguished, as the selection time of a toggle switch is negligibly small. Another measure regarding the interaction performance is the error rate. In particular when interacting in a fully virtual environment, it is possible to inadvertently trigger adjacent elements. The number of these erroneous interactions is recorded and analyzed.

Participants

As this trial focuses on interaction and does not include any manual flying tasks, it is not necessary that the participants are licensed pilots. 17 male subjects participated in the trials with a mean age of 34 with a minimum of 24 years and a maximum of 52 years. All participants signed an informed consent form.

Data Analysis

A repeated-measures analysis of variance with one within-subjects factor is conducted: The simulation environment, virtual reality versus conventional. Alpha is set at .05.

In the following sections, the results of the two described trials will be presented. The results are structured with an emphasis on the described dependent measures, not the individual trials.

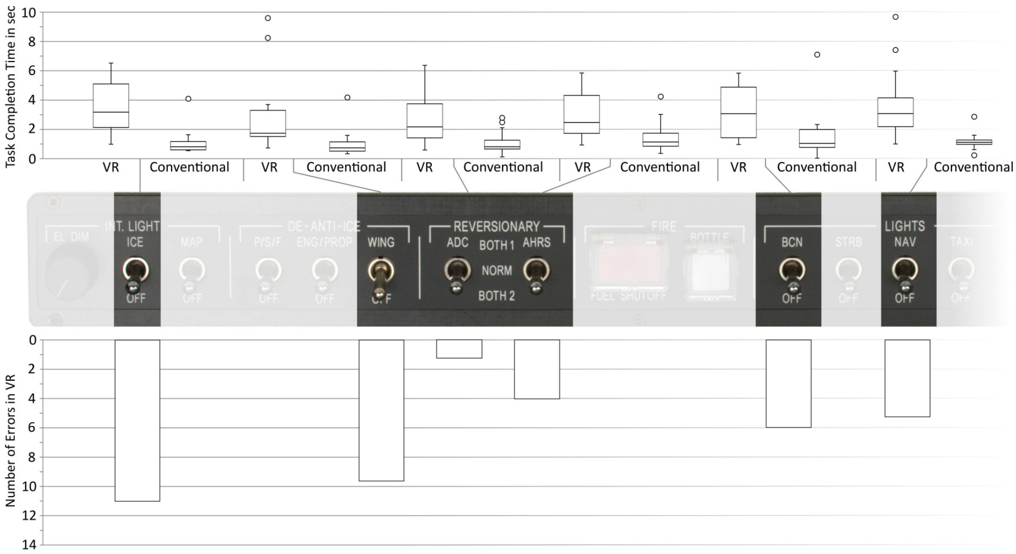


Figure 5.11.: Box plots of the virtual interaction task completion time (top) and the number of erroneous interactions in the virtual environment of all participants (bottom). The task completion time is equivalent with the movement time.

5.2. Cockpit Interaction

In this section, the results of the interaction performance with mixed mock-up and fully virtual control elements will be presented, as well as the interaction with integrated external displays.

5.2.1. Virtual Buttons

The results for the interaction with fully virtual buttons show a significant increase in the task completion time, which, in this case, is equivalent to the movement time for all the switches that have been triggered both scenarios. Table 5.2 shows an overview of the statistical analysis. The mean task completion time with the hardware control elements varies from 1 to 1.532 seconds. The mean task completion time for the fully virtual interaction varies from 2.601 to 3.681 seconds. This increase is also visualized in Figure 5.11 as box plots. The number of erroneous interactions per control element is also visualized in this figure. Erroneous interactions did occur in the virtual environment only.

5.2.2. Hardware Elements

The results for interacting with hardware elements show that the movement time for the interaction with rotary knobs in the FCU in the virtual environment is longer than in the conventional simulator, as shown in Table 5.3. It took between .945 and

Table 5.2.: Analysis of the task completion times of the interaction with fully virtual buttons. The task completion time is equivalent with the movement time.

	Mean	SE	F(1,17)	p	η^2	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
ADC							
VR	2.601	.395	12.456	**.001**	.25	1.975	3.226
Conventional	1.068	.180				.442	1.693
AHRS							
VR	2.790	.356	9.531	**.004**	.19	2.205	3.376
Conventional	1.532	.260				.947	2.118
BCN							
VR	3.399	.446	11.706	**.002**	.23	4.070	5.893
Conventional	1.486	.390				.682	2.289
ICE							
VR	3.596	.445	29.971	**.000004**	.45	2.933	4.258
Conventional	1.071	.202				.409	1.734
NAV							
VR	3.681	.5516	19.592	**.0001**	.35	2.863	4.498
Conventional	1.169	.5516				.352	1.986
WING							
VR	2.801	.593	8.111	**.007**	.173	1.890	3.711
Conventional	1.000	.219				.089	1.910

1.165 seconds longer to reach one of the rotary knobs while a pilot was immersed in virtual reality and flying the aircraft. On ground, i.e. in a low workload situation, it took between 1.113 and 1.681 seconds longer to reach a control element in the FCU while immersed in the virtual environment. The statistical analysis shows that during both, low-workload as well as high-workload situations, this difference is significant with p values well under .05.

The interaction with the flaps in the central pedestal shows that there is a significant difference in the movement time for setting the flaps before take-off (1.01 seconds longer) and in flight for the final configuration (.536 seconds longer). There is no significant difference in setting the flaps to zero after take-off in the two environments.

5.2.3. External Flight Displays

The use of external, streamed flight displays, in particular with the delay of almost 100 milliseconds that come with the use of external sources, has an impact on multiple dependent measures: The spatial situation awareness, and in consequence the flight circuit performance, are influenced by the delay of the virtual displays. Yet, the performance is influenced by other confounding factors as well. A factor directly linked to the display delay is the heading and altitude bug selection time. The pilot can change these values on the FCU but has to monitor the effect on the PFD and MFD displays. According to a model developed by MacKenzie and Ware (1993), visual delay has a negative linear impact on the task performance in two-dimensional interactive systems. In an experimental trial MacKenzie and Ware (1993) showed that this effect is already easily measurable at a delay of 75 milliseconds. Hence, the delay of the flight displays influence the time it takes a pilot to select the target value.

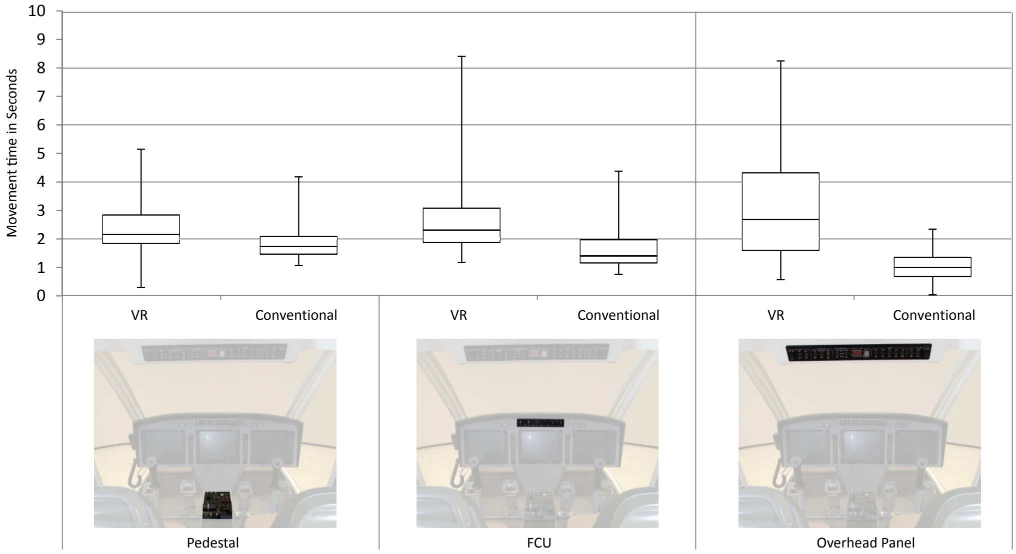


Figure 5.12.: Box plots of the movement time with hardware buttons during the simulator trial.

Table 5.3.: Analysis of the movement times during the simulator trial. Mean differences, standard errors, and 95% confidence intervals refer to differences between the movement times in virtual reality and in the conventional simulation.

Control Element	Mean Difference (SE)	Pearson Coeff. of Correlation r	p	N	Repeated Measures ANOVA F	p	μ^2	95% Confidence Interval Lower Bound	Upper Bound
In Flight									
Heading Bug 080	.945(.204)	.37	.04	23	$F(1,22) = 21.43$	**.0001**	.49	.522	1.369
Heading Bug 350	.957(.199)	−.05	.40	25	$F(1,24) = 23.11$	**.0001**	.49	.546	1.367
Heading Bug 260	1.165(.274)	.48	.006	26	$F(1,25) = 18.13$	**.0001**	.42	.602	1.729
Flaps Up	.863(.603)	.08	.36	23	$F(1,22) = 2.04$	.17	.09	–	–
Flaps Final	.536(.219)	.18	.19	24	$F(1,23) = 6.00$	**.02**	.20	–	–
On Ground									
Heading Bug 170	1.113(.314)	.02	.47	25	$F(1,24) = 12.55$	**.002**	.34	.465	1.762
Altitude Bug 2000	1.681(.460)	−.02	.45	25	$F(1,24) = 13.35$	**.001**	.36	.732	2.631
Flaps Take-off	1.010(.203)	.28	.09	25	$F(1,24) = 24.68$	**.0001**	.51	.590	1.430

Table 5.4.: Analysis of the selection times of the heading and altitude bug during the simulator trials. Mean differences, standard errors, and 95% confidence intervals refer to differences between the selection times in virtual reality and in the conventional simulation.

		Pearson Coeff. of Correlation			Repeated Measures ANOVA			95% Confidence Interval	
Control Element	**Mean Difference (SE)**	**r**	**p**	**N**	**F**	**p**	μ^2	**Lower Bound**	**Upper Bound**
In Flight									
Heading Bug 080	3.199(.789)	.22	.15	24	$F(1,23) = 16.45$	**.0001**	.41	1.568	4.831
Heading Bug 350	2.268(.594)	.40	.02	25	$F(1,24) = 14.58$	.001	.38	1.042	3.493
Heading Bug 260	1.538(.376)	.44	.01	27	$F(1,26) = 16.76$	.0001	.39	.766	2.310
On Ground									
Heading Bug 170	2.549(.788)	.06	.39	25	$F(1,24) = 10.46$	.004	.30	.923	4.175
Altitude Bug 2000	.407(.190)	.47	.009	25	$F(1,24) = 4.60$	.04	.16	.015	.799

The limited field of view of the HMD might also contribute to longer selection times. During the selection process in the virtual environment, pilots cannot monitor the display and the outside visual simultaneously. This leads to head movements to check the outside visual which might prolong the selection process. The head movements and the scanning patterns in the two environments are discussed in detail in section 5.4.1.

Table 5.4 shows the statistical analysis of the selection time. It becomes clear that the time to select a target value is significantly longer in the virtual environment compared with the conventional simulator. The mean difference between the selection time in virtual reality and in the conventional simulation ranged from .407 to 2.549 seconds on the ground, i.e. in low workload conditions, and ranged from 1.538 to 3.199 seconds during flight. However, the selection time on the ground cannot be compared with the selection time in flight as the gap between start and target value differed due to the given scenario (Oberhauser, Dreyer, Braunstingl, et al., 2017b).

5.3. Workload

5.3.1. Mental Workload

The pilots' self-ratings on the mental workload show a significantly higher workload in virtual reality than in the conventional flight simulator. These self-ratings are mental demand, temporal demand, performance, effort, and frustration, as shown in Table 5.5. All these metrics, collected with the NASA-TLX questionnaire, are significant with p-values below the alpha of .05.

5.3.2. Physical Workload

The pilots' self-ratings on physical workload, the NASA-TLX questionnaire on physical demand, shows a significantly higher workload in the virtual environment compared

Table 5.5.: Analysis of workload metrics measured during the simulator trial. Mean differences, standard errors, and 95% confidence intervals refer to differences between the workload in virtual reality and in the conventional simulation.

Metric	Mean Difference (SE)	Pearson Coeff. of Correlation r	p	N	Repeated Measures ANOVA F	p	μ^2	95% Confidence Interval Lower Bound	Upper Bound
Cognitive Workload									
Mental Demand	1.750(.311)	.78	.0001	28	$F(1,27) = 31.61$	**.0001**	.54	1.111	2.389
Temporal Demand	.857(.373)	.59	.001	28	$F(1,27) = 5.26$	**.03**	.16	.091	1.623
Performance	−1.750(.426)	.32	.05	28	$F(1,27) = 16.87$	**.0001**	.39	−2.624	−.876
Effort	2.500(.387)	.41	.02	28	$F(1,27) = 41.81$	**.0001**	.61	1.707	3.293
Frustration	2.214(.533)	.45	.008	28	$F(1,27) = 17.26$	**.0001**	.39	1.212	3.308
Physiological Workload									
Physical Demand	2.357(.458)	.51	.003	28	$F(1,27) = 26.21$	**.0001**	.50	1.418	3.296
Mean Heart Rate	2.407(1.668)	.72	.0001	27	$F(1,26) = 2.08$	.16	.07	–	–

with the conventional simulator ($p < .0001$). However, there are no significant differences in the mean heart rate in both environments.

Another objective measure of physical workload is the head movement, a metric that is related to situation awareness as well and will be discussed in the next section.

5.4. Situation Awareness

5.4.1. Head Movements and Scanning Patterns

The head movements were recorded using the ART tracking system and transformed from quaternions into three rotational degrees of freedom: Pitch, roll, and yaw. Due to the limited field of view of the head-mounted display, the scanning behavior during flight is different in the virtual environment. The pilots can either see the PFD completely or capture the outside visual. Hence, pilots tended to move the head up and down to switch between the head-down instruments and the outside visual. Figure 5.13 shows a typical curve of a pilot's head pitch in the virtual environment and in the conventional simulator. The root mean square of the pilot's head pitch also shows a significant increase, from a mean of 5.027 degrees in the conventional simulator to 6.593 degrees in the virtual environment. The statistical analysis as presented in Table 5.6 shows that this increase is significant ($p < .003$).

No significant difference in the pilot's azimuth (yaw) scanning behavior could be observed. This behavior was expected as in the scenario no extensive horizontal scanning was necessary. A decrease of head roll movements was observed in the virtual environment: A root mean square of 2.462 degrees in the virtual environment compared with a root mean square of 3.474 degrees is a significant difference ($p < .0001$).

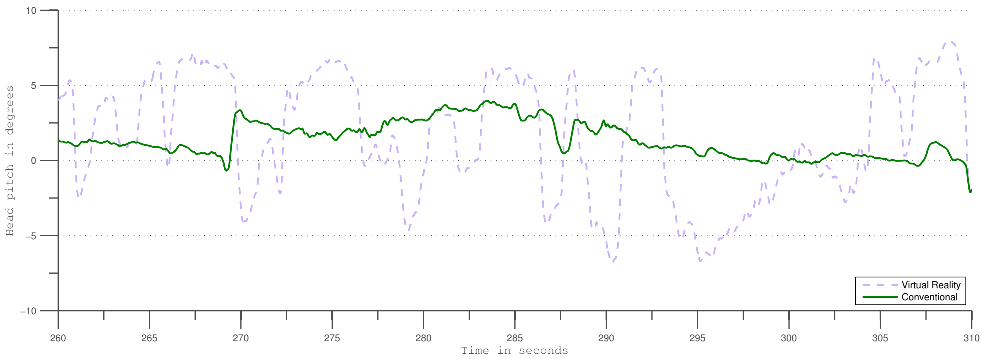


Figure 5.13.: Exemplary plots of the head pitch of one pilot (No. 14) in the conventional and the virtual environment.

Table 5.6.: Statistical analysis of the head movements captured with the head tracking system in both environments. The pitch, yaw, and roll axis is analyzed. All axes are absolute values from zero pitch, yaw and roll.

	Mean	SE	F(1, 25)	p	η^2	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Head Movements							
Pitch							
VR	6.593	.360	9.487	**.003**	.136	5.871	7.314
Conventional	5.027	.360				4.305	5.748
Yaw							
VR	14.164	.456	2.229	.142	.022	13.250	15.079
Conventional	13.202	.456				12.288	14.117
Roll							
VR	2.462	.209	11.721	**.0001**	.166	2.043	2.881
Conventional	3.474	.209				3.054	3.893

5.4.2. Threshold Estimation

The estimation of the threshold position during flight does not show a significant difference between the two environments. The null hypotheses could not be rejected. Hence, in the two environments, the pilots were well aware about their position in the traffic circuit as well as the position of the runway.

5.5. Performance

The results of the statistical analysis of the flight performance are presented in Table 5.7. The root mean square of the heading deviation shows a significant difference between the two environments in all three analyzed legs: Take-off/go-around ($p <$ .001), crosswind ($p <$ .02), and downwind ($p <$ .036). As shown in Figure 5.14, the heading accuracy in the virtual environment is inferior compared with the conventional

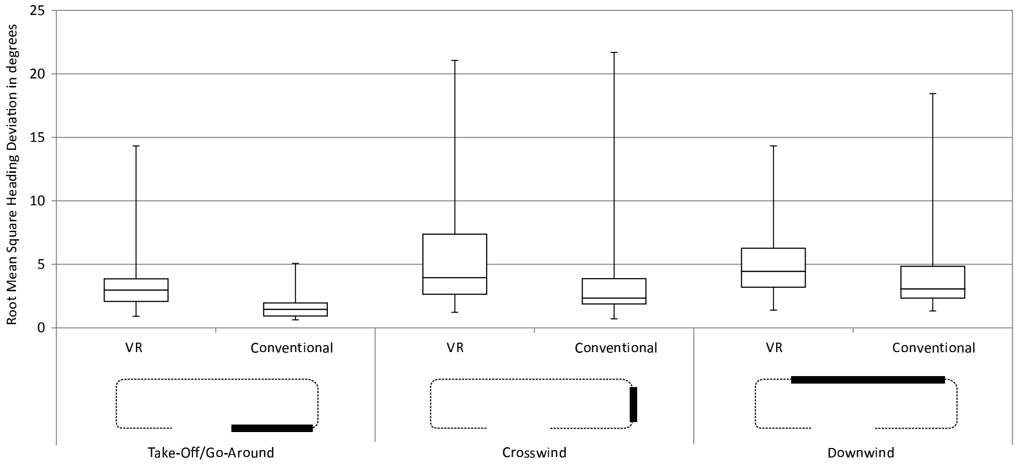


Figure 5.14.: Box plots of the heading deviation of three segments of the traffic circuit during the simulator trial (Oberhauser, Dreyer, Braunstingl, et al., 2017a).

Table 5.7.: Analysis of the flight performance during the traffic circuit.

						95% Confidence Interval	
	Mean	**SE**	**F(1,25)**	**p**	η^2	**Lower Bound**	**Upper Bound**
RMS Heading TO/GA							
VR	3.446	.425	15.21	**.001**	.37	2.572	4.319
Conventional	1.657	.148				1.353	1.961
RMS Heading Crosswind							
VR	5.392	.554	6.16	**.02**	.20	4.251	6.533
Conventional	3.575	.610				2.318	4.831
RMS Heading Downwind							
VR	4.981	.443	4.92	**.036**	.17	4.070	5.893
Conventional	3.912	.614				2.648	5.176
RMS Altitude Deviation							
VR	120.183	16.989	14.64	**.001**	.37	85.194	155.172
Conventional	41.164	77.888				41.164	77.888
Runway Alignement Error							
VR	9.871	1.226	23.89	**.0001**	.49	7.345	12.396
Conventional	3.330	.468				2.367	4.294
RMS Cross Track Error							
VR	37.808	8.284	14.56	**.001**	.37	20.746	54.870
Conventional	4.655	7.115				4.655	7.115
Lateral Touch-Down Dev.							
VR	4.872	.575	16.56	**.0001**	.40	3.687	6.056
Conventional	2.335	.373				1.566	3.103

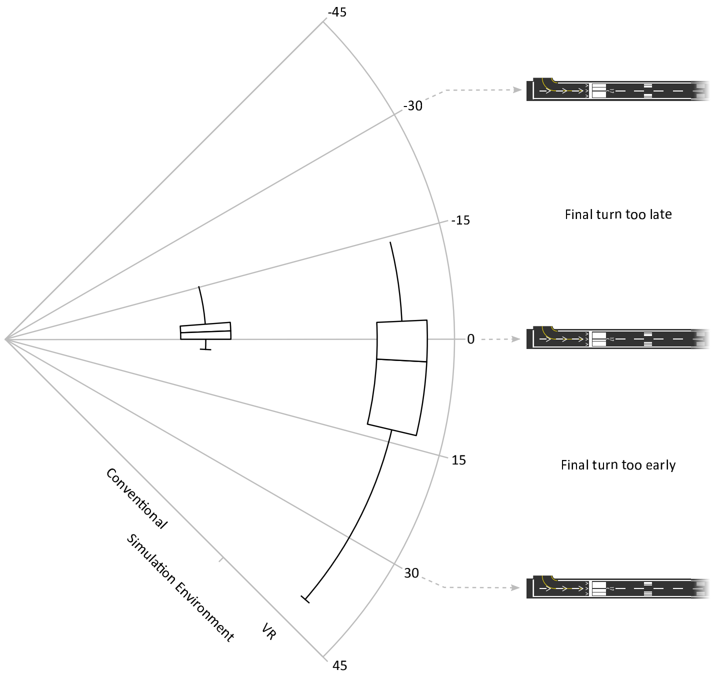


Figure 5.15.: Angular box plots of the runway alignment error after the final turn (Oberhauser, Dreyer, Braunstingl, et al., 2017a).

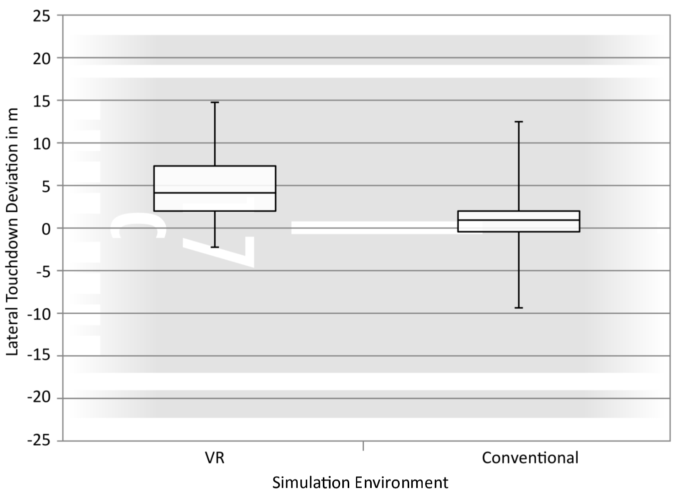


Figure 5.16.: Visualization of the lateral touchdown deviation (Oberhauser, Dreyer, Braunstingl, et al., 2017a).

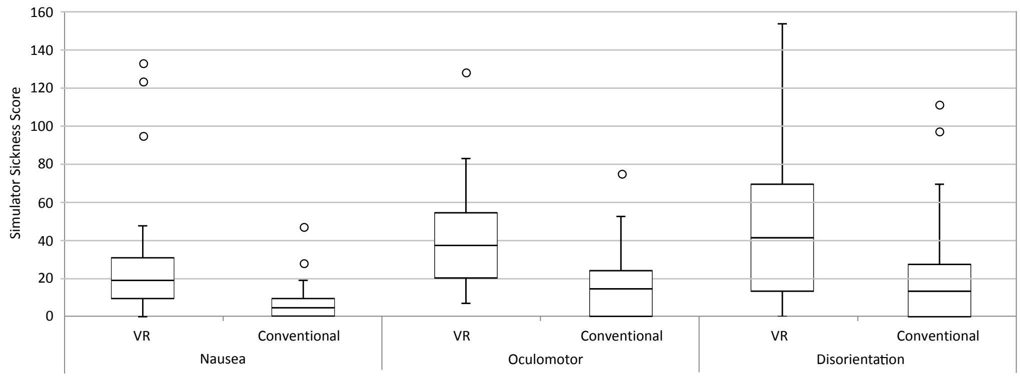


Figure 5.17.: Simulator Sickness Questionnaire results.

simulator for all three legs. The same applies for the root mean square deviation from the pattern altitude, which is significantly higher ($p < .001$) in the virtual environment.

Regarding the runway alignment error, there is a significant difference between the two environments ($p < .0001$). It can be observed that in the virtual environment the runway alignment error is more scattered with a standard deviation of 1.226 degrees compared with a standard deviation of only .468 degrees as shown in Figure 5.15. In addition, there is a tendency to perform the final turn too early, i.e. undershoot in the virtual environment.

After the final turn, the pilots perform the final approach. The root mean square of the cross track error, i.e. the deviation from an extended centerline also shows a significant difference between the two environments ($p < .001$). With a mean of 37.808 meters in the VR compared to a mean of 5.885 meters in the conventional environment, this metric is degraded in the virtual environment, which makes the final approach less stable in the virtual environment.

In the virtual environment, the pilots tended to land left of the centerline. This lateral touchdown deviation from the centerline (see Figure 5.16) shows a significant difference for the landing in the VR and the conventional simulator ($p < .0001$).

5.6. Simulator Sickness

The results of the Simulator Sickness Questionnaire show a degradation in the virtual environment for all three categories that were measured with the questionnaire. As depicted in Figure 5.17, nausea changes from a mean of 8.177 to a mean of 28.961, oculomotor changes from a mean of 19.927 to a mean of 40.931 and disorientation changes from a mean of 23.366 to 53.691 in the virtual environment. The statistical analysis presented in Table 5.8 shows that this change is significant for all three categories.

Table 5.8.: Statistical analysis of the Simulator Sickness Questionnaire.

	Mean	SE	$F(1, 25)$	p	η^2	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
Nausea							
VR	28.961	6.482	9.238	**.004**	.128	19.267	38.655
Conventional	8.177	2.179				−1.517	17.871
Oculomotor							
VR	40.931	5.558	9.443	**.003**	.131	31.241	50.622
Conventional	19.927	3.978				10.237	29.617
Disorientation							
VR	53.691	8.402	9.026	**.004**	.125	39.381	68.001
Conventional	23.366	5.594				9.056	37.676

5.7. Discussion

5.7.1. Interaction

The results show that the overall task completion time with the virtual reality mixed mock-up knobs and levers takes significantly longer than in the conventional simulator, as illustrated in Table 5.9. It took approximately 2.5 to 4.3 seconds longer in the virtual reality environment to set a target value in the FCU than in the conventional environment. In general, it must be considered that the interaction with the control elements in the pedestal is less complex than interacting with elements in the FCU. It is a selection task that relies rather on muscle memory than on hand-eye-coordination. As a consequence, only half of these measurements show a significant, yet, minor degradation of the task selection time of half a second between the VR and the conventional environment.

The overall task completion time is composed of two different variables with one independent influencing factor each. First, the movement time is longer due to confounding factors of the virtual reality environment such as an artificial spatial vision, a limited field of view, and inaccuracies in the virtual hand model. Second, the selection time in particular for the rotary knobs is degraded due to delays of the displays, which are streamed into the virtual environment; a factor specific to the implementation of the virtual cockpit.

A more generic metric, which is only influenced by the the virtual environment and which can be transferred to other cockpits as well, is the movement time. For the FCU buttons, it took pilots approximately one second longer to reach the control element. From a mean of 1.6 to a mean of 2.6 seconds of all interactions with the FCU, this is an increase of approximately 60 percent. For the fully virtual buttons in the overhead panel, the time to reach the correct control element increased from a mean of 1.2 seconds to 3.1 seconds: An increase of almost 160 percent.

These longer task completion times have to be considered when designing experiments in a virtual environment. It is important to keep in mind that a time critical sequence of interactions can cause problems when completing a task in the virtual environment. In particular the interaction with fully virtual elements does not only take considerably longer in VR, it is also prone to error. As shown in Figure 5.11, some users will

Table 5.9.: Analysis of the task completion times of the hardware buttons during the simulator trial. Mean differences, standard errors, and 95% confidence intervals refer to differences between the task completion time in virtual reality and in the conventional simulation.

		Pearson Coeff. of Correlation			Repeated Measures ANOVA			95% Confidence Interval	
Control Element	**Mean Difference (SE)**	**r**	**p**	**N**	**F**	**p**	μ^2	**Lower Bound**	**Upper Bound**
In Flight									
HDG Bug 080	4.251(.915)	.16	.22	24	$F(1,23) = 21.57$	**.0001**	.48	2.358	6.145
	3.199(.789)	.53	.002	26	$F(1,25) = 10.47$	**.003**	.30	.771	3.474
HDG Bug 350	3.224(.604)	.44	.01	25	$F(1,24) = 28.45$	**.0001**	.54	1.977	4.471
	2.434(.550)	.16	.21	27	$F(1,24) = 14.58$	.001	.38	1.042	3.493
HDG Bug 260	2.525(.537)	.41	.02	27	$F(1,26) = 22.10$	**.0001**	.46	1.421	3.629
	2.429(.373)	.51	.004	26	$F(1,25) = 42.31$	**.0001**	.63	1.660	3.198
Flaps Up	1.147(.660)	.04	.43	23	$F(1,22) = 3.02$	.09	.12	−.222	2.516
	.532(.251)	−.18	.19	26	$F(1,25) = 4.48$	**.04**	.15	.015	1.050
Flaps Final	.538(.218)	.19	.80	24	$F(1,23) = 6.08$	**.02**	.20	.087	.989
	.241(.136)	.25	.10	26	$F(1,25) = 3.14$	.09	.11	−.039	.522
On Ground									
HDG Bug 170	3.663(.913)	.02	.46	25	$F(1,24) = 16.08$	**.001**	.40	1.778	5.547
ALT Bug 2000	2.09(.490)	.16	.22	25	$F(1,24) = 18.13$	**.0001**	.43	1.076	3.100
Flaps Take-off	1.074(.284)	.00	.50	25	$F(1,24) = 14.33$	**.001**	.37	.488	1.659

accidentally activate surrounding virtual buttons if they can be activated and are close by. The scenario, or the cockpit functionality, has to be designed in a way that erroneous interactions can be tolerable or that they are ruled out due to the design of the cockpit.

5.7.2. Workload

The workload, both physical and mental, increases in the virtual environment. It was measured with the subjective NASA-TLX questionnaire.

The physical workload is influenced mainly by the weight and ergonomics of the head-mounted display: 11 out of 27 participants complained about the ergonomic limitations of the used head-mounted display. Another contributing factor to the physical workload is a significant increase in head pitch movements in VR due to the limited field of view. In addition, the need to keep the focus on the two displays directly in front of the eyes leads to eye strain, which cannot be relieved while being immersed in the virtual environment.

The additional mental workload might be attributed to a more difficult process of information gathering due to the low field of view, limitations in the visual appearance of the environment, and a more cumbersome interaction with the cockpit. None of the pilots had prior experience with VR systems. It can be assumed that the level of mental workload will be reduced if pilots get used to this type of environment.

5.7.3. Situation Awareness

The limited field of view of the HMD is the main driver for limiting the situation awareness in the virtual environment. The lateral and vertical field of view limitations have two respective effects when flying in the simulation: The lateral field of view limitation inhibits peripheral stimuli, which are important for the perception and control of the attitude (Hosman and Stassen, 1999). The vertical field of view limitation makes it difficult to simultaneously perceive the instruments and the outside visual. Due to this limitation, the eye scanning pattern between the head-up and head-down areas has to be substituted by head movements. This changes the familiar scanning pattern and has a negative effect on information gathering, e.g. the outside visual cannot be observed while looking at the instrument panel.

Another measure that was investigated in this trial is the threshold estimation. It showed that there was no difference in the two environments. Hence, pilots were aware of their position in the traffic circuit at any time. The wider field of regard in the virtual environment did not help to enhance this estimation.

5.7.4. Performance

The participants were able to safely and reliably complete the flight task after a short acclimatization phase in the virtual environment. Yet, the limitations with regard to the functional fidelity and task fidelity, the higher workload levels, and the lower situation awareness had a significant impact on the flight performance.

The higher physical and especially the mental workload have a negative impact on all performance metrics in the virtual environment. The degraded situation awareness also influences the performance metrics: The ability to control the flight parameters "heading" and "altitude" where significantly degraded, especially due to the limited vertical field of view and the changed scanning behavior. The limited lateral field of view caused pilots to undershoot the final turn as the spatial awareness, i.e. the perception of the position of the runway, was degraded. This factor also contributes to the rather unstable final approach tracks that have been observed. Here, in particular the missing peripheral stimuli and their effect on the vestibular system are should be mentioned.

5.7.5. Simulator Sickness

The simulator sickness ratings show a significant degradation in all three categories nausea, oculomotor, and disorientation. The particular symptoms are influenced by various factors. The symptoms might be caused especially by a disparity of the virtual reality representation and head movements, i.e. the delay of the system. In this experiment, the external tracking system has a system latency of approximately 17 milliseconds, the scene takes 16.67 milliseconds to render and it takes the same time to present the complete image to the user through the HMD. This leads to an overall delay from a head movement to the corresponding visual representation of at least

50 milliseconds. This is well above the 20 milliseconds threshold under which a head tracking delay is believed to be imperceptible and is a magnitude that, according to Carmack (2013), "... will feel responsive, but still subtly lagging" (para. 7). The weight and ergonomic deficiencies of the HMD might also be contributing factors. One participant experienced simulator sickness at a level so severe that the experiment had to be aborted.

5.8. Conclusion

In this chapter, the usability and fidelity of some aspects of the Virtual Reality Flight Simulator were evaluated. The psychological-cognitive fidelity is degraded in the virtual environment, a fact that leads to an adapted scanning and interaction behavior and increases mental workload. The task fidelity, i.e. the flight performance as well as the interaction performance, was significantly degraded in the virtual environment as well. Yet, the pilots were able to complete the realistic flight task successfully. These confounding factors have to be considered in the experiment design when performing research in the VRFS. Therefore, these factors will be taken into account in the design recommendations presented in Chapter 7.

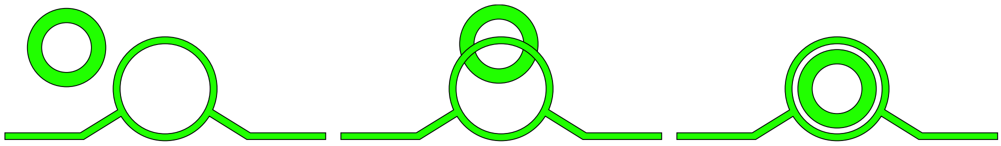
6. User Studies

In this chapter, four user studies will be presented that demonstrate the capabilities of the Virtual Reality Flight Simulator for the development and evaluation of HMI components in the flight deck. Deaton and Morrison (2009) presented a classification of human factors studies with different levels of technology maturity and evaluation goals. Each user study presented here represents one type of study from this classification: Basic research, applied research, advanced development, and engineering development. Table 6.1 gives an overview of the presented studies, the integrated technology as well as the used scenarios.

The goal of this chapter is to focus on the demonstration of the capabilities of the VRFS in HMI research and development projects, not to present the studies in all detail. Each study is reported following a common scheme with the sections Introduction, Methodology, Results, and Discussion. In the Discussion section of each study, implications on the usability and validity of the simulation environment will be addressed as well as the results from the HMI component evaluation itself. Different features have been used in the four studies that stem from the modeled use cases presented in Section 3.1. An overview and a thorough discussion of these studies, with an emphasis on the integrated prototypes and human factors methods based on the practical application will be presented at the end of this chapter. Parts of this chapter have been published in conference proceedings and as a journal publication (Oberhauser and Dreyer, 2017; Oberhauser, Dreyer, Mamessier, et al., 2015). A preliminary version of study 1, the head-up display study and study 3, the system management evaluation, have been published in conference proceedings as well (Dreyer and Oberhauser, 2016a; Dreyer, Oberhauser, and Bandow, 2014).

Table 6.1.: An overview of the presented user studies, the type of research, the integrated technology, and the selected scenario.

Study	Type of Study	Technology	Scenario
(1) Head-up display symbology evaluation	Applied research	Virtual demonstrator	Realistic low-visibility approach scenario
(2) Peripheral horizon display evaluation	Basic research	Early research prototype	Simplified, non operational flying task
(3) System management evaluation	Advanced development	Hardware prototype	Complex, realistic multi-system failure scenario
(4) Head-worn device evaluation	Engineering development	Mature technology prototype	Simplified, non operational flying task, semi-realistic ILS approach



(a) Low and too far right.
(b) Too low.
(c) Optimal flight path.

Figure 6.1.: The head-up guidance system. Three different positions of the guidance cue are visualized. The aircraft is deviated from the optimal approach flight path (left and center). The Flight Path Vector (FPV) and the guidance cue are aligned and the aircraft is on an optimal path (right).

6.1. Head-up Display Symbology Evaluation (Study 1)

6.1.1. Introduction

In this study, a novel Head-up Display (HUD) symbology is developed and evaluated with the help of the VRFS. The idea behind this system is to increase situation awareness, in particular during low-visibility approaches and decrease the pilots' mental workload. This study was conducted as part of the European Union's Seventh Framework Programme (FP 7) project All Condition Operations and Innovative Cockpit Infrastructure (ALICIA) with the goal to provide a common cockpit architecture for fixed and rotary wing aircraft and to develop technologies that enhance safety especially during low-visibility operations (ALICIA, 2014). A preliminary version of this study has been published by Dreyer, Oberhauser, and Bandow (2014).

Head-up Displays

Head-up displays are a technology to superimpose information about the attitude and flight parameters as well as system information in the pilots' head-up field of vision. They were first, and today are foremost, used in military aviation (Moir, Seabridge, and Jukes, 2006, p. 411). Yet, head-up displays are also available for the civil flight deck. According to Wickens, Ververs, and Fadden (2004), head-up displays serve three major goals:

1. Reducing the amount of visual scanning between the outside visual and the head-down instrumentation;
2. Reducing the need for re-accommodation, i.e. changing the focus of the eyes from the outside visual at optical infinity to near head-down instrument panels;
3. Presenting conformal information about outside visual features, although they are not visible, e.g. in low-visibility conditions.

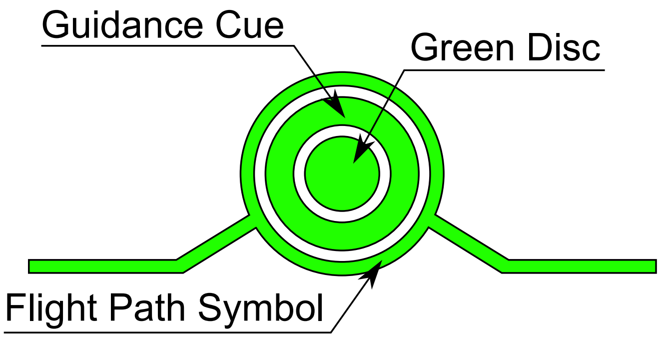


Figure 6.2.: The green disc concept. A green disc appears in the guidance cue if all systems are in the final configuration (Dreyer, Oberhauser, and Bandow, 2014).

The Head-up Guidance System

In this study, a head-up display with a Head-up Guidance System (HUGS) is used as a baseline. HUGSs provide intuitive information on the optimal flight path and energy management (Rockwell Collins, 2016). In this system, a guidance cue in the form of a circle represents the ideal flight path. The pilot has to match the aircraft's flight path, i.e. the flight path vector symbol with the guidance cue as shown in Figure 6.1.

According to studies conducted by Arbon et al. (1991) and Vandel and Weener (2009), head-up displays equipped with a HUGS would have had a positive influence on the outcome of 31 % to 38 % of all aircraft accidents since 1959. These studies are based on subjective assessments of historical aircraft accident reports and therefore have to be treated with caution regarding their validity. A comprehensive study based on a user trial with 60 airline crews conducted by Bandow (2006) suggests that head-up guidance systems decrease workload and increase situation awareness during low-visibility approaches.

The Green Disc Concept

During an approach, no matter if a HUD is used or not, the PF has to cross check information about final checklist items in the head-down area. For a Canadair Regional Jet CRJ-200, the aircraft model that is used for this study, these are: "Flaps: 45 Degrees", "Thrust Reverser: Armed" and "Landing Gear: Three Green". These checklist items are called out by the pilot not flying and have to be visually checked and repeated by the pilot flying. This requires re-accommodation from the outside visual and an extended visual scanning pattern which contradicts one of the benefits of a HUD in this critical phase.

In order to eliminate the operational demand for head-down checks during approach, the so-called *Green Disc Concept* was developed. This concept integrates the system states of the final checklist into the HUD symbology by adding a green disc to the guidance cue as shown in Figure 6.2. This symbol appears if the final configuration is met; and it is only visible if the radar altitude is between 2000 feet and 500 feet, the altitude at which the final check is usually conducted. This allows the pilot flying to remain head-up and confirm the single final checklist item "All Green". A (red) flashing disc shows that one or more criteria for the final configuration are not met and that the approach cannot be continued.

Hypotheses

The proposed Green Disc Concept will be evaluated in a low-visibility approach. With the introduction of this system to a HUGS, several cognitive and operational benefits are being expected. The hypotheses for this experiment are:

1. The Green Disc Concept increases the head-up time during approach compared with a conventional HUD symbology.
2. The Green Disc Concept increases situation awareness during approach compared with a conventional HUD symbology.
3. The Green Disc Concept decreases mental and physical workload during approach compared with a conventional HUD symbology.

6.1.2. Methodology

Prototype Integration

The approach trial with the Green Disc Concept is conducted with a setup similar to the typical setup that has been described in Chapter 4. A head-up display was integrated into the virtual cockpit of the VRFS. The symbology of this HUD is inspired by a HUD by Rockwell Collins that features a HUGS (HGS Model 2100). As this system is available for Canadair Regional Jets, a CRJ-200 flight model was used for the trials.

The implementation and integration of the head-up display was conducted by leveraging the animation features of X-Plane. The symbols of the HUD were modeled as 3D objects and integrated into the 3D cockpit. Every symbol was attached to an X-Plane dataref as presented in Section 4.2.2 on page 44. To create the illusion of collimation, i.e. a focal plane at infinity, the 3D symbols have been moved ten meters in front of the combiner in the virtual cockpit.

In a cockpit with a head-up display, the pilot has to adjust the seat position to be able to see the full head-up display symbology; the pilot's head has to be within the so-called pilot eye box. The effect of moving outside the pilot eye box is emulated with 3D geometry that masks the visibility of the HUD at its edges, an implementation similar to the one presented by Rollon (2008). Additional symbols that could not be implemented with basic animations have been implemented with the help of a plug-in with OpenGL calls. Figure 6.4 shows the architecture of the implemented HUD in the virtual cockpit with the Green Disc Concept.

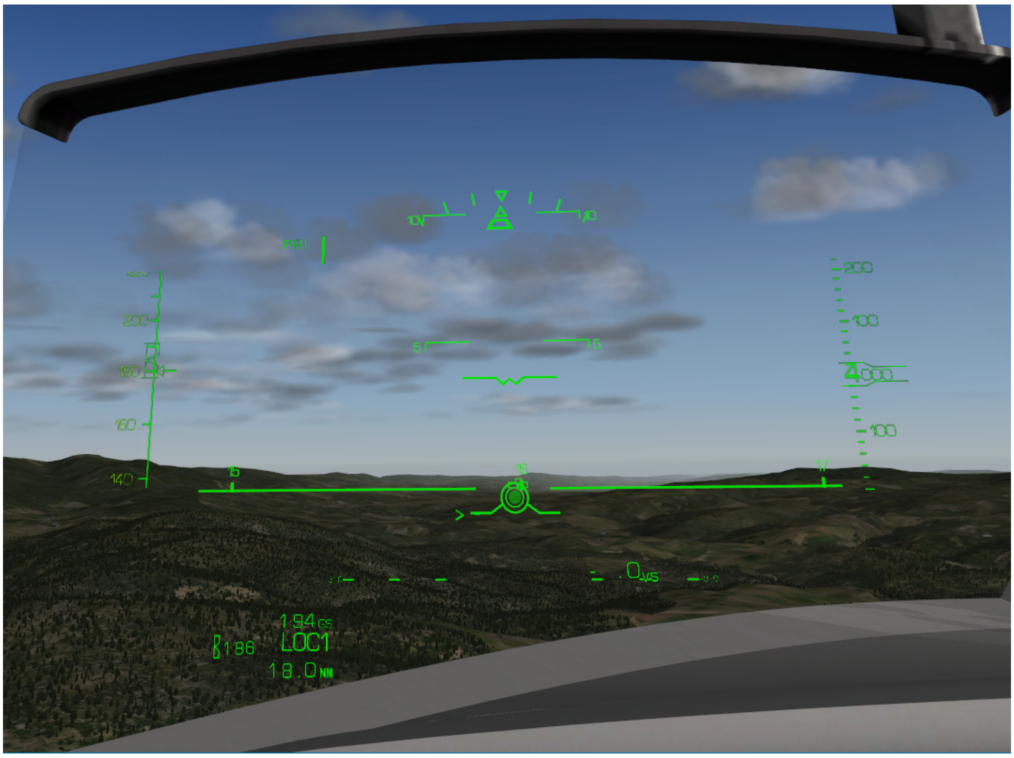


Figure 6.3.: The implemented head-up display with the green disc concept. The HUD is in primary mode with the airspeed and altitude tape visible.

Scenario

As a scenario for the Green Disc Concept trials, an approach to Clermont Ferrand Airport (ICAO code LFLC) towards Runway (RWY) 26 is conducted as shown in the approach chart in Figure 6.5. The approach takes place under low-visibility conditions (CATIIIa[1]) with a Decision Height (DH)[2] of 50 feet. The scenario starts at the Initial Approach Fix (IAF) RIMOR with the autopilot activated and programmed to catch the signal of the ILS of LFLC RWY 26. As the simulated aircraft does not have an autothrottle system, the pilot has to control the engine thrust in all phases of flight. After the autopilot has been set to the Approach (APPR) mode, the aircraft automatically establishes the final track as given by the instrument landing system. After that the pilot has to deactivate the autopilot and use manual control in combination with the indication from the HUGS to continue the approach. During the approach, the pilot flying has to command the pilot not flying to establish the final configuration. Eventually, the aircraft reaches the decision height of 50 ft without visual contact of the

[1] There are different categories for the Instrument Landing System with different minimum visibility requirements: CATI, CATII and the most demanding categories CATIIIa, CATIIIb and CATIIIc.

[2] At the decision height, the pilot has to have visual contact with the runway. Otherwise, a go-around maneuver has to be performed.

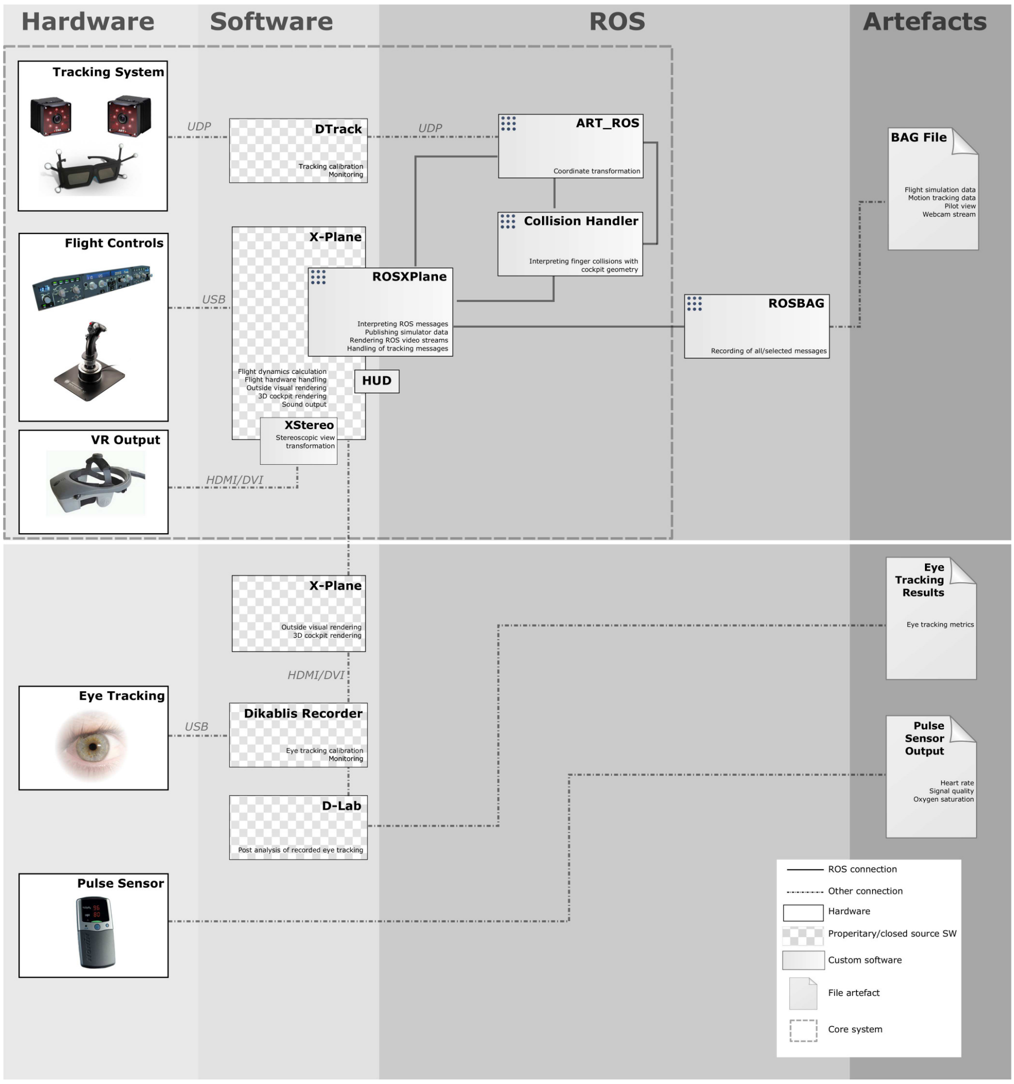


Figure 6.4.: The architecture of the VRFS for the head-up display evaluations. A pulse oximeter is used to gather physiological data and an eye tracking system is integrated to analyze the eye scanning behavior.

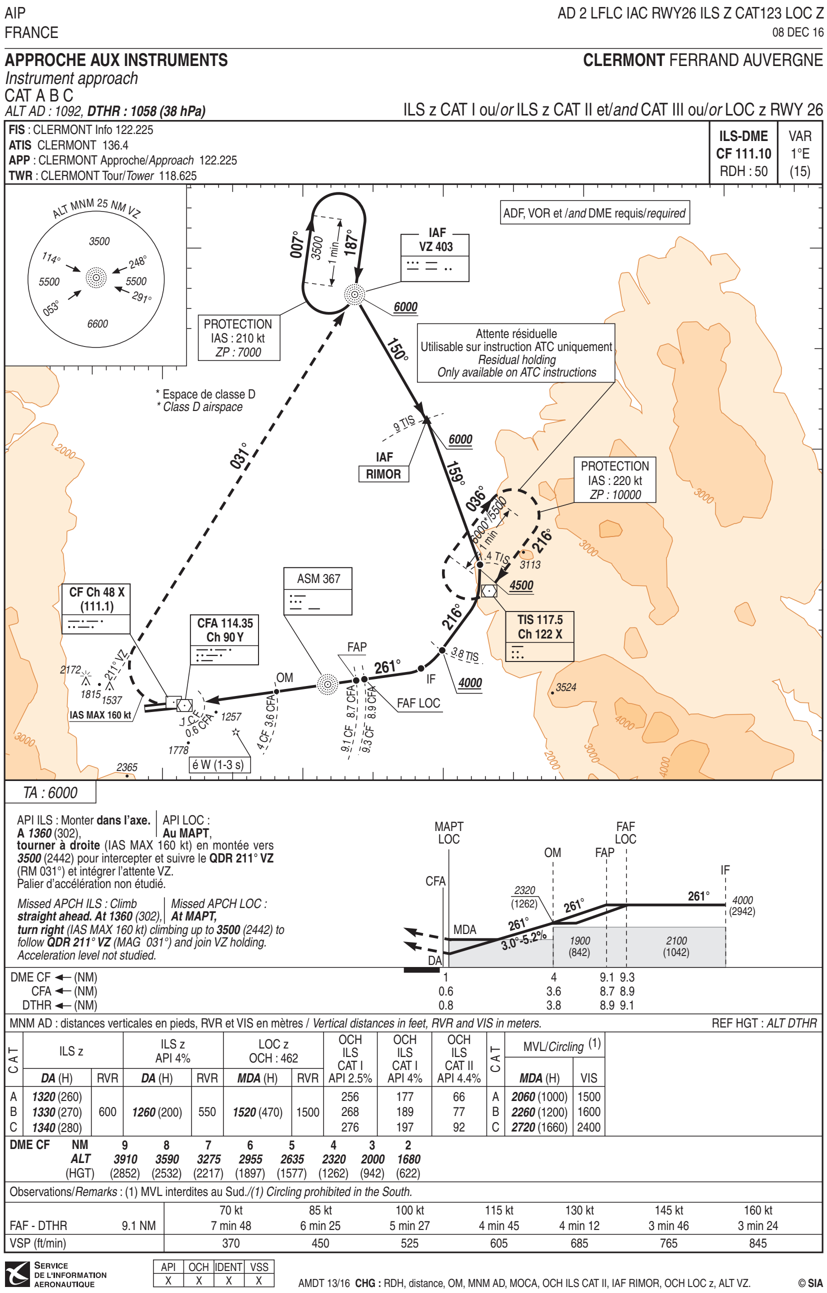


Figure 6.5.: The approach chart of Clermont Ferrand (LFLC) Runway 26 (Service de l'Information Aeronautique, 2016).

runway. This should lead to a go-around with a missed approach procedure heading back to the IAF RIMOR.

One pilot, immersed in the virtual environment, participates in each scenario. The experiment supervisor, who does not wear an HMD, plays the role of the pilot not flying. With hardware controls only accessible to the PNF, he or she is able to conduct the procedural actions that are necessary to reach the final configuration like lowering the landing gear or setting the flaps.

For familiarization purposes, a similar approach under Ceiling and Visibility OK (CAVOK) conditions, without the Green Disc Concept, is conducted before the actual trials.

Dependent Measures

Head-up Time. The head-up time during the low-visibility approach, which includes the window area and the FCU, is observed using the integrated eye tracking system.

Workload and Situation Awareness. Two questionnaires are handed to the pilots after each scenario: The quantitative NASA-TLX questionnaire to evaluate mental and physical workload and the quantitative SART questionnaire for measuring situation awareness.

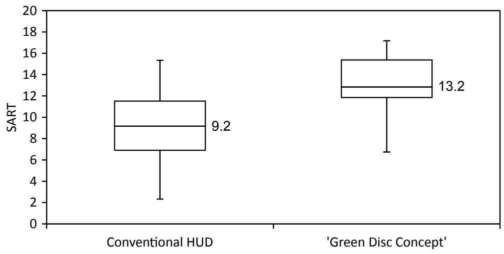
Heart Rate. A pulse oximeter (Nonin Medical, Inc., Model 2500) is used to capture data on the pilots' heart rate. Data gathered with this device can be stored and transferred to a desktop computer using the nVISION data management software. As the pulse oximeter is not connected to the simulation framework, the beginning and the start time of each scenario must be synchronized manually. Hence, the start and the end time of each scenario has to be recorded.

Independent Variable

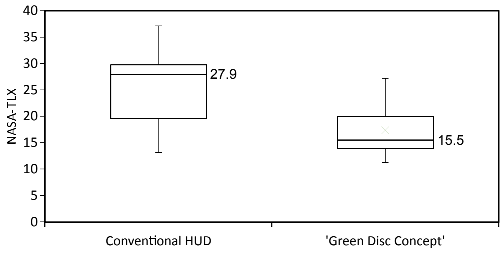
The experiment has one within-subject variable: The type of HUD. Two variants are tested: The conventional HUD and the HUD with the Green Disc Concept. Apart from the HUD symbology, the two cockpits are identical.

Participants

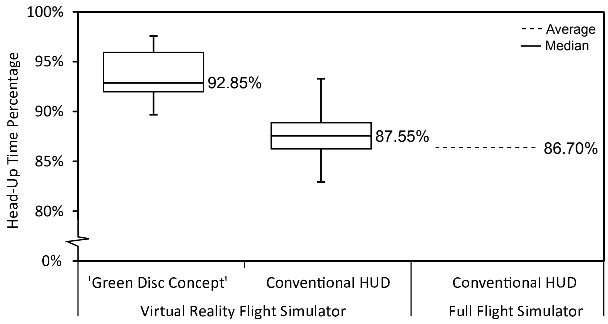
In the Green Disc Concept trials, the participation of professional pilots is necessary. Eleven male commercial airline pilots from several German airlines participated in the trials with a mean age of 49 years and an average experience of 10,200 flight hours. The participation of operational airline pilots, in contrast to test pilots, provides feedback with a focus on normal operations (Reuzeau and Nibbelke, 2004).



(a) Situation Awareness Rating Technique



(b) NASA Task Load Index



(c) Head-up time.

Figure 6.6.: Box plots of the SART and NASA-TLX ratings of the HUD trials (top) and the head-up time (bottom). The box plot of the head-up time also includes the mean head-up time of a preceding trial in a conventional flight simulator (Oberhauser, Dreyer, Mamessier, et al., 2015).

Data Analysis

A Wilcoxon signed-rank test for paired samples was conducted for the analysis of the questionnaires and the head-up time. Alpha was set at .05. The eye tracking data was processed and analyzed using D-Lab, the analysis tool of the used eye tracking system (Ergoneers GmbH, 2011a).

6.1.3. Results

Workload and Situation Awareness Questionnaires

The analysis of the subjective data of the NASA-TLX (see Figure 6.6b) shows a decrease of the physical and mental workload with a mean from 25.3 for the conventional HUD to a mean of 17.4 for the Green Disc Concept. The statistical analysis depicted in Table 6.2 shows that this decrease is significant ($p = .00335$) with a large effect ($w = .63$).

The analysis of the situation awareness that was conducted with the SART questionnaire revealed that there is an increase of SA from a mean of 9.2 to 13.2 when comparing the two concepts (see 6.6a). The statistic analysis showed that this increase is significant ($p = .00335$) with a large effect ($w = .63$).

Table 6.2.: Analysis of the SART and NASA-TLX questionnaires for n=11 subjects, comparing the conventional HUD and the Green Disc Concept as well as the head-up percentage based on eye-tracking data with n=10 participants. The eye-tracking data of one participant could not be used due to a calibration error.

			Wilcoxon Signed-rank Test for Paired Samples ($\alpha = .05, n = 11, df = 1$)		
HUD	**Mean**	**SE**	**Z**	**p**	**w**
SART					
Conventional HUD	9.2	1.243	−2.934	**.00335**	.6255
Green Disc Concept	13.2	.8958			
NASA-TLX					
Conventional HUD	25.310	2.248	−2.934	**.00335**	.6255
Green Disc Concept	17.351	1.5			
Head-up time in %					
Conventional HUD	87.48	.00946	−2.803	**.00506**	.6268
Green Disc Concept	93.45	.00856			

Eye Tracking

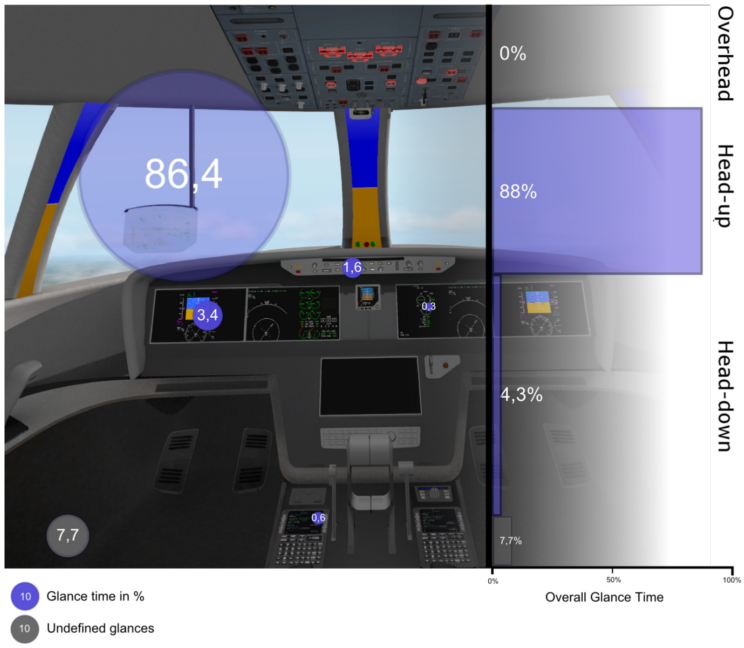
All eye-gaze dwell times are presented as a percentage relative to the time of the complete scenario from the IAF RIMOR to the end of the scenario. For a semi-automatic analysis of the eye tracking data, the cockpit was divided into areas of interest. Figure 6.7 shows the dwell times for these AOIs. The mean head-up time (see Figure 6.6c) increases with the Green Disc Concept from 87 percent to 93 percent. The mean head-down time was reduced from 4.3 percent to 1.5 percent. Figure 6.7 shows a detailed visualization of the eye gaze dwell times on various AOIs.

The statistical analysis (see Table 6.2) shows that the increase of head-up time with the Green Disc Concept is significant ($p < .03789$) with a large effect ($w = .63$).

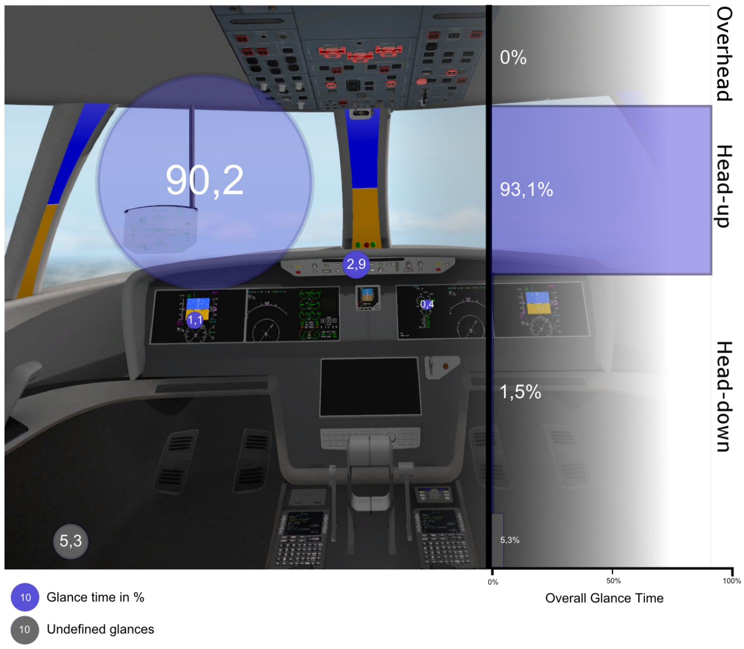
6.1.4. Discussion

The eye tracking analysis shows that the head-up time can be increased significantly with the novel symbology. This confirms the first hypothesis. With the new head-up display there is no operational need for gathering information from the head-down area during the final approach. This would lead to a head-up percentage of 100 %. Still, some pilots cross-checked the HUD indications with the information provided in the PFD or performed other non-procedural actions. The results also confirm the hypothesis that the Green Disc Concept has a significantly positive influence on situation awareness and workload.

The user study demonstrates the straightforward integration of a head-up display into a holistic flight simulation environment. The use of HUD technologies in conventional hardware simulators usually involves additional efforts. Using an actual device can lead to high costs or to constraints if the device is emulated in the outside visual. In a virtual environment though, head-up displays can be simulated with parallax effects and an illusion of collimation.



(a) The conventional HUD



(b) The Green Disc Concept

Figure 6.7.: The analysis of the areas of interest from the eye tracking data. The cockpit is divided into an overhead, head-up, and head-down area. The area of the blue circles represents the gaze time on each area. The gray circle represents undefined gazes that could not be tracked by the system.

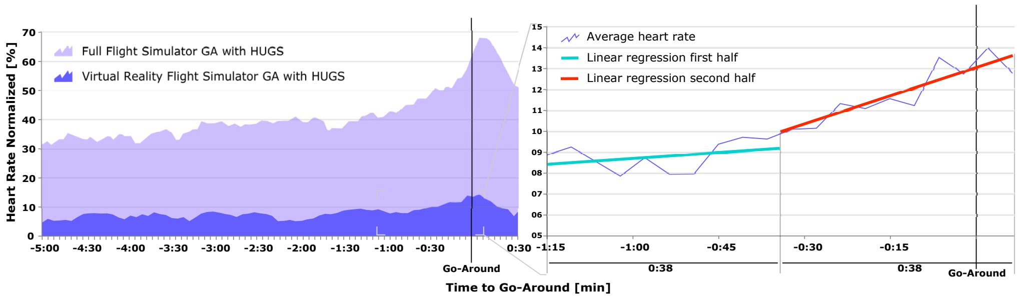


Figure 6.8.: Comparison of the heart rate in a conventional simulator and the VRFS (left) and a discontinuity regression analysis of the heart-rate in the virtual environment at go-around (right) (Oberhauser, Dreyer, Mamessier, et al., 2015).

As a similar trial was conducted by Bandow (2006), indications on the validity of the VRFS can be gathered. This trial in a conventional full flight simulator used a similar approach/go-around scenario. The eye scanning behavior of the pilots in the conventional environment has also been observed with an eye tracking system. Figure 6.6c shows that the head-up time in the VRFS with the conventional HUD (87.55 %) is similar to the head-up time in the hardware simulator with the conventional HUD (86.70 %). The observed eye scanning patterns (see Figure 6.9) are as anticipated by the author based on previous research (Bandow, 2006, p. 169).

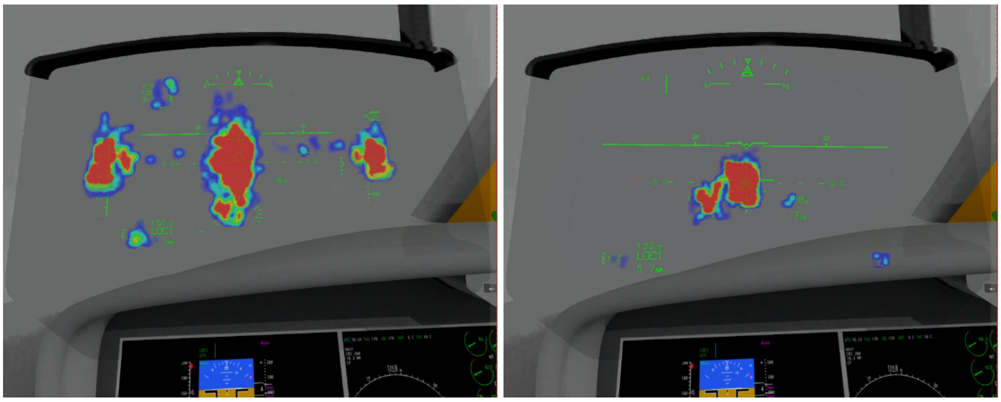
Another parameter that was measured in both the virtual environment and the conventional simulator is the heart rate: Figure 6.8 shows the average heart rate of all participants of both scenarios synchronized at the time of the go-around. The data was normalized based on the resting heart rate (Luczak, 1997, p. 72). A peak in the heart rate can be observed in both environments at the go-around; yet, the heart rate in the conventional environment is higher by the factor three to four. A discontinuity regression analysis (Shadish, Cook, and Campbell, 2002, p. 207) was conducted with a section of the heart rate data before and during the go-around. The analysis leads to the hypothesis that the heart rate peak at the go-around is significant. Yet, further research has to be conducted to confirm or reject this hypothesis.

This study shows that even small changes in a head-up display symbology can lead to significant improvements, both, operational and with regard to human factors. It also demonstrates the rapid prototyping capabilities of the VRFS, which made the implementation and evaluation of the head-up display symbology feasible and cost-effective.

6.2. The Peripheral Vision Horizon Display (Study 2)

6.2.1. Introduction

In the previous user study, the advantages of head-up displays have been described. Besides these positive effects, some perceptual and cognitive issues come along with this technology (Crawford and Neal, 2006). These issues include so-called cognitive



(a) The standard HUD symbology

(b) The decluttered symbology during the final approach

Figure 6.9.: Eye tracking heat map of an approach with a head-up display in the cruise mode (left) with a wide scanning pattern and the decluttered mode (right) with a centered scanning pattern (Oberhauser, Dreyer, Mamessier, et al., 2015).

or mental tunneling effects: Fischer, Haines, and Price (1980) observed a degraded ability to perceive runway incursions during approaches with head-up display, an effect that is connected to mental tunneling, which has been confirmed by other studies as well (Fadden, Ververs, and Wickens, 2001; Wickens and Long, 1995). The study presented in this section deals with this negative side effect. It proposes and evaluates a system that is designed to counter mental tunneling: The Peripheral Vision Horizon Display (PVHD).

The Peripheral Vision Horizon Display

During the final approach, a decluttered symbology mode that offers optimized information presentation is usually activated in the HUD. In this mode, among others, the speed and altitude tapes as well as the vertical speed indicator are replaced by small numerical indications attached to the FPV. This design draws the focus of attention to the flight path vector, the HUGS, and the runway in the outside visual with the primary flight parameters close by. This decluttered design has an impact on the eye scanning behavior of pilots (see Figure 6.9) and can intensify mental tunneling effects (Bandow, 2006, p. 168).

To counter the mental tunneling effect, Bandow (2006, p. 184) proposed the use of a Peripheral Vision Horizon Display. The PVHD, or Malcolm Horizon, provides a peripheral stimulus to indicate the attitude of the aircraft (Malcolm, 1984). In its original form, as integrated in the reconnaissance aircraft Lockheed SR-71 Blackbird, the PVHD is a line that is projected on the instrument panel to increase attitude awareness (Assenheim, 1992). Following this approach, Bandow (2006, p. 184) proposes LED strips, which indicate the roll attitude of the aircraft, integrated into the struts of the cockpit as shown in Figure 6.10. He suggests that this peripheral stimulus can

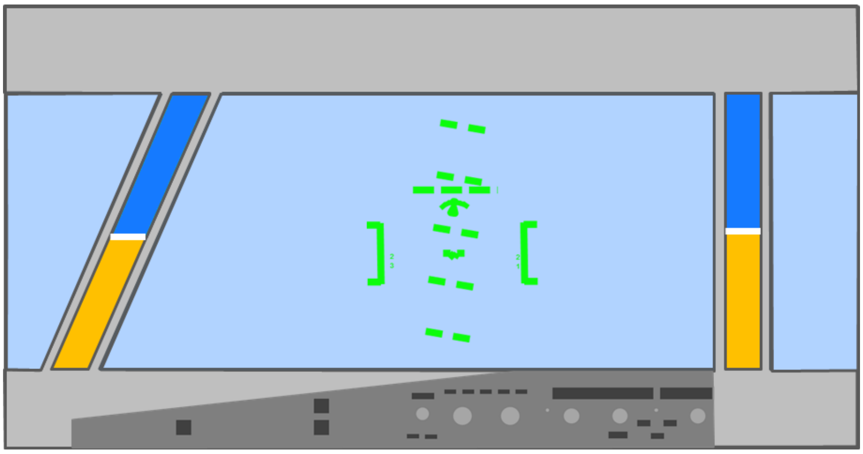


Figure 6.10.: A schematic representation of the proposed peripheral horizon display, integrated into the center and left strut of the cockpit.

counter mental tunneling effects that are caused by the use of decluttered and centered HUD symbologies. This first prototype of the PVHD was designed with an emphasis on roll movements. Roll movements are frequent and create significant motion cues in the peripheral vision. Future versions of the PVHD could represent a combination of pitch and roll attitude in order to act like a artificial horizon.

Hypothesis

The influence of the PVHD on mental tunneling effects will be evaluated in a trial. The hypothesis for this trial is: The PVHD decreases mental tunneling effects when using a decluttered HUD symbology in combination with a head-up guidance system on a HUD.

6.2.2. Methodology

Prototype Integration

The PVHD was integrated by leveraging the functionality offered by X-Plane's panel texture. Two generic instruments were created in Plane Maker, both connected to the roll attitude of the simulated aircraft. These instruments have been mapped to the struts of the 3D cockpit as shown in Figure 6.12. This results in two peripheral displays moving up and down according to the roll attitude.

As shown in Figure 6.11 the PVHD trials have been conducted using a powerwall for the visualization as the typically used HMD does not provide sufficient peripheral vision for this purpose.

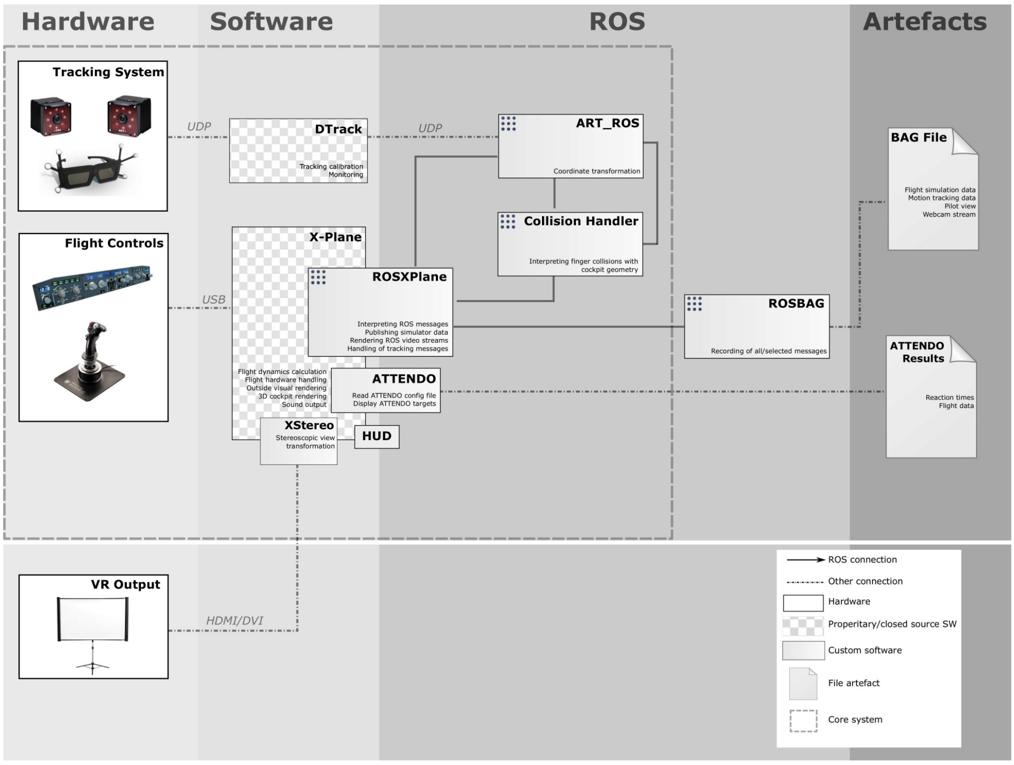


Figure 6.11.: The architecture of the VRFS for the PVHD evaluations. The ATTENDO method is integrated as an X-Plane plug-in.

Scenario

The primary task that has to be conducted during the PVHD trial is following a flight path indicated by the head-up guidance system on the HUD. This means the participants have to adjust pitch and roll in order to follow the guidance cue. Simultaneously, the participants have to maintain an airspeed of 209 knots. This flight path, which is the same in each trial, does not follow an operational procedure, i.e. is not realistic. It is simply used to create workload and a mental tunneling effect while using the head-up guidance system.

Each participant conducts a scenario for familiarization. This scenario is conducted without the PVHD but with the same flight path as in the main scenario. This helps the participants to get familiar with the primary flying task and the secondary cognitive task.

Dependent Measures

ATTENDO. The ATTENDO method is used to evaluate the distribution of visual attention during the PVHD trial. The available ATTENDO plug-in is configured to show

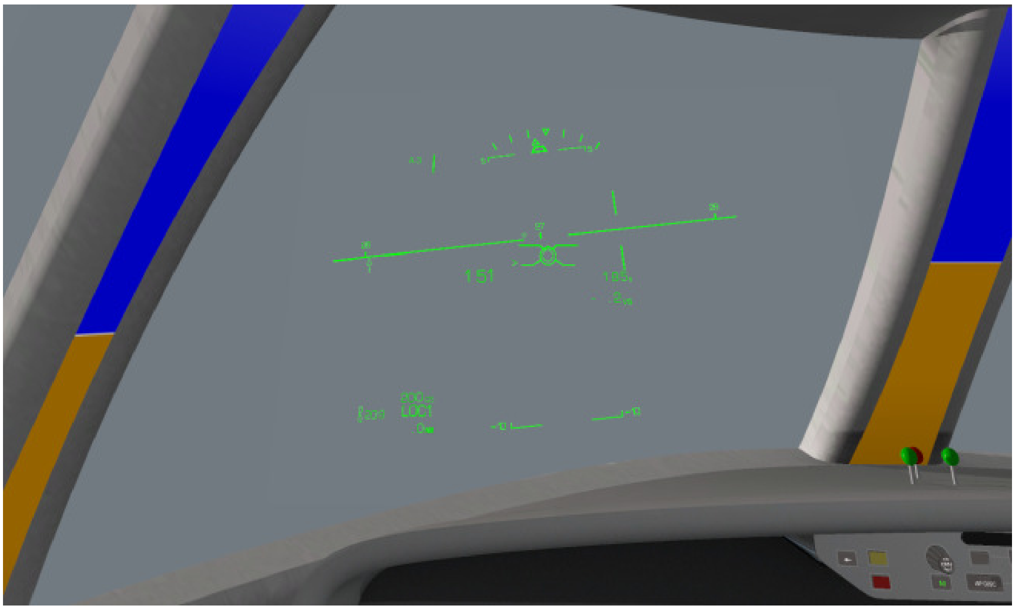


Figure 6.12.: The peripheral horizon display integrated in the virtual cockpit with a head-up display in X-Plane.

targets at six different locations, as shown in Figure 6.13, and in two layers of depth. Six target positions in a layer distant from the pilot and six target positions in a layer close to the pilot are used. The pilot's reaction time to the appearance of these targets is used to analyze the allocation of visual attention.

Flight Performance. The performance of the flight task, i.e. the deviation from the flight path, is measured during the PVHD trials. Yet, no in-depth evaluation of the flight path is performed as this parameter is solely used as a minimum quality criterion. If the average deviation from the flight path, i.e pitch and roll values, differs more than three standard deviations from the average across all participants, the minimum quality criterion is not met and the results obtained with the ATTENDO method are dismissed.

Independent Variable

The experiment has one within-subject variable: The presence of the PVHD and a cockpit without PVHD. The flight task is conducted twice for each participant, once with and once without the PVHD. The order of the two cockpit layouts is counterbalanced to avoid artifacts.

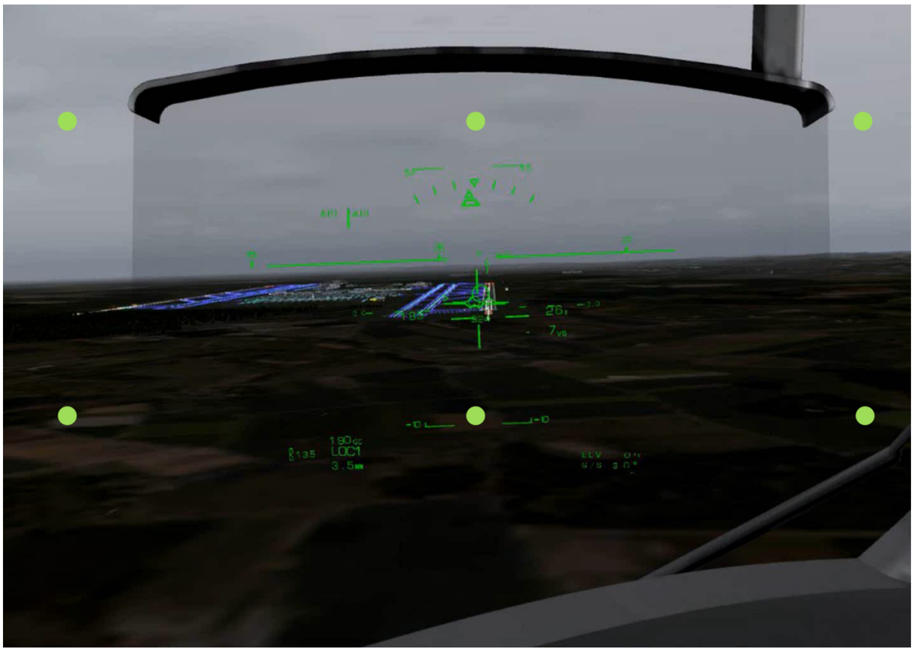


Figure 6.13.: The position of the ATTENDO targets (green dots) in the the virtual cockpit. The targets are presented in two different layers, a distant and a close layer.

Data Analysis

The distribution of visual attention in the two cockpit environments is compared and an Analysis of Variance (ANOVA) conducted. Differences in the detection rate of the distant layer and the close layer are analyzed as well.

Participants

Kasarskis et al. (2001) shows that experienced pilots have a different eye scanning behavior than novices. In order to exclude confounding factors that originate from different scanning behaviors, a group of eleven licensed pilots as well as twelve non-pilots were chosen for this trial. As the multi-control task is non-operational, no flying skills are necessary to complete the trial. The selected pilots are between 23 and 44 years old with experience ranging from 30 to 17,000 flight hours. They are either glider pilots, private pilots or hold a higher license. The twelve non-pilots are between 22 and 44 years old. All participants signed an informed consent form.

6.2.3. Results

The overall detection rate of all ATTENDO targets was 77.19 %. The anticipated increase in the detection rate in the peripheral vision as a consequence of the PVHD could not be confirmed. With the PVHD, the peripheral detection rate was 73 % and without the PVHD it was 76 %. The ANOVA analysis shows no significant difference in the detection rate due to the PVHD. In addition, there is no significant difference in the detection rate of the distant and the close layer targets and no significant influence of the PVHD on this detection rate.

6.2.4. Discussion

The results of the PVHD trials show that there is no significant difference in the detection rate of targets in the peripheral vision when using the PVHD. Yet, it cannot be ruled out that the result is influenced by confounding factors stemming from the virtual environment. Furthermore, the use of non-pilots is a viable point of criticism: The trained eye scanning behavior of airline pilots might influence the level of mental tunneling and therefore could have an impact on the presented results.

The trial demonstrates the rapid integration of a peripheral horizon display into a holistic flight deck; a task that would require extensive work in a conventional flight simulator. In addition, it shows that it is feasible to use the ATTENDO method in a virtual environment, which is an opportunity for future studies.

6.3. System Management Evaluation (Study 3)

6.3.1. Introduction

In this study, a novel HMI concept for system management with an emphasis on System/Component Failure/Malfunction (SCFM) in the aircraft cockpit is presented. A preliminary version of this study has been published as a paper by Dreyer and Oberhauser (2016a). This chapter is an extended version of this paper.

Research by Reveley et al. (2009) shows that between 1988 and 2003 20 % of all aircraft accidents and incidents were caused by or involved some kind of SCFM. These problems could be solved with more reliable system components. Yet, with reliability already at a very high level, there is not much room for improvement without a paradigm change like the introduction of more electrical aircraft (Rosero et al., 2007). Hence, pilots will be confronted with SCFM; the cockpit design should assist them in handling these issues.

Resolving a system-related issue in the cockpit involves the use of checklists and interaction mainly in the overhead panel. In the OHP, single systems manifest in the form of buttons, knobs, and switches. This system architecture based design has many advantages in terms of safety, accessibility, and redundancy and it is certified and well proven (Dreyer, Oberhauser, and Bandow, 2014). Yet, some issues can arise: The improper use or the non-use of checklists in connection with SCFM can be the cause for an incident or an accident (Degani and Wiener, 1990; Doolen, Nicolade, and Funk, 2000). Multiple redundant interactions have to be conducted to reach the desired system state, which leads to long task completion times that can be problematic in time critical incidents. For example, after a double engine failure during US Airways Flight 1549 shortly after take-off, the checklist could not be completed before the successful ditching into the Hudson River (National Transportation Safety Board, 2010, p. 87). Another issue comes with the complexity of the overhead panel, which correlates to the complexity of the underlying systems, which is again related to the weight of the panel.

A novel, user-centered approach for system management HMI is presented in this user study. Thus, low-level single actions have been replaced by high-level functions and corresponding automation. Other functions were moved to a screen in the head-down area. The system management concept was developed in two consecutive studies. First, an optimized overhead panel was designed and evaluated as part of the ALICIA project. In a following study that was conducted as part of the Advanced Cockpit for Reduction of Stress and Workload (ACROSS) project, a complementary head-down display was developed and evaluated in a user trial.

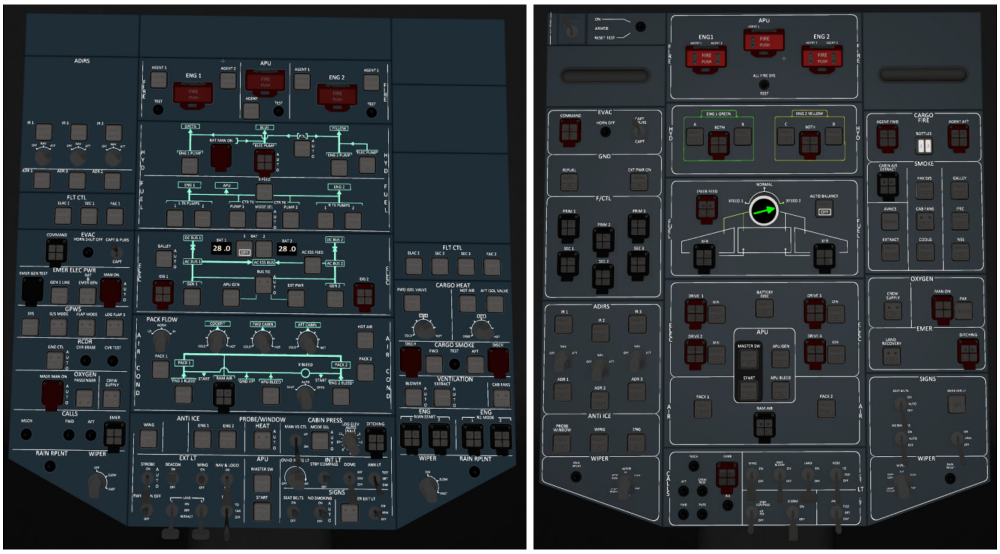
Analysis

An analysis of the system management concepts of three existing aircraft (ATR72, Airbus A330 and A380) was conducted. The aircraft that have been investigated reflect different sizes (a regional aircraft, a wide-body airliner, and today's biggest airliner), different propulsion technology (twin turboprop, twin engine jet, and four engine jet) and age (entry of service 1989, 1994 and 2007) (Dreyer and Oberhauser, 2016a). The overhead panels of these three aircraft have been analyzed with regard to the control elements (push-buttons, guarded push-buttons, knobs, and levers), their appearance in checklists, and their criticality. Criticality is defined by the appearance in abnormal or emergency situation procedures and the irreversibility of actions triggered by a control element (Schmidt, 2011).

Overhead Panel Design

Based on the analysis of the criticality, a generic set of functions for a twin jet engine A320-like aircraft was compiled. This set led to an emergency overhead panel that only serves hardware controls for highly critical actions. Based on non-procedural knowledge gathered from expert interviews, more functions were added to the overhead panel. For example, frequently used elements like the wipers or the light panels have to be present as hardware elements for usability and accessibility reasons. Control elements that have to be accessible in smoke conditions (in the cockpit) have to be hardware elements as well in order to provide haptic feedback in situations, where pilots' vision is limited.

All these considerations led to the design of an overhead panel as shown in Figure 6.14b. Compared with the legacy design, an A320 overhead panel as shown in Figure 6.14a, the OHP has less control elements, i.e. is less complex. Besides the reduction of complexity, other key features include a rotary knob with an integrated system indication (Dreyer and Oberhauser, 2016c), a novel fire push-button (Dreyer, Oberhauser, and Schmidt, 2015), and a re-arrangement of the panels' functional groups. For



(a) The legacy overhead panel

(b) The novel overhead panel

Figure 6.14.: The legacy Airbus A320 overhead panel (left) and the novel overhead panel (right).

example the APU panel was integrated into the closely related ELEC panel; in addition, a new SMOKE panel was introduced that holds all SMOKE counteraction buttons. A new EMER(gency) panel was introduced that provides quick access to functions like ditching or emergency oxygen supply.

The rotary knob is used in case of a fuel leak or a fuel imbalance. It controls the fuel transfer functionality between the outer wing tanks in an intuitive way without the necessity of controlling single pumps. If the automation commands a fuel transfer, the indication on the rotary knob changes accordingly. The novel fire push-button adds automation with an optimized level of indication: The legacy Airbus fire fighting procedure includes several single actions like pushing the fire push-button and releasing fire extinguishing agents. These single actions have been combined into one single control element as shown in Figure 6.15. This paradigm of integrating single actions

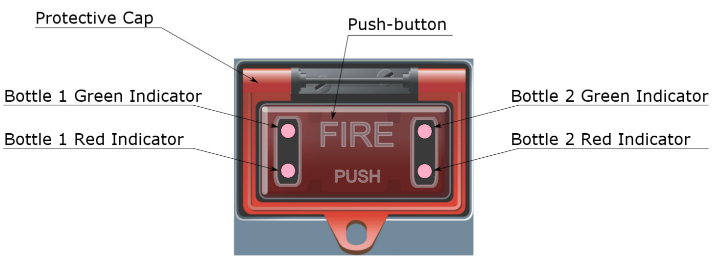
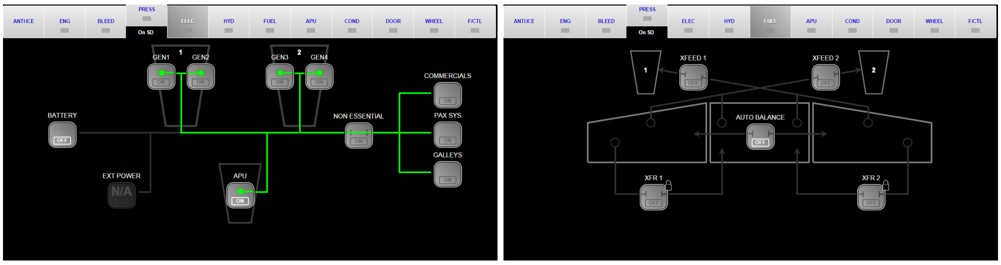


Figure 6.15.: The elements of the integrated fire push-button developed for the new overhead panel (Dreyer and Oberhauser, 2016a).



(a) The ELEC page of the Soft-ICP (b) The FUEL page of the Soft-ICP

Figure 6.16.: The system management touchscreen Soft-ICP. The ELEC page (left) and the FUEL page (right).

into high level, automated actions was used repeatedly in the design of this novel OHP.

Soft-ICP Design

The control elements that do not meet the criticality criteria or do not have other reasons to be included in the OHP, were moved to the head-down area. The functionality was included into a touchscreen close to the Electronic Centralised Aircraft Monitor (ECAM), a display that presents information about the aircraft system states. The proposed display, called Soft-ICP, consists of multiple system pages that are closely related to the layout of the overhead panel and to the system pages of the head-down System Display (SD). The control elements in the Soft-ICP recreate the look-and-feel of OHP buttons, a design choice that reflects the need to control both the OHP for critical actions and the Soft-ICP for non-critical actions in one checklist procedure. Figure 6.16 shows two system pages of the Soft-ICP.

The Soft-ICP was developed as a prototype in HTML 5 on a tablet device (see Figure 6.17). During the development, the system was presented to Airbus test pilots in part-task-evaluations. Here, a think aloud protocol, a commonly used method in HCI design and research (Holzinger, 2005), was used to test the proposed interface. This feedback was then used to further refine the design before the final evaluation.

Hypotheses

The proposed new HMI for system management, i.e. the overhead panel and the Soft-ICP will be evaluated. The hypotheses for this evaluation are:

1. The novel system management HMI decreases the task completion time of system management tasks.
2. The novel system management HMI decreases mental and physical workload.
3. The novel system management HMI increases situation awareness.

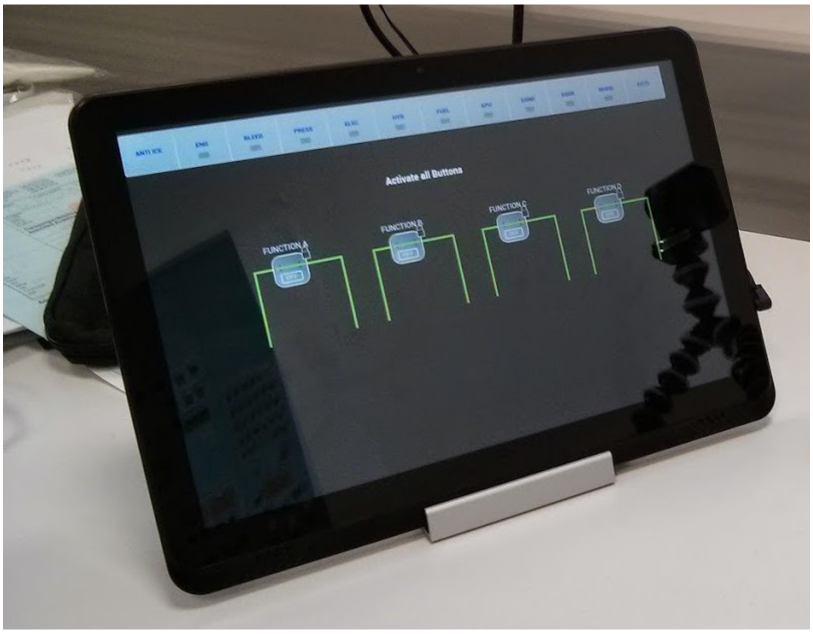


Figure 6.17.: The standalone prototype of the Soft-ICP that was used for the part-task evaluations. The prototype was implemented as a HTML 5 application and deployed on a consumer tablet device.

6.3.2. Methodology

Prototype Integration

The OHP was designed in CATIA V5 and integrated into the VRFS with the presented import pipeline (see Section 4.2.2 on page 42). All the buttons have been animated and, if necessary, equipped with dynamic textures. In the collision handler in Unity3D, appropriate box colliders with a connection to VRFS commands have been defined. These steps led to a fully functional virtual overhead panel.

The Soft-ICP, which was implemented as a HTML 5 prototype, was integrated into the VRFS using the approach presented in Section 4.4.2 (on page 69). The content of the HTML 5 prototype is streamed into the virtual cockpit via ROS. The display is placed in the central head-down area matching the location in the virtual cockpit, i.e. it creates a mixed mock-up. Figure 6.19 shows the overhead panel and the Soft-ICP in the virtual cockpit. The architecture of the VRFS for this trial is illustrated in Figure 6.18.

Scenario

Two user trials are conducted to evaluate the novel system management HMI concept. First, the OHP is evaluated: The scenario starts in cruise altitude. After an AIR PACK FAULT, the crew has to perform an emergency descent followed by system management interactions as given in the checklist. Shortly after completing the checklist, an engine

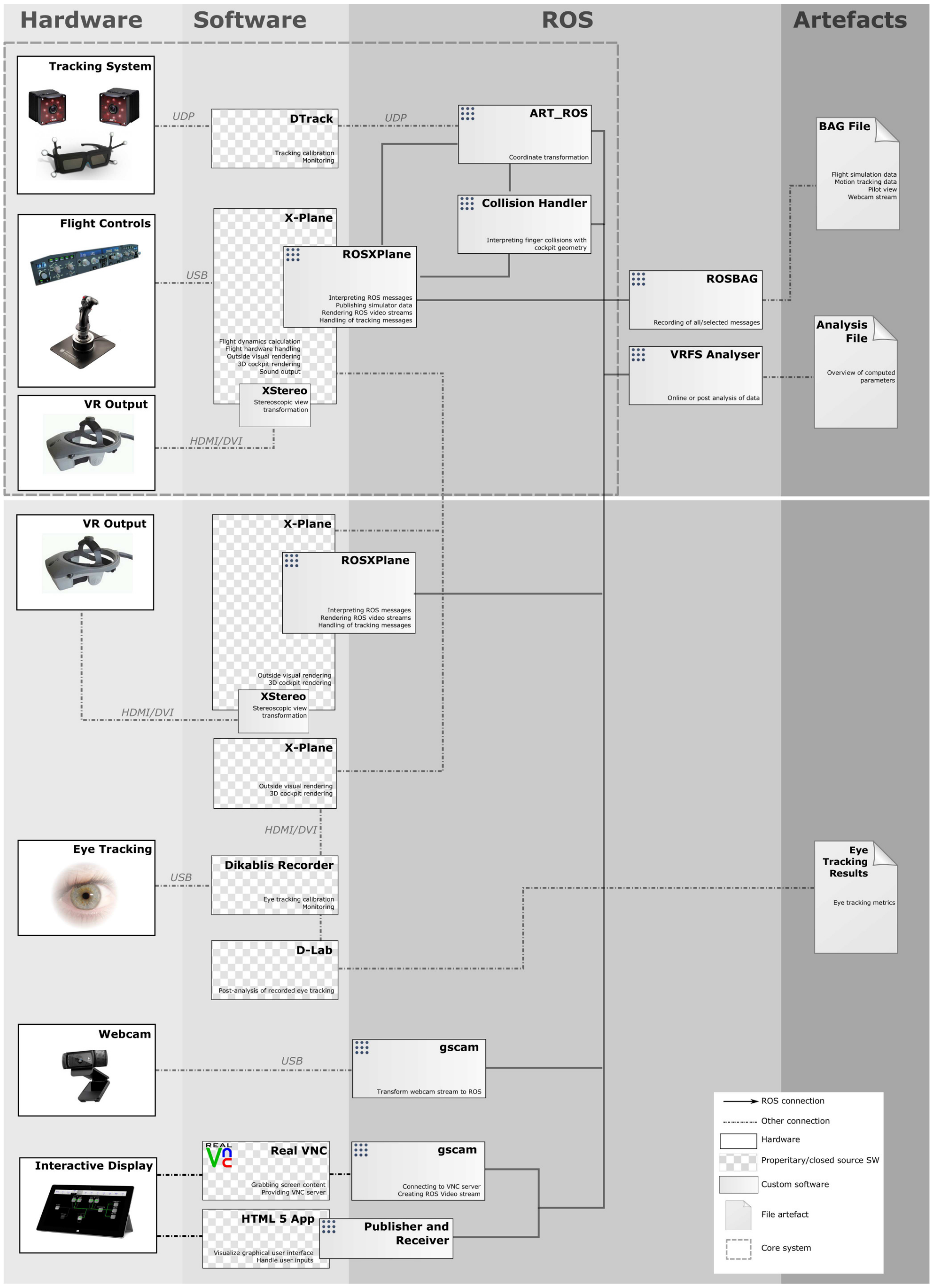


Figure 6.18.: The architecture of the VRFS for the system management HMI evaluations.

Figure 6.19.: The virtual cockpit with the overhead panel and the Soft-ICP in the central display of the cockpit.

fire occurs. Again, the crew has to deal with this situation with the help of the appropriate checklist.

In a second trial, the Soft-ICP is evaluated: In this scenario, a multiple SCFM situation is chosen. Only a few seconds after a loud explosion noise, several messages appear on the ECAM display regarding the FUEL and the ELEC system. This results in the loss of one fuel pump. The checklist requires the crew to monitor the fuel tanks for a leak. As a leak is suspected, the crew has to isolate single tanks to find it and to take further actions according to the checklist afterwards.

During both scenarios, the subject has the role of the pilot not flying. The experiment supervisor, who is also immersed in the virtual environment (see Figure 6.20), is mimicking the pilot flying. Both pilots can see each other in the form of a manikin. In both scenarios, a familiarization scenario in the virtual environment is conducted by the subjects in order to get used to the virtual environment.

Dependent Measures

In both scenarios, the novel system management approach is compared with that of a legacy system, the Airbus A320. Thus, both trials have one within-subject variable, the system management concept.

Independent Variables

Performance. The performance with regard to the system management task is measured using the task completion time. In this particular case, the TCT is defined

Figure 6.20.: The pilot (front) and the experiment supervisor (back). Both are immersed in the virtual environment.

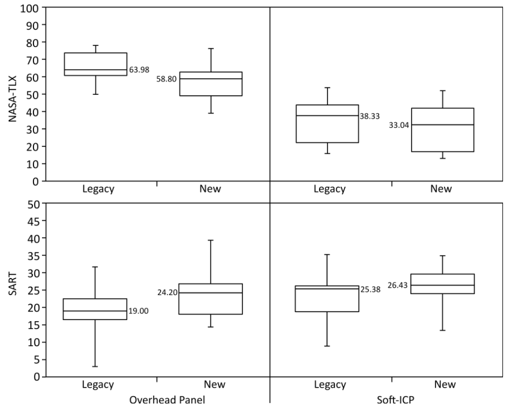
by the beginning of the incident to the point in time when the incident has been resolved.

Workload and Situation Awareness. In order to measure the workload, the quantitative NASA-TLX questionnaire is handed to the pilots after each scenario. To measure and analyze situation awareness, the quantitative SART questionnaire is used after the scenario.

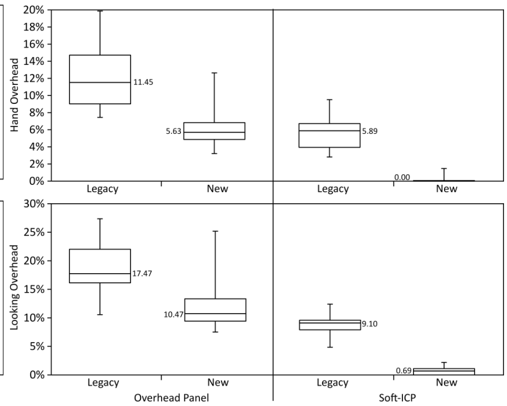
Posture Analysis. In order to evaluate physical workload, two parameters are derived from the pilot's posture: First, the hand overhead time, i.e. the time that the pilot interacts with the overhead panel. This overhead interaction without support can cause moderate physical strain in the shoulders and back area (Rodgers, 2004). Second, the head overhead time, i.e. the time the pilot is looking at the overhead panel, is measured. The head overhead time increases physical workload. In addition, it decreases situation awareness with regard to the primary flying task as the pilot is not looking at the head-down instruments during that time.

Data Analysis

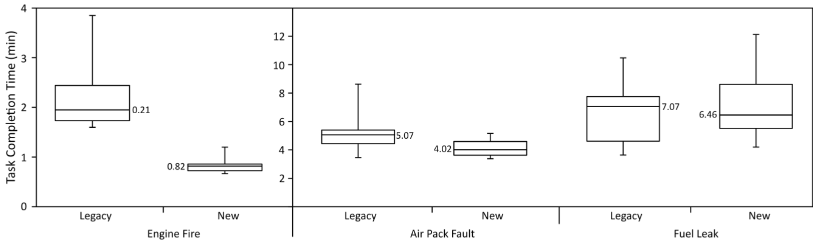
The results from both scenarios, the legacy system management approach and the novel system management concept, are compared. The data is analyzed with a Wilcoxon signed-rank test for paired samples. Alpha was set at .05.



(a) SART and NASA-TLX



(b) The posture analysis: Looking overhead and hand overhead times



(c) Task Completion Time (TCT)

Figure 6.21.: The results of the posture analysis, the questionnaires, and the task completion time visualized as box plots.

Table 6.3.: Statistical analysis of the task completion time of the overhead panel and the Soft-ICP trials.

	Task Completion Time in Minutes		**Wilcoxon Signed-rank Test for Paired Samples** ($\alpha = .05, n = 11, df = 1$)			
	Mean	**SE**	**n**	**Z**	**p**	**w**
OHP						
AIR Pack FLT						
Legacy	5.124	.407	11	2.356	**.0092**	.502
New	4.124	.184				
ENG Fire						
Legacy	2.215	.209	11	2.890	**.0005**	.616
New	.824	.047				
Soft-ICP						
FUEL Leak						
Legacy	6.657	.798	7	.254	.3999	.068
New	7.298	9.01				

Participants

The first trial was conducted in the same session as the Green Disc Concept trial presented in the user study in Section 6.1.1. Therefore, the same eleven pilots from various German airlines participated.

Eight pilots participated in the second trial. The pilots volunteered for the experiment or were part of the project consortium. They are active airline pilots from European airlines as well as retired pilots. These pilots have a mean age of 52 years and an average experience of 10,225 flight hours. Each participant signed an informed consent form prior to the experiment.

6.3.3. Results

The results of the statistical analysis of the dependent measures are discussed in this section. An overview of the results is presented in Figure 6.21.

Task Completion Time

The results of the task completion time are summarized in Table 6.3 with a visualization in Figure 6.21c. For the overhead panel scenarios, the time to resolve the AIR Pack Fault is reduced from a mean of 5.124 minutes to a mean of 4.124 minutes with the novel overhead panel design. The ENG Fire could be extinguished in well under one minute (.824 minutes) with the new design, whereas it took over two minutes (2.215 minutes) with the legacy design. The Wilcoxon signed-rank test for paired samples shows that the differences in both measures are significant with a large effect ($w = .502$ and $w = .616$). For the Soft-ICP trials, the null hypothesis cannot be rejected

Workload

The workload analysis conducted with the NASA-TLX questionnaire shows a significant reduction ($p < .0068$) of workload with the novel overhead panel design, with a large effect ($w = .502$) as shown in Figure 6.21a and Table 6.4. The total NASA-TLX score is reduced from a mean of 65.674 with the legacy overhead panel to a mean of 32.032 with the new overhead panel. The NASA-TLX scores for the Soft-ICP trials show no significant change. Thus, the null hypothesis cannot be rejected.

Another indicator for workload, in particular physical workload, is analyzed with the help of the hand overhead time. The dwell time in which the hand was overhead is significantly longer with the legacy OHP, both in the OHP trial ($p < .0005$) and in the Soft-ICP trial ($p < .0078$) both with large effects ($w = .616$ and $w = .610$) as shown in Table 6.5. In the overhead panel trials, the hand overhead dwell time is reduced from a mean of 12.55 percent of the trial length to 6.30 percent. In the Soft-ICP trials, this reduction is even stronger with a reduction from 8.80 % to only .73 %. A visualization of the hand positions during the Soft-ICP trials of one participant is presented in Figure

(a) The legacy overhead panel. (b) The novel overhead panel.

Figure 6.22.: Three-dimensional visualization of the hand movements in the legacy cockpit (left) and the novel cockpit (right).

6.22 as an example. It is clearly visible that with the Soft-ICP no interaction in the OHP is necessary in the given scenario.

Situation Awareness

The ratings of the subjective SART questionnaire show an increase ($p < .0269$) of situation awareness in the OHP trial with a medium to large effect ($w = .408$), as shown in Table 6.4. No significant change could be observed during the Soft-ICP trials, i.e. the null hypothesis cannot be rejected.

The posture analysis (see Table 6.5) is used to measure the dwell time it takes the pilot to look at the overhead panel. During this time, the pilot is not able to look outside or look at the head-down displays; this decreases the situation awareness, in particular the spatial awareness and system awareness. The analysis shows that the looking overhead time is significantly decreased in both, the OHP ($p < .0005$) and the Soft-ICP ($p < .0078$) scenario with large effects ($w = .616$ and $w = .610$).

6.3.4. Discussion

This study presents a user-centric design approach for system management HMI that is stemming from the analysis of existing aircraft HMI concepts. This analysis resulted in the design of an overhead panel and a touchscreen device, the Soft-ICP. Both devices have been compared with a baseline Airbus A320 cockpit HMI, which, although it is a design from the 1980s, still reflects the philosophy of modern flight decks (Dreyer and Oberhauser, 2016a).

The results of the evaluation are not conclusive for the Soft-ICP. Neither for subjective workload, situation awareness nor task completion time the null hypotheses can be rejected. Yet, a significant decrease of hand overhead time and looking overhead time was observed. Nevertheless, subjective comments from the pilots that participated in

Table 6.4.: Statistical analysis of the NASA-TLX and SART questionnaires.

HMI Concept	Questionnaire Ratings mean	SE	Wilcoxon Signed-rank Test for Paired Samples ($\alpha = .05, n = 11, df = 1$) n	Z	p	w
SART						
OHP						
Legacy	18.492	2.363	11	1.912	**.0269**	.408
New	24.095	2.411				
Soft-ICP						
Legacy	23.056	2.766	8	.770	.2305	.193
New	26.250	2.329				
NASA-TLX						
OHP						
Legacy	65.674	2.866	11	2.356	**.0068**	.502
New	56.577	3.641				
Soft-ICP						
Legacy	35.276	4.884	8	.910	.1814	.228
New	32.032	5.555				

Table 6.5.: The statistical analysis of the posture analysis: The hand overhead dwell time and the looking overhead dwell time.

	Dwell Time in % Mean	SE	Wilcoxon signed-rank test for paired samples ($\alpha = .05, n = 11, df = 1$) n	Z	p	w
Hand Overhead						
OHP						
Legacy	12.55	1.34	11	2.890	**.0005**	.616
New	6.30	.82				
Soft-ICP						
Legacy	5.70	.81	7	2.282	**.0078**	.610
New	.23	.21				
Looking Overhead						
OHP						
Legacy	18.15	1.57	11	2.890	**.0005**	.616
New	12.07	1.55				
Soft-ICP						
Legacy	8.80	.77	7	2.282	**.0078**	.610
New	.73	.32				

the trials lead the way for further improvement of the Soft-ICP. In these comments, the use of touchscreens was generally rated positively and some pilots suggested that the technology should be further exploited, i.e. further deviation from the overhead panel's single system philosophy is desirable according to these pilots.

The other investigated HMI component is the new overhead panel. The design showed significant advantages in situation awareness and workload in both the subjective as well as objective measures. These results indicate that there is a huge potential for improvement in legacy button panels of flight decks, in particular for dealing with SCFM.

In this user study, a complex system management scenario was created. It requires numerous interactions with the overhead panel, an integrated touchscreen, a virtual checklist, and a second crew member. The user study demonstrates that it is possible to recreate such a complex scenario using the VRFS. Yet, the limitations of fully virtual interactions became apparent—an observation that has been quantified in the evaluation of virtual interaction in Section 5.1.2.

6.4. Head-worn Device Evaluations (Study 4)

6.4.1. Introduction

Head-worn Device

Head- or helmet-mounted displays in aircraft cockpits are a combination of a head tracking system and a see-through device on which information is displayed. These systems overcome the forward facing field of view limitations of head-up displays by providing an unlimited field of regard (Moir, Seabridge, and Jukes, 2006, p. 434). These displays are well established in modern military fighter jets and rotary-wing aircraft. In the commercial civil flight deck, head-mounted displays are not yet in operational use. Nevertheless, research on the use of such systems in the civil flight deck exists. Arthur et al. (2007) presented a concept for using a HMD for ground operations. Visibility limitations during taxiing are overcome by adding conformal symbology and the augmentation of 3D objects. In a study by Ernst, Doehler, and Schmerwitz (2016), not only the outside visual is visually enhanced but also the head-down displays.

Recently, new types of devices with see-trough displays have emerged. These new Head-worn Devices (HWDs) are more than just displays. The term is used for devices, in particular smart glasses, that include sensors such as a (depth) camera, an accelerometer, an integrated computing unit, and wireless connectivity. In this user study, as a part of the ACROSS project, a head-worn device will be evaluated that closely resembles the functionality of a head-up display. In a first trial, a basic design choice for the HWD symbology is investigated, i.e. the position of the non-conformal symbology. In a second trial, a novel symbology is evaluated to aid the task of a parallel runway approach.

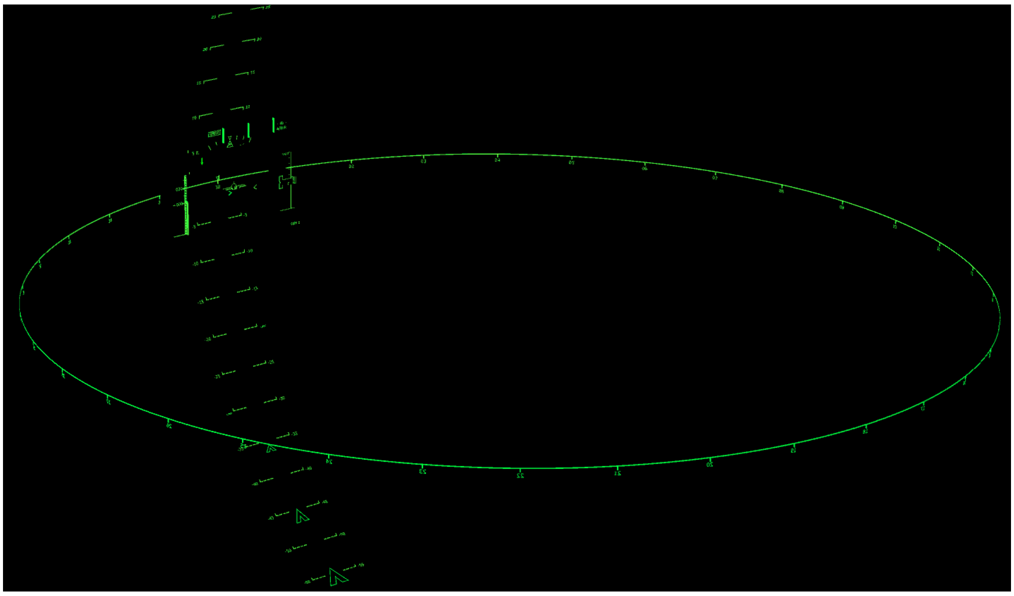


Figure 6.23.: The implementation of the head-worn device symbology in Unity3D. The ring has three degrees of freedom, pitch, heading and roll and is always aligned with the horizon. The non-conformal symbols behave according to the symbology reference modes.

There are two types of symbology: conformal and non-conformal. Conformal symbology represents outside visual features, e.g. the artificial horizon, the pitch ladder, or a runway representation. Non-conformal symbology has no representation in the outside visual, this includes for example sensor or navigation information (Long and Wickens, 1994).

The head-worn device symbology was implemented using Unity3D, a popular gaming engine. The conformal symbology has to be designed spherically around the users' viewpoint to match the outside visual. This spherically arranged information is world referenced, as it represents distinct positions in the outside world (Wickens, Ververs, and Fadden, 2004). In Unity3D, this was implemented using 3D objects with the main camera at the center of the spherical arrangement (see Figure 6.23). These 3D objects were animated to create the functionality of a head-worn device symbology. The system is deployed to a pair of Epson BT-200 see-through glasses that run on the Android operating system. This hardware is already in use in General Aviation (GA) cockpits (Elder and Vakaloudis, 2015) and therefore could be a viable option for commercial applications or at least for flight tests in commercial flight decks. Hence, not only the proposed symbology is subject to this research but also the used augmented reality hardware in order to contribute to the further development of this technology.

For the positioning of the non-conformal symbology, three different symbology reference modes have been designed and implemented:

1. Aircraft referenced: The non-conformal symbology is attached to the aircraft reference, i.e. it is always forward facing, attached to the aircraft's structure. This is similar to fixed head-up displays.

(a) The trajectory of the scenario: The yellow trajectory indicates the own aircraft; the cyan trajectory indicates the other aircraft with the late runway change.

(b) The symbology of the head-worn device: The pilot is looking slightly to the right to fixate the other aircraft which already entered the non-transgression zone.

Figure 6.24.: The implementation of the parallel runway aiding system in the head-worn device symbology (bottom). The trajectory of the scenario with the late runway change (top).

2. Flight path referenced: The non-conformal symbology is attached to the flight path vector, i.e. in the direction the aircraft is flying.
3. Head referenced: The non-conformal symbology is attached to the pilot's head and is always visible in his or her field of view.

The goal of this pre-study is to obtain feedback on the three reference modes from a small group of expert pilots.

Parallel Runway Approach

An additional research question is the evaluation of a system to aid parallel runway approaches. These types of approaches are becoming more common as the density of air traffic increases. Yet, with a (linear) separation between the aircraft of only 1.5 nautical miles and a minimum runway spacing (parallel separation) of only 1525 meters today, this is a challenging task for both air traffic controllers and pilots (Lisker, 2014).

Hence, as the Traffic Collision Avoidance System (TCAS) cannot sufficiently cover the case of parallel runway approaches (Pritchett et al., 1995), a new system is being proposed in this research. Based on the TCAS data, the linear distance separation and the parallel distance separation of other aircraft are monitored during the parallel approach. In the HWD, the other aircraft is highlighted with a green square and the distance to the other aircraft is presented. If the parallel distance becomes critical the square turns red and an additional warning is issued in the center of the display. This behavior is illustrated in Figure 6.24b.

6.4.2. Methodology

Prototype Integration

The software of the HWD symbology was connected to the flight simulation via rosbridge (see Figure 6.25). In order to do this, a websocket library was integrated into the Unity3D application. An ART tracking target was added to the glasses in order to get the pose of the pilot's head. The rotation of this tracking target is passed to the Unity3D application and drives the main camera of the scene. Hence, a precise head tracking and an interactive representation of the symbology are generated.

For the visualization, two variants have been implemented. First, a projector is used for visualizing the cockpit and the outside visual as shown in Figure 6.26. A second, slightly modified version of the HMD can be worn on top of the smart glasses as shown in Figure 6.27. The latter approach of combining the see-trough device and the virtual reality headset enables a full 360 degree field of regard and enables thorough testing of the symbology. However, due to space limitations this only works if the subject does not wear prescription glasses. With both approaches, the actual hardware head-worn device is used for displaying the symbology, which means that the characteristics of the device itself e.g. resolution, contrast, or brightness can be experienced by the wearer.

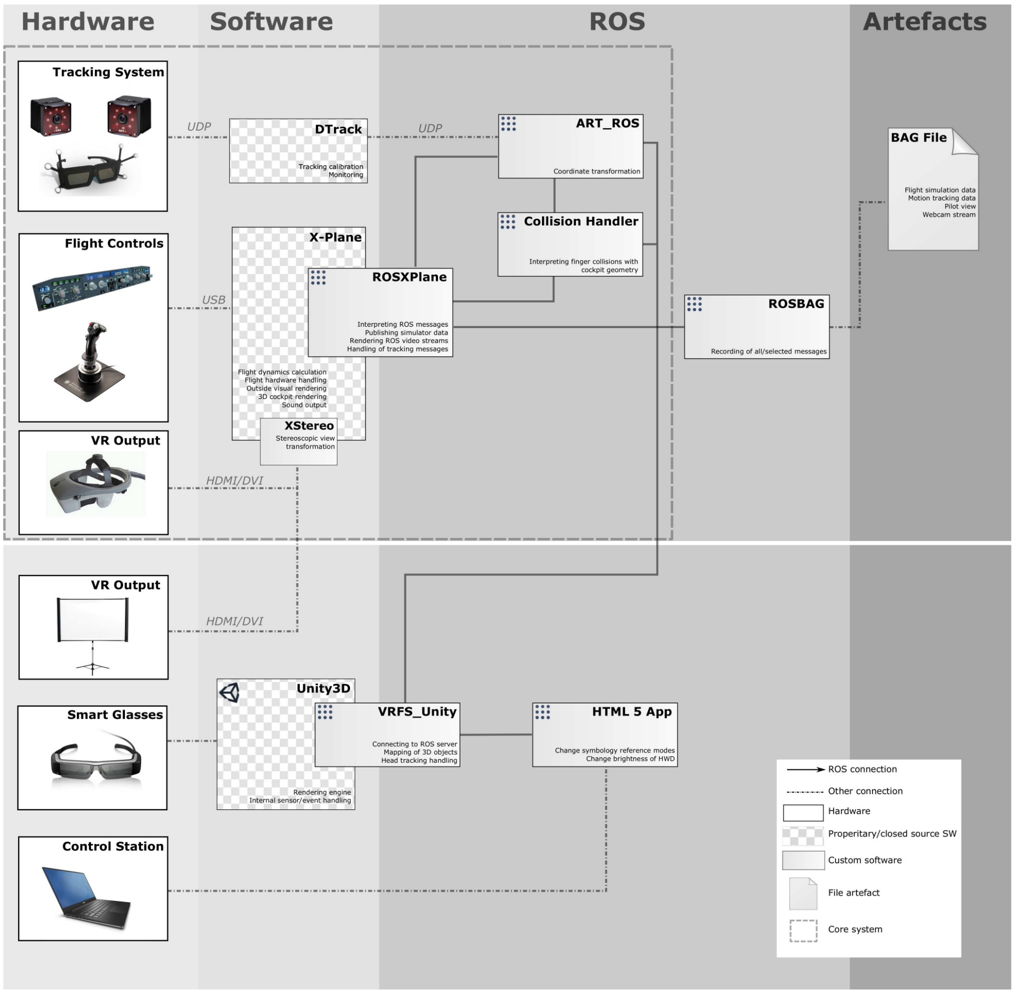
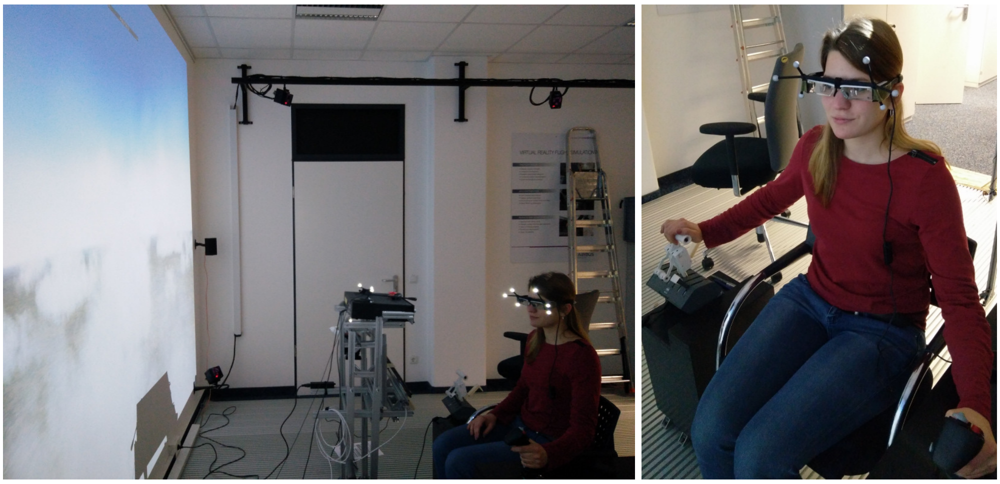


Figure 6.25.: The architecture of the VRFS for the head-worn devices evaluations. The head-worn device is implemented on a pair of smart glasses that run an application developed with Unity3D. A control station, implemented as an HTML 5 website, is used to change the implemented symbology reference modes on the glasses.



(a) Participant in front of the powerwall

(b) The head-worn device and the ART tracking target

Figure 6.26.: The experiment setup of the head-worn device study with the powerwall for visualization of the outside visual.

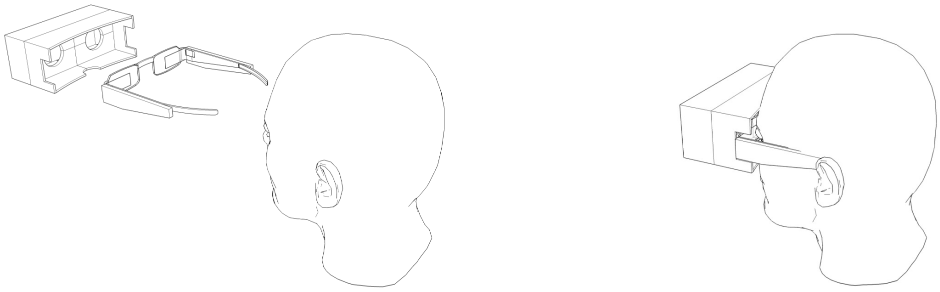
Scenario

Two scenarios have been developed for this evaluation. In the first scenario, the aircraft is set to cruise altitude. The pilot has to fly the aircraft manually while the different symbology reference modes are presented. The subject can freely choose, which concept he or she wants to try as the experiment supervisor has the ability to change the modes on an external web interface.

In the second scenario, a fully automated approach to the RWY 08L of the airport of Munich (EDDM) is conducted. The scenario already starts in the final approach phase with the ILS fully captured and the final configuration already set. Thus, besides monitoring the approach, no interaction from the pilot is necessary. A second aircraft, which is approaching the parallel RWY 08R is ahead of the simulated aircraft. Shortly before touchdown the other aircraft performs a late runway change without prior notice. With this maneuver, the parallel and linear distance falls below the limits and the pilot has to perform a go-around, which is illustrated in Figure 6.24a.

Questionnaires

After the two scenarios, subjective questionnaires are handed to the pilots. Most questions have been designed based on the Likert scale which uses a scale of five categories to rate a question: Strongly agree, agree, undecided, disagree, strongly disagree. In addition, more subjective questions have been designed to evaluate the symbology reference modes and the parallel runway approach symbology.



(a) The components of the head-worn device and head-mounted display. (b) The head-worn device and the head-mounted display combined and worn by a user.

Figure 6.27.: The user of the head-worn device in combination with the head-mounted display (Dreyer, Oberhauser, Prücklmeier, et al., 2017).

For the three symbology reference modes, the pilots have to rank the concepts. This ranking is then transferred into points: Two points for the first place, one point for the second place, and no points for the last place.

Participants

Seven pilots participated in the HWD trials. All these pilots are members of the project consortium and part of a so-called External Expert Advisory Group (EEAG), a group of advisers that support the ACROSS project. The participants have a mean age of 47 years and have an average flying experience of 8800 flight hours.

6.4.3. Results

The Head-worn Device Hardware

In a first subjective questionnaire, the head-worn device hardware itself was evaluated. The results, as shown in Figure 6.28, in general show a good acceptance of the hardware choice. Only the field of view (question 1) and the weight of the device (question 4) received some negative feedback but are still rated positively by most of the participants. The delay of the system received mixed feedback. Yet, the delay is not solely related to the head-worn device itself but to the tracking system, the data processing, and the network connection. For the integration into a flight-test cockpit environment these components should be optimized to minimize the delay.

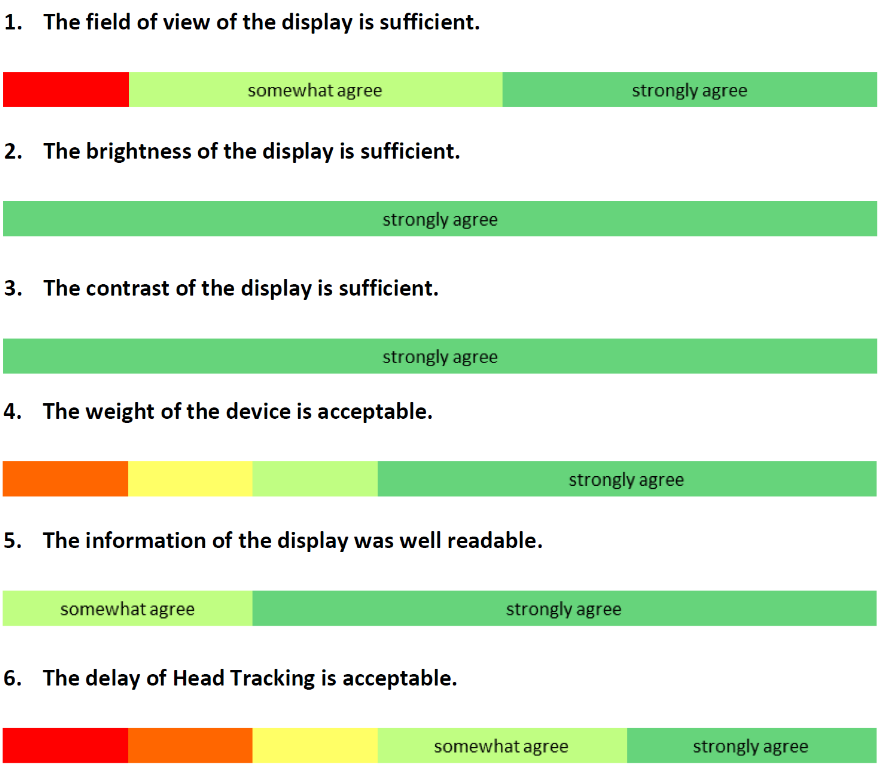


Figure 6.28.: The results of the subjective questionnaire regarding the head-worn device hardware. The results of the answers on the Likert scale are visualized with the help of colors. Dark red: strongly disagree, red: somewhat disagree, yellow: undecided, green: somewhat agree, dark green: strongly agree.

Symbology Reference Modes

The ranking of the symbology reference modes show a tendency towards the head reference mode (ten points). The flight path vector mode follows with five points and the aircraft reference mode received three points. Four pilots, who did not wear subscription glasses, experienced the reference modes through the combination of HWD and HMD. The other pilots used the HWD in combination with the powerwall.

The Parallel Runway Approach

The results of the questionnaire (see Figure 6.29) after the parallel runway approach indicate an overall positive attitude towards the use of the presented system. Information on the position and distance of other aircraft is desirable for a majority of the participants. Different opinions were expressed whether the altitude of the other aircraft should be displayed or not. How this system could be used in combination with the already installed TCAS system was one of the main remarks in informal exchange

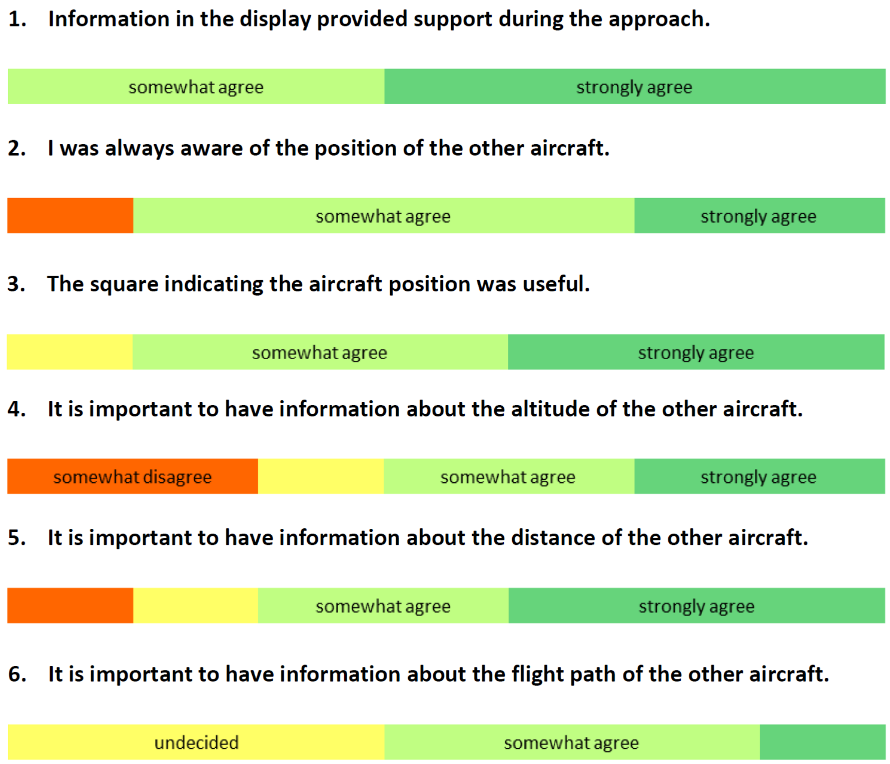


Figure 6.29.: The results of the subjective questionnaire regarding the parallel approach aid. The results of the answers on the Likert scale are visualized with the help of colors. Dark red: strongly disagree, red: somewhat disagree, yellow: undecided, green: somewhat agree, dark green: strongly agree.

after the trials. More research on this topic is necessary to provide an unambiguous alerting system that does not interfere with existing components.

6.4.4. Discussion

In general, the trials show a positive attitude towards the use of head-worn devices in the flight deck and, in particular, in using the Epson BT-200 device for this task. As there was only a small sample of pilots involved in the trials, these results are not representative. Yet, the results support the further development of head-worn displays in the flight deck.

The user study demonstrated the integration and evaluation of a hardware component into the VRFS. The combination of a see-trough device with a virtual reality headset proves to be a viable option for testing the usability of such devices in the future.

Table 6.6.: An overview of the use cases and their demonstration in the user studies.

	User Studies Presented in this Chapter			
	(3) OHP/Soft-ICP	(1) HUD	(2) PVHD	(4) HWD
Pilot Use Case				
1.1 Trigger a Virtual Button	●	●		
1.2 Trigger a Guarded Virtual Push-button	●			
1.3 Trigger a Physical Push-button	●	●		
1.4 Use a Hardware Lever	●	●	●	●
1.5 Use a Flight Display	●	●	●	●
1.6 Interact with a Second Pilot	●			
Developer				
2.1 Import Cockpit Geometry	●	●		
2.2 Animate Objects	●	●	●	
2.3 Assign Dynamic Textures	●	●	●	
2.4 Integrate Prototypes*	●	●	●	●
Controller Use Case				
3.1 Load a Scenario	●	●	●	●
3.2 Induce System Malfunction	●			
Human Factors Practitioner				
4.1 Online Analysis of Data	●			
Shared User Studies				
5.1 Access Simulation Data	●			
5.2 Record Simulation Data	●	●	●	●
5.3 Integrate Human Factors Methods*	●	●	●	●

* A detailed table describing the integrated prototypes and human factors methods is presented in the respective chapters.

6.5. Discussion

6.5.1. Demonstrated Use Cases

The presented user studies demonstrate key features that have been implemented based on the use cases presented in Table 3.1 (on page 22). Table 6.6 shows an overview of the use cases that have been covered in this chapter. The interaction with virtual cockpit components was demonstrated mainly in the system management user study 3. Here, numerous time critical interactions with control elements in the overhead panel were conducted in close cooperation with a second immersed crew member. User study 1, the head-up display evaluation, also demonstrates the use of virtual buttons, yet, to a much smaller extent. Mixed mock-up interactions with the hardware FCU were used in both studies. In all presented user studies, hardware levers and flight displays were used to some extent, in order to control the simulated aircraft.

Table 6.7.: An overview of the user studies and the used types of prototype and human factors methods.

	User Studies Presented in this Chapter			
	(3) OHP/Soft-ICP	(1) HUD	(2) PVHD	(4) HWD
Prototype Integration				
System Prototype	●			
Detached Display	●			
Functional Display		●	●	●
Interactive Display	●			
Hardware Prototype				●
Human Factors Methods				
Eye Tracking	●	●		
ATTENDO			●	
Heart Rate Monitor		●		
Performance	●		●	
Questionnaires	●	●	●	●
Posture Assessment	●			

In study 1 and study 3, the emphasis was put on the 3D cockpit and the interaction in this holistic environment rather than the use of single systems like the PVHD in study 2 or the HWD in study 4. Thus, new cockpit elements were integrated: A head-up display in study 1 and a new overhead panel in study 3. In both studies, further modifications to the cockpit were conducted, such as the animation of objects and the implementation of new display content using the panel texture. To integrate the PVHD in study 3, this approach was used as well.

All four user studies rely on a specific, repeatable scenario that was set up once and loaded for every participant by the simulation controller. In user study 3, the controller was also responsible for inducing system failures.

In all scenarios, the human factors practitioner used recorded data (telemetry data, physiological data, audio/visual data) for conducting a post-analysis. In user study 3, the analysis of the head-up time and the posture analysis were conducted during the experiment, i.e. online. Therefore, the HF practitioner needed to be able to retrieve this data from the simulation environment during run-time.

6.5.2. Demonstrated Prototype Integration

In each user study, a specific technology was integrated and evaluated. As shown in Section 4.4, different types of prototypes can be integrated into the VRFS. Table 6.7 presents an overview of the user studies and shows which type of prototype was integrated.

In study 3, the system management evaluation, the FUEL, and the ELEC system of the baseline aircraft had to be modified. Hence, the existing system was deactivated and replaced by a new system simulation. In terms of HMI prototypes, an external display prototype was integrated to control these new systems. The display prototype consists of multiple system pages, some of which are not connected to the system simulation (detached) and some are used to control the aircraft systems (interactive). All other studies use functional display prototypes: In study 1, a head-up display symbology was integrated, study 2 uses attitude information to drive a peripheral horizon display, and study 4 uses telemetry data to drive symbology on a head-worn device. The study (study 4) that includes a see-trough HWD is the sole user study that does not only evaluate an HMI concept but also a hardware device for the use in the flight deck.

6.5.3. Demonstrated Human Factors Methods

In order to evaluate the integrated HMI technologies, human factors methods have been used. Table 6.7 gives an overview of these human factors methods.

The eye tracking system was used in the system management study (study 1) and the head-up display study (study 3). Further physiological measures were used in the form of a heart rate monitor in study 1 and a posture analysis in study 3. This posture analysis was conducted both manually, based on recorded video footage, and automatically by leveraging the data from the optical tracking system. The ATTENDO method was solely used in the PVHD trials (study 2) to evaluate the distribution of attention with the new HMI concept. The pilots' task performance was used as a measure in the system management trial (study 3). In the PVHD study (study 2), the flight performance was used as a criterion to measure quality. Various questionnaires were handed to the subjects after all user studies. These included subjective standardized questionnaires (e.g. NASA-TLX, SART) and questionnaires specifically designed for the evaluation of this concrete trial.

6.6. Conclusion

This chapter with its four user studies demonstrated the capabilities of the VRFS for the evaluation of HMI components in the cockpit. With the modular architecture of the system, the technologies that were subject of the studies were implemented with low effort and in a short amount of time. With the integrated human factors methods, the research questions of each study could be addressed successfully. Study 1 even indicated that some aspects of the human behavior, like the eye scanning pattern and the heart rate data, are similar to user trials in a conventional full flight simulator. The user studies demonstrated that the VRFS is a viable tool that can support evaluation efforts in different phases of the design process.

7. Discussion and Future Work

In this research, a Virtual Reality Flight Simulator for human factors engineering was developed based on use cases that stem from prototypes and experience from the flight deck development process. In a study, the fidelity of this virtual environment was evaluated in comparison to a conventional hardware simulator. In four user studies, which represent different types of human factors engineering studies in different phases of the development process, the practical application of the virtual environment was demonstrated. In this chapter, the results of the fidelity assessments and the implications of the user studies will be discussed. Based on the knowledge gathered, recommendations for the design of virtual reality flight decks and experiments to evaluate HMI components in this environment will be derived. Future research to enhance the system and extend it to new fields of application will be discussed at the end of this chapter.

7.1. Simulator Fidelity

The comparative studies that were conducted in Chapter 5 evaluated two types of simulator fidelity: task fidelity and cognitive-psychological fidelity.

The evaluation of the task fidelity showed that it is possible to conduct a realistic operational scenario while interacting with the mixed mock-up, i.e. interact with hardware cockpit controls while beeing immersed in the virtual environment. Yet, the time needed to interact with these elements is significantly longer compared with a conventional hardware simulator. The interaction with fully virtual control elements–control elements that are triggered without any haptic feedback–takes even more time in the virtual world. Despite this additional amount of time for interactions, it is still possible to reliably operate the virtual aircraft. The presented user study on a novel system management HMI (study 3 in Section 6.3) consisted of a complex operational scenario with numerous time critical interactions in the overhead panel. Despite the challenging and time-consuming interaction with virtual control elements, all pilots were able to successfully complete the given task (Dreyer and Oberhauser, 2016a).

The cognitive-psychological fidelity is also degraded in the virtual environment. The evaluation revealed a lower level of situation awareness that leads to a degraded performance when conducting a manual flying task. Nevertheless, the pilots were able to safely and reliably complete the given flight tasks that were presented in the evaluation in Chapter 5 and the user studies in Chapter 6. The degradation of flight performance is significant but not critical to complete the task. The information gathering behavior of the pilots is similar in both a virtual and conventional environment with a single flight display. This is demonstrated by a similar eye-tracking pattern in the head-up display evaluation (see Study 1 in Section 6.1). However, gathering information from multiple displays and the outside visual leads to a different behavior: An increased head movement activity was found while pilots conducted the head-up/head-down scan (see Section 5.4) in the virtual environment. This behavior can be attributed to the degraded field of view of the head-mounted display.

7.2. Human Factors Engineering

The user studies in Chapter 6 show the practical use of the VRFS in the flight deck development process. They also demonstrate the practical application of the integrated features. These are methods for integrating HMI components, designing new cockpit layouts, setting up and controlling the experiment as well as using the integrated human factors methods.

The evaluation in Chapter 5 and the user studies in Chapter 6 demonstrate the development, adaption, and integration of novel cockpit components. The flexibility of the virtual cockpit enables the fast design and implementation of new cockpit layouts and the rapid integration of HMI components. With the presented import pipeline, novel designs based on CAD data, commonly used by the industry, can be integrated quickly and cost-effectively. Depending on the particular use case and the type of evaluation, some additional hardware elements are necessary. Simple, non-functional plates or 3D printed elements can help provide haptic feedback. Functional elements like flight sticks, button panels or levers can either be purchased from gaming manufacturers or can be built by using widely available micro-controller systems and model making techniques. This enables the demonstration of new cockpit layouts and concepts in a fast and cost-effective way. The integration of single HMI components for their demonstration or evaluation in an existing virtual cockpit is another main purpose of the system. With the Robot Operating System, different types of single-system prototypes can be integrated into the system: easy-to-use interfaces for data, video, and audio are provided for developers.

Jorna and Hoogeboom (2004) state that:

> In case the real world or context contains factors that can change the behavior of either the system or the human pilot, a discrepancy between simulated and real world will be present (p. 257).

To avoid this discrepancy, all presented trials have been designed as comparative studies. With this approach, confounding factors that stem from the simulation environment can be minimized (Oberhauser, Dreyer, Mamessier, et al., 2015). Thus, the used human factors methods deliver relative within-subject results gathered in the Virtual Reality Flight Simulator that cannot easily be transferred to other simulation environments or to real life cockpits. This research provides initial indications to transfer these results to other environments with the comparative evaluation presented in Chapter 5. Yet, accurate statements for each particular human factors method cannot be provided as the collected metrics, based on subjective questionnaires like workload or situation awareness, highly depend on the given scenario and are only useful in comparison with a controlled experiment. Objective metrics like the heart rate did not show a significant difference between the conventional and virtual simulator. Still, the user studies in Chapter 6 show the general viability of some human factors methods. The eye tracking data show the same scanning pattern in the virtual environment as in a conventional simulator (see study 1 in Section 6.1). The heart rate measurement in the same study shows a similar arousal during a high-workload go-around situation in the two environments.

Additional human factors methods will be, or already have been integrated into the system, in particular in the scope of the I-Vision project: Mamessier, Dreyer, and Oberhauser (2014) demonstrated the use of the Situation Awareness Global Assessment Technique (SAGAT), an objective, quantitative method for assessing situation awareness; Mamessier and Feigh (2016) presented the use of a semi-automated Hierarchical Task Analysis (HTA); Käfer, Harth, and Mamessier (2016) demonstrated the integration of semantic web technologies for flight deck design into the virtual cockpit environment.

7.3. Design Recommendations

7.3.1. Cockpit Design

When designing and implementing a three-dimensional cockpit, usually, the available resources are limited. This leads to a trade-off between a focus on visual quality (textures, surface effects, level of detail) and functionality (control elements, flight displays). In order to evaluate the relevance of these two factors, post-trial interviews with pilots that used the Virtual Reality Flight Simulator were conducted. They show that for the majority of pilots the functionality of the cockpit and the simulation are more important than the visual quality.

Design Recommendation 1 *Functionality above visual quality: The focus of attention when designing a cockpit should lie on the remodeling of flight displays, the underlying system simulation, and the functionality of buttons, switches, and levers; "eye candy" in the cockpit is of second priority.*

In a high-workload environment with a focus on a specific task, secondary stimuli like photo-realistic cockpit textures or a detailed outside visual are subordinate. W. F. Moroney and B. W. Moroney (2009) observed this phenomenon in traditional flight simulators as well:

> Despite the loss in perceptual fidelity, the illusion is compelling and becomes "virtual reality" when the pilot becomes involved in his or her flying tasks (p. 19-13).

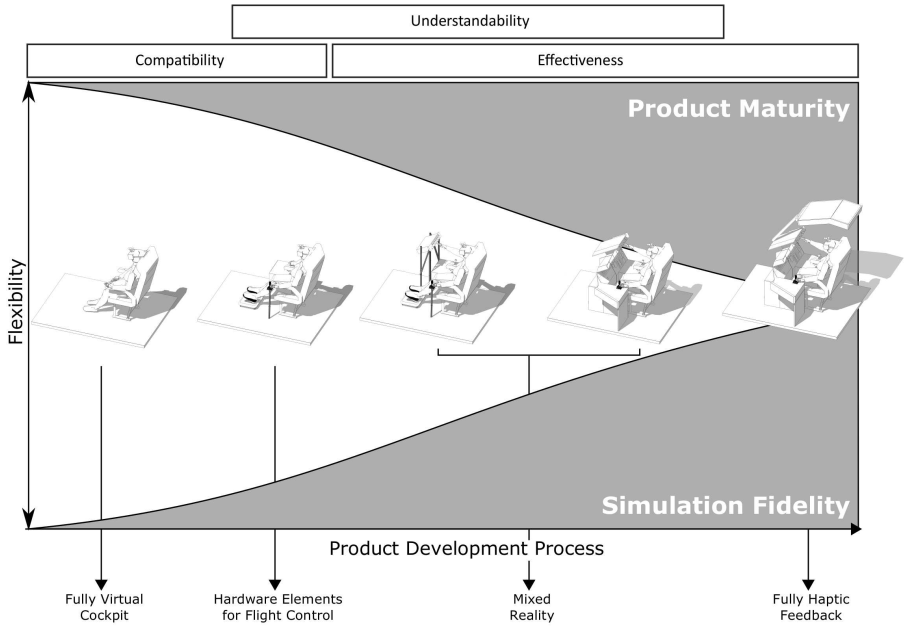


Figure 7.1.: The mixed mock-up in the product development process (Oberhauser, Dreyer, Convard, et al., 2016).

7.3.2. Interaction Design

The usability of a fully virtual cockpit highly depends on its design. In particular virtual interactions with numerous closely packed buttons can and will lead to less accurate interactions and an increased task completion time. The interaction with levers, rotary knobs or touchscreens in a virtual environment is even more challenging as it requires high accuracy and complex gestures. As the cockpit cannot and should not be redesigned for the particular use in the virtual environment, a mixed mock-up approach with hardware control elements that can be triggered from within the virtual environment is preferable. Yet, with every hardware element that is added to the simulator, the level of flexibility decreases whereas the required effort increases. The mixed mock-up approach should be chosen carefully and in correspondence with the respective level of evaluation and maturity of the cockpit. As the cockpit gets more mature during the development, more hardware elements can be added to the environment and the simulator can grow from a virtual demonstration of the cockpit geometry to a fully functional mixed mock-up, as shown in Figure 7.1.

Besides the flexibility of the cockpit, the research objective should be considered: If the usability evaluation of a certain HMI component is the subject of an evaluation, this component should be provided with a high interaction fidelity. Hence, it should preferably be a mixed mock-up element, in particular if it requires complex or time critical interactions or if it is a control element that features specific haptics.

Design Recommendation 2 *Use as many mixed mock-up hardware elements as necessary without compromising the flexibility of the virtual environment or the research objective.*

If virtual interaction is necessary, for example controlling a fully virtual overhead panel, complex gestures should be avoided. Instead, push interactions with virtual buttons can also be adapted to drive rotary knobs, levers or other elements.

Design Recommendation 3 *Design simple virtual push interactions instead of complex gestures.*

Furthermore, if buttons are closely packed, depending on the scenario, the control elements that are not in use can be deactivated. Erroneous interactions can be expected but they should not affect the system simulation.

Design Recommendation 4 *Provide virtual interaction functionality only for those buttons that are necessary for the scenario.*

7.3.3. Experiment Design

The simulator sickness evaluation in Chapter 5 shows a significant impact of the virtual environment on the users. Yet, in all experiments including the evaluation and the user studies with a total of 94 participants, only one pilot had to terminate the session due to severe symptoms of simulator sickness. The fact that the exposure in all these user trials was limited to a maximum of 15 minutes might have contributed to this rather small termination rate. Although no study dedicated to the length of exposure was conducted, experience indicates that this is a reasonable limit. However, experience shows that it is possible to have several consecutive sessions on a single day with sufficient breaks between the immersive phases.

Design Recommendation 5 *Limit the exposure time in the virtual environment to a maximum of* 15 minutes *per session.*

The confounding factors of the virtual environment are diverse and hard to quantify in general. Only the time to interact in the virtual environment, the task completion time, and the movement time might be transferable, with a multiplying factor, to a conventional flight simulator. This research only offers initial data with regard to identifying these confounding factors. Hence, comparative evaluations, i.e. evaluation of a novel HMI component in comparison with a legacy system, both in the VRFS, are considered as good practice.

Design Recommendation 6 *Conduct comparative evaluations in the VRFS to minimize confounding factors that stem from the virtual environment.*

The evaluation of the interaction performance in the virtual environment shows a significantly longer time to reach mixed mock-up components and fully virtual buttons. On average, it took up to three seconds longer to reach these control elements in the virtual environment. Hence, in the design of the scenario, this degradation has to be considered. A time critical scenario has to be designed in a way that it still can be conducted successfully despite this slower interaction time.

Design Recommendation 7 *Set the pace of a scenario in a way that makes it feasible with the degraded interaction performance.*

7.4. Future Work

7.4.1. Progress in Virtual Reality Hardware

The current interest in virtual reality technology initiated by the development of the Oculus Rift head-mounted display led to a dynamic market where new HMDs and tracking systems emerge on a regular basis (Hamburger, 2014). Two major advantages are expected by this technological advance that the Virtual Reality Fight Simulator will benefit from:

High field of view and ultra high field of view devices with high pixel densities are currently under development and will be available in the future. These devices can reduce the limitations with regard to situation awareness of the currently used HMD but still provide a sufficient resolution in order to use the flight displays and other cockpit elements. Rendering on these high-resolution displays and still keeping a high frame rate is a challenge for further research.

The advances in accurate tracking systems for the consumer market might be able to deliver a system that has an accuracy comparable to the ART tracking system that is used in the typical setup of the VRFS, yet, at only a fraction of its price. This would make the VRFS more affordable and accessible without compromising its usability.

7.4.2. Motion

The history of flight simulators is closely connected to motion platform systems. These motion platforms induce accelerations and add stimuli to the operator's vestibular system. Motion platforms are mandatory for certified level A-D full flight simulators (European Aviation Safety Agency, 2012). Yet, there is an ongoing dispute whether these systems are necessary for all types of flight simulators (Bürki-Cohen, Soja, and Longridge, 1998). With the current setup of the VRFS using a reference target for the positional and orientational tracking of head and fingers, the foundations to add motion already exists. If the reference target and all other targets are placed on the platform, the movement of the platform is felt by the user but is not visible in the virtual environment (Dreyer and Oberhauser, 2016b).

Figure 7.2.: A prototype of the Virtual Reality Flight Simulator on a motion platform. Alternatively the system can be used without head-mounted display, but instead with displays for the outside visual and head-down instruments.

A first prototype of the Virtual Reality Flight Simulator on a moving platform was developed in the scope of this research as shown in Figure 7.2. The system is based on a consumer motion platform that comes with a connection to the X-Plane flight simulation. The structure that holds mixed mock-up flight displays, a chair, the reference target, and primary flight controls was realized using Bosch Rexroth aluminum profiles.

Further research has to be conducted on the influence of motion stimuli in virtual environments, in particular the influence on simulator sickness. In addition, the fidelity of the motion stimuli has to be enhanced and evaluated.

7.4.3. Spatial Sound

Audio cues are important to operate an aircraft, the absence or inaccuracies of these cues can lead to problems as Jarvis, Lalonde, and Spira (2008) state:

> Inaccurate audio cues can have a subtle, yet pervasive influence on pilots' perception of handling qualities (p. 4).

The stereo audio that is provided by X-Plane is already present in the VRFS. In future research, spatial sound cues could be implemented by using 3D audio headphones or a surround sound system in the VRFS to increase immersion.

7.4.4. Training

In the patent for the first virtual reality device in history, Heilig (1962) already pointed out the possibility of using it as a training device:

> There are increasing demands today for ways and means to teach and train individuals without actually subjecting the individuals to possible hazards or particular situations. (...) Industry, on the other hand, is faced with a similar problem due to present day rapid rate of development of automatic machines. Here, too, it is desired to train a labor force without the accompanying risks (p. 1).

Flight simulators were, and still are, for the most part used for pilot training. Hence, it is obvious that the VRFS developed in this research can be used for training as well. Yet, at the current state of development, the VRFS with its drawbacks in interaction and perception does not meet the level of fidelity that would be necessary to replace a Full Flight Simulator (FFS) used for the training of civil flight crews. Full flight simulators (Levels A-D) have to fulfill strict requirements by the European Aviation Safety Agency (2012) and other authorities. For example, an exact replica of the cockpit, including the function and haptic feedback of each control element, is required in these ultra realistic simulators.

(a) Oculus Rift (b) Google Cardboard

Figure 7.3.: The VRFS on Oculus Rift (left) and on Google Cardboard (right).

Procedure Trainer

Other Flight Training Devices (FTDs) such as procedure trainers do not fall under these strict requirements. Today, these procedure trainers consist of simplified cockpit replicas, panels represented on touchscreens, or are paper based. They are used to get familiar with the cockpit layout and train normal and abnormal procedures. Bauer and Klingauf (2008) already presented and evaluated the prototype of a virtual reality environment as a procedure training device with promising feedback from operational pilots.

The virtual environment offers the opportunity to change the cockpit environment in an instance, which makes multiple hardware FTDs for every type of aircraft obsolete. In addition, more training related information, like highlighting procedure flows or the presentation of additional system information can be included in the system. Some of these technologies have been investigated as part of the I-Vision project, which used parts of the VRFS presented in this research (Mamessier, Dreyer, and Oberhauser, 2015).

Personal Training Device

With today's technological advances in the consumer industry, a flight training device could be provided at such a low price that it could even be used as a personal training device. Available low-cost virtual reality headsets consist of a frame that holds two lenses and a regular smartphone that acts as the display–a concept that was first presented by Google as a simple do-it-yourself cardboard kit (Lyons, 2016).

Whether it is a cardboard-like device or a consumer headset, both can be distributed to pilots and student pilots. With this kind of personal training device, procedure training or cockpit familiarization could be trained at any time and place. This could lead to a reduction of on-site procedure training whilst maintaining or even exceeding the overall training time for practical procedure training.

The VRFS has been demonstrated using a consumer virtual reality headset as shown in Figure 7.3a; a simplified version of a virtual cockpit has been developed for the use with Google's cardboard in the scope of the I-Vision project (see Figure 7.3b).

Remote Pilot(s) or Instructor

Pilots are on duty worldwide often not stationed near the airline's main airport. Bringing together a flight crew and an instructor at a place with adequate simulator facilities can be a difficult task. Thus, remote virtual collaboration can be a solution for this problem.

Pilots or instructors equipped with personal training devices or in different stationary virtual reality flight training devices can collaborate in a remote shared virtual space. The participants can see and interact with each other via 3D mannequins similar to the virtual collaboration presented in Section 4.3.4.

8. Conclusion

Using virtual reality environments for engineering tasks, in particular in flight deck development, is common practice. With these systems, ergonomic or visibility aspects can be evaluated among others. In comparison with these static environments, the Virtual Reality Flight Simulator presented in this research enables the evaluation of physiological, cognitive, and behavioral aspects of the flight in a simulated operational environment. With this capability and the flexibility of the VRFS, it is a valuable addition to the already available simulator environments in the flight deck development process.

The evaluation in Section 5 and the user studies in Section 6 showed that the main expected contribution of this research, i.e. developing a Virtual Reality Flight Simulator, was met. Both, the evaluation and the user studies showed that the environment provides a sufficient fidelity to conduct complex operational tasks. Yet, it also revealed limitations: The task fidelity, i.e. the ability to recreate a task in the simulator, suffers in particular when interacting with fully virtual buttons. The time needed to reach and trigger these control elements in the virtual environment is significantly higher than in a conventional simulator. This effect is reduced if a mixed mock-up approach is used, yet, it still is significant. In addition, the cognitive-psychological fidelity is degraded in the virtual environment. The limited field of view of the used head-mounted display is for the most part responsible for this limitation. This fidelity evaluation, the second contribution of this research, will help to further enhance the system and to design user studies in a way that they are appropriate for the specific requirements of the virtual environment.

With these limitations in mind, the user studies presented in Chapter 6 demonstrate the practical application of the VRFS a and the integrated human factors methods–the fourth contribution of this research. The presented studies show a broad spectrum of technology maturity levels and experiment setups in the flight deck development process. All studies have in common that the implementation and experiment were conducted with relatively low effort and in a rapid prototyping manner. The use of the Robot Operating System, an open source network framework initially developed for robotic applications, proved to be a good design choice in order to make the system modular and accessible. It enables the use of basic network functionality as well as enhanced visualization and monitoring tools without reinventing the wheel (Rönnau, 2016). For the VRFS, it makes the integration of new components like system or display prototypes straightforward–which is the third contribution.

This thesis started with a quote by Frank L. Wright: "You can use an eraser on the drafting table or a sledge hammer on the construction site." The use of virtual reality flight simulation in the flight deck development process provides the cockpit designer with a "drafting table" that is functional, immersive, and still highly flexible. The availability of this low-cost, medium-fidelity flight deck environment encourages a user centered, rapid prototyping design process, which allows to conduct more user studies, with more pilots on more design variants compared with conventional flight simulators. Subject matter experts can dive into this virtual cockpit to detect design flaws, optimize workflows or test and enhance preliminary ideas with low effort. Thus, the Virtual Reality Flight Simulator and other virtual reality product engineering methods can decrease the need for a "sledge hammer" and at the same time increase the usability and safety of flight decks.

Appendix

Appendix A.

Use Case Descriptions

1 Pilot Use Case

Number	**1.1**
Name	**Trigger a Virtual Button**
Goal	The user shall be able to push a fully virtual button and get visual feedback regarding the state of the button.
Precondition	The simulation system and X-Plane is up and running. The hand is fully tracked.
Postcondition	The button has changed its indication
Actors	Pilot
Scenario	1. The user moves his/her finger to the virtual position of the button. 2. The user's hand turns green while colliding with the button. 3. The button changes its indication to a different color or texture.
Alternative Scenario	02a. While colliding with the button the button geometry is translated. 03a. Eventually the button indication is changed by the flight simulation system to a different indication.
Non-Functional Requirements	The collision with the button geometry should be easy to accomplish for the user. It should be error tolerant. No double executions should be possible.

Number	**1.2**
Name	**Trigger a Guarded Virtual Push-button**
Goal	The user shall be able to push a fully virtual button that is covered by a protective cap and get visual feedback regarding the state of the button.
Precondition	The simulation system and X-Plane is up and running. The hand is fully tracked.
Postcondition	The button has changed its indication.
Actors	Pilot
Scenario	1. The user monitors the state of a protected button and notices a change of its indication. 2. The user moves his finger to the virtual position of the protected button. 3. After the collision the cap opens with a fast animation. 4. The user moves his/her finger to the position of the button. 5. The users hand turns green while colliding with the button. 6. The button changes its indication to a different color or texture.
Alternative Scenario	04a. While colliding with the button the button geometry is translated. 06a. Eventually the button indication is changed by the flight simulation system to a different indication. 06b. If the open cap is touched it should close once again.
Non-Functional Requirements	The collision with the button geometry should be easy to accomplish for the user. It should be error tolerant. No double executions should be possible.

Number	**1.3**
Name	**Trigger a Physical Push-button**
Goal	The user shall be able to push a physical button that has also a virtual representation and get visual feedback regarding the state of the button.
Precondition	The simulation system and X-Plane is up and running. The hand is fully tracked. A hardware device is placed in the exact spatial position as modelled in the virtual reality. This hardware device is functional and connected to the simulation.
Postcondition	The button has changed its indication.
Actors	Pilot
Scenario	1. The user moves his/her finger to the virtual position of the button. 2. The user feels the hardware button and pushes it. 3. The button changes its indication to a different color or texture.
Alternative Scenario	02a. While colliding with the button the button geometry is translated. 06a. Eventually the button indication is changed by the flight simulation system to a different indication.
Non-Functional Requirements	The collision with the button geometry should be easy to accomplish for the user. It should be error tolerant. No double executions should be possible.

Number	**1.4**
Name	**Use a Hardware lever**
Goal	The user shall be able to use a lever in the cockpit and see the change of its position or orientation interactively.
Precondition	The simulation system and X-Plane is up and running. The hand is fully tracked. A Hardware device is placed in the exact spatial position as modelled in the virtual reality. This hardware device is functional and connected to the simulation.
Postcondition	The lever has changed its position and the state of the simulation has changed
Actors	Pilot
Scenario	1. The user moves his finger to the virtual position of the lever. 2. The user feels the hardware lever and changes its position. 3. The animation of the lever is visible to the user and it is aligned with the hand geometry. 4. The system that is controlled by the lever is being manipulated.
Alternative Scenario	
Non-Functional Requirements	The collision with the button geometry should be easy to accomplish for the user. It should be error tolerant.

Number	**1.5**
Name	**Use a Flight Display**
Goal	The user shall be able to get information of the flight status from a display interactively.
Precondition	The simulation system and X-Plane is up and running.
Postcondition	The user knows parameters about his current flight status
Actors	Pilot
Scenario	1. The user looks at a display. 2. The user changes aircraft systems or the flight path using buttons or levers. 3. The impact of these actions is visible on the display immediately.
Alternative Scenario	02a. The user touches a part of the display to change its state (Touchscreen). 01a. The user has a virtual display in his hand, simulating an Electronic Flight Bag with touch screen functionality. The device may be fully virtual or may be connected to a tracked hardware dummy.
Non-Functional Requirements	The information should be presented with a high frame rate and low latency.

Number	**1.6**
Name	**Interact with a Second Pilot**
Goal	The user shall be able to see and interact with a second pilot in the cockpit.
Precondition	The simulation system and X-Plane is up and running. A second user with a second HMD is present.
Postcondition	The user knows the position and current intention of the second pilot.
Actors	Pilot
Scenario	1. The user looks to the side. 2. The user sees a mannequin of the other pilot. 3. The other pilot also sees a mannequin of the other pilot.
Alternative Scenario	03a. One of the pilots passes a mixed reality object to the other pilot.
Non-Functional Requirements	The mannequin should be an accurate representation of a pilot.

2 Cockpit Designer/Engineer Use Case

Number	**2.1**
Name	**Import Cockpit Geometry**
Goal	The developer shall be able to use geometry for the virtual cockpit in commonly used 3D formats.
Precondition	3D data is available in a commonly used CAD or polygonal file format.
Postcondition	The 3D data is textured and integrated into the VRFS.
Actors	Cockpit Designer/Engineer
Scenario	1. The engineer converts 3D data of cockpit geometry or a component. 2. The engineer textures the 3D geometry. 3. The textured geometry is integrated and visible into the VRFS.
Alternative Scenario	
Non-Functional Requirements	The process should be as effective and easy as possible.

Number	**2.2**
Name	**Animate Objects**
Goal	The developer shall be able to bind a variable to an animated object.
Precondition	Textured 3D geometry is present and it is properly structured.
Postcondition	The 3D geometry is moving according to a variable that is provided by the simulation.
Actors	Cockpit Designer/Engineer
Scenario	1. Add an axis to the 3D geometry (rotation or translation). 2. Specify the binding between the axis and the geometry. 3. Test the movement of the animation.
Alternative Scenario	1a. Define multiple axes for a 3D geometry (considering relative movement).
Non-Functional Requirements	The binding should be user friendly and fast. Binding and testing could take place in one application.

Number	**2.3**
Name	**Assign Dynamic Textures**
Goal	The developer shall be able to map a dynamic texture to a mesh/object.
Precondition	A textured 3D geometry is loaded and is properly structured. A dynamic texture stream is available.
Postcondition	A dynamic texture is mapped to a 3D geometry object or mesh.
Actors	Cockpit Designer/Engineer
Scenario	1. The developer chooses a visual input (X-Plane texture, VNC stream, video stream). 2. Mapping of dynamic texture on a mesh or 3D geometry. 3. Preview of the dynamic texture in the development application.
Alternative Scenario	
Non-Functional Requirements	

Number	**2.4**
Name	**Integrate Prototypes**
Goal	The developer shall be able to integrate an external component/prototype.
Precondition	The virtual cockpit is functional and ready to use.
Postcondition	An additional system (System simulation, interactive display, touch display) is added to this cockpit and functional.
Actors	Cockpit Designer/Engineer
Scenario	1. The developer chooses a position for the prototype in the virtual cockpit. 2. The developer maps an external display stream to this part of the cockpit. 3. On the prototype a VNC server has to be installed.
Alternative Scenario	1a. The developer deactivates a part of the system simulation.
Non-Functional Requirements	

3 Controller Use Case

Number	**3.1**
Name	**Load a Scenario.**
Goal	The supervisor shall be able to load a preconfigured scenario.
Precondition	The simulation system and X-Plane is up and running.
Postcondition	The scenario is loaded and the user may proceed by starting it.
Actors	Controller
Scenario	1. The experiment controller uses the scenario management GUI in order to load a scenario file.
Alternative Scenario	01a. The user can load the corresponding scenario file via command line interface as a back-up means in case of GUI unavailable or with the built-in X-Plane functionality.
Non-Functional Requirements	The GUI should have a common desktop applications look-and-feel, such as an ‘Open File’ dialog

Number	**3.2**
Name	**Induce a System Malfunction**
Goal	The controller shall be able to induce a predefined behavior of the aircraft that simulates a system malfunction (e.g. engine fire).
Precondition	The simulation system and X-Plane is up and running. A scenario is running.
Postcondition	The simulation behaves accordingly.
Actors	Controller
Scenario	1. The experiment controller uses the scenario management GUI in order to induce a previously defined system malfunction.
Alternative Scenario	01a. The user can execute the system malfunction via command line interface as a back-up means in case of GUI unavailable.
Non-Functional Requirements	The GUI should have a common desktop applications look-and-feel.

4 Human Factors Practitioner

Number	**4.1**
Name	**Online Analysis of Data**
Goal	The HF practitioner shall be able to create a component that analyzes simulation data or data about human behavior during run-time.
Precondition	The simulation system and X-Plane is up and running. A user is present.
Postcondition	The online analysis tool calculated metrics from sensors.
Actors	HF Practitioner
Scenario	1. The HF practitioner defines a script that calculates metrics from sensor inputs and telemetry data. 2. The script is connected to the ROS network and computes the defined metrics during run-time.
Alternative Scenario	
Non-Functional Requirements	

5 Shared Use Cases

Number	**5.1**
Name	**Access Simulation Data**
Goal	The controller and the HF practitioner shall get access (read and/or write) to a variable that is available in the VRFS.
Precondition	The simulation system and X-Plane is up and running.
Postcondition	The value of the variable is retrieved.
Actors	Controller, HF Practitioner
Scenario	1. The controller or HF practitioner looks up the available variables in the VRFS. 2. The controller or HF practitioner selects a variable and retrieves the value.
Alternative Scenario	
Non-Functional Requirements	

Number	**5.2**
Name	**Record Simulation Data**
Goal	The controller and HF practitioner shall be able to record all data that is available in the VRFS.
Precondition	The simulation system and X-Plane is up and running.
Postcondition	All recorded simulation data and other measures are stored at their specific locations. The data is timestamped
Actors	Controller, HF Practitioner
Scenario	1. The controller or HF practitioner starts the recording
Alternative Scenario	1a. The controller or HF practitioner selects a subset of variables before recording the data.
Non-Functional Requirements	

Number	**5.2**
Name	**Integrate Human Factors Methods**
Goal	The engineer in cooperation with the HF practitioner shall be able to integrate HF methods.
Precondition	The simulation system and X-Plane is up and running.
Postcondition	A human factors method is integrated into the VRFS.
Actors	Controller
Scenario	1. The experiment supervisor uses the scenario management GUI in order to pause a scenario
Alternative Scenario	01a. The user can pause running scenario via command line interface as a back-up means in case of GUI unavailable
Non-Functional Requirements	The GUI should have a common desktop applications look-and-feel, such as an ‘Pause Scenario’ button

Number	**5.3**
Name	**Continue a Scenario**
Goal	The user shall be able to continue a paused scenario.
Precondition	The simulation system and X-Plane is up and running. A scenario is on pause.
Postcondition	The simulation unfreezes; simulation data and other measures recordings are continued at the last position (append). State machines and storyboards are continued with the last value. The visible VR content for the Pilot View is re-set to the initial status within the scenario file.
Actors	Controller
Scenario	1. The experiment supervisor uses the scenario management GUI in order to continue a paused scenario
Alternative Scenario	01a. The user can continue paused scenario via command line interface as a back-up means in case of GUI unavailable
Non-Functional Requirements	The GUI should have a common desktop applications look-and-feel. The continuation shall be possible either by pressing the ‘Play Scenario’ button, or by de-selecting the ‘Pause Button’ again.

Number	**3.4**
Name	**Stop a Scenario**
Goal	The user shall be able to stop a running scenario at any time.
Precondition	The simulation system and X-Plane is up and running. A scenario is running.
Postcondition	The simulation stops; simulation data and other measures recordings are stopped. The visible VR content for the Pilot View is set to a pre-defined 'end status' (e.g. black screen). The scenario management GUI automatically launches the 'Save recorded data' dialog.
Actors	Controller
Scenario	1. The experiment supervisor uses the scenario management GUI in order to stop a running scenario
Alternative Scenario	01a. The user can stop a running scenario via command line interface as a back-up means in case of GUI unavailable
Non-Functional Requirements	The GUI should have a common desktop applications look-and-feel. The Stop command shall be executed via a 'Stop scenario' button.

Number	**3.6**
Name	**Induce human error**
Goal	The user shall be able to induce a pre-defined behavior of the virtual co-pilot that simulates a human error.
Precondition	The simulation system and X-Plane is up and running. A scenario is running. A virtual co-pilot is running.
Postcondition	The virtual co-pilot executes the commanded error and the simulation behaves accordingly.
Actors	Cockpit Designer/Engineer, HF Practitioner
Scenario	1. The engineer and the HF practitioner identifies the needs of the HF methods. 2. Depending on the method, the engineer uses the capabilities of ROS to integrate the method into the VRFS. 3. The HF practitioner uses the method to assess the pilot's performance or behavior.
Alternative Scenario	
Non-Functional Requirements	

Appendix B.

Questionnaires Evaluation Trial

Biographischer Erhebungsbogen **CODE_____ Reihenfolge __________**

1. Geschlecht ☐ männlich ☐ weiblich

2. Geburtsjahr ___________________________________

3. Größe ___________________________________

4. Händigkeit ☐ rechtshändig ☐ linkshändig ☐ beidhändig

5. Benutzen Sie beim Fliegen eine Sehhilfe?

☐ nein ☐ ja Wenn ja, welche? ________________

6. Haben Sie heute bereits Medikamente genommen? nein ☐ ja ☐

Wenn ja, welche und wie viel? _______________________________________

7. Ausbildung (letzter Schulabschluss, Studium) __________________________

8. Beruf ____________________________________

9. Fluglizenzen, Berechtigungen ____

10. Berufliche Ausübung einer fliegerischen Aktivität ☐ nein ☐ ja, welche:______________

Seit wann üben Sie Ihre berufliche fliegerische Tätigkeit aus? _________________

1. Absolvierte Flugstunden ohne Simulatorstunden (ungefähr)

	Anzahl Flugstunden der letzten 90 Tage	Anzahl Flugstunden Total
Gesamte Flugstunden ohne Simulator		

2. Erfahrungen mit Flugsimulationen

Simulator	Anzahl Simulatorstunden	
	letzte 90 Tage	Total

3. Erfahrungen mit Virtual Reality ☐ ja ☐ nein

Wenn ja, welches System? _______________________________________

Fragebogen **nach den Platzrunden** im Virtual Reality Flugsimulator

Mit den folgenden Skalen sollen Sie die **geistigen Anforderungen** der soeben durchgeführten Flugaufgaben beurteilen.

sehr niedrig	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr hoch

Mit den folgenden Skalen sollen Sie die **körperlichen Anforderungen** der soeben durchgeführten Flugaufgaben beurteilen.

sehr niedrig	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr hoch

Mit den folgenden Skalen sollen Sie die **zeitlichen Anforderungen** der soeben durchgeführten Flugaufgaben beurteilen.

sehr niedrig	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr hoch

Mit den folgenden Skalen sollen Sie die **Leistung** beurteilen, die Sie bei den soeben durchgeführten Flugaufgaben erzielt haben.

Misserfolg	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	perfekter Erfolg

Mit den folgenden Skalen sollen Sie die **Anstrengung** beurteilen, die Sie bei den soeben durchgeführten Flugaufgaben aufgewendet haben.

sehr gering	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr groß

Mit den folgenden Skalen sollen Sie beurteilen, wie hoch der Grad der **Frustration** war, den Sie bei den soeben durchgeführten Flugaufgaben empfunden haben.

sehr niedrig	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr hoch

Fragebogen **<u>nach den Ausweichszenarien</u>** im Virtual Reality Flugsimulator

Mit den folgenden Skalen sollen Sie die **geistigen Anforderungen** der soeben durchgeführten Flugaufgaben beurteilen.

sehr niedrig	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr hoch

Mit den folgenden Skalen sollen Sie die **körperlichen Anforderungen** der soeben durchgeführten Flugaufgaben beurteilen.

sehr niedrig	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr hoch

Mit den folgenden Skalen sollen Sie die **zeitlichen Anforderungen** der soeben durchgeführten Flugaufgaben beurteilen.

sehr niedrig	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr hoch

Mit den folgenden Skalen sollen Sie die **Leistung** beurteilen, die Sie bei den soeben durchgeführten Flugaufgaben erzielt haben.

Misserfolg	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	perfekter Erfolg

Mit den folgenden Skalen sollen Sie die **Anstrengung** beurteilen, die Sie bei den soeben durchgeführten Flugaufgaben aufgewendet haben.

sehr gering	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr groß

Mit den folgenden Skalen sollen Sie beurteilen, wie hoch der Grad der **Frustration** war, den Sie bei den soeben durchgeführten Flugaufgaben empfunden haben.

sehr niedrig	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr hoch

SSQ Virtual Reality Flugsimulator

Bitte füllen Sie folgenden Fragebogen aus, der verschiedene Symptome beschreibt, die bei der Benutzung von virtuellen Umgebungen auftreten können. Bitte geben Sie an, ob und gegebenenfalls wie stark die folgenden Symptome auf Ihren Zustand zutreffen.

		gar nicht	etwas	mittel	stark
1	Allgemeines Unwohlsein	0	1	2	3
2	Ermüdung	0	1	2	3
3	Kopfschmerzen	0	1	2	3
4	überanstrengte Augen	0	1	2	3
5	Schwierigkeiten scharf zu sehen	0	1	2	3
6	Erhöhte Speichelfluss	0	1	2	3
7	Schwitzen	0	1	2	3
8	Übelkeit	0	1	2	3
9	Konzentrationsschwierigkeiten	0	1	2	3
10	Druckgefühl im Kopf	0	1	2	3
11	verschwommenes Sehen	0	1	2	3
12	Schwindel (Augen offen)	0	1	2	3
13	Schwindel (Augen geschlossen)	0	1	2	3
14	Drehschwindel	0	1	2	3
15	Magen macht sich bemerkbar	0	1	2	3
16	Aufstoßen	0	1	2	3

Bitte bewerten Sie die Realitätsnähe der virtuellen **Flugsimulation**

Gesamteindruck	unrealistisch 1 2 3 4 5 6 realistisch
Visualisierungssystem	unrealistisch 1 2 3 4 5 6 realistisch
Umgebungsdarstellung	unrealistisch 1 2 3 4 5 6 realistisch
Anzeigeinstrumente	unrealistisch 1 2 3 4 5 6 realistisch
Bedienelemente	unrealistisch 1 2 3 4 5 6 realistisch
Andere Flugzeuge	unrealistisch 1 2 3 4 5 6 realistisch
Wie realistisch konnten Sie Entfernungen einschätzen?	unrealistisch 1 2 3 4 5 6 realistisch
Wie realistisch konnten Sie Geschwindigkeiten einschätzen?	unrealistisch 1 2 3 4 5 6 realistisch

Markieren Sie bitte mit einem „X“ die Aussage, die Ihrem Gefühl und Ihrer Meinung am besten entspricht

1. Wie finden Sie den virtuellen Flugsimulator?

extrem nützlich	nützlich	neutral	nicht nützlich	kontraproduktiv

2. Wenn Sie die Möglichkeit hätten, würden Sie den virtuellen Flugsimulator in Zukunft nutzen?

sicher	vielleicht	ich bin mir nicht sicher	nein

3. Würden Sie den virtuellen Flugsimulator anderen PilotInnen empfehlen? □ ja □ nein

4. Nennen Sie 2 Eigenschaften/Funktionen des virtuellen Flugsimulators, die Sie am Nützlichsten finden:

-
-

5. Nennen Sie 2 Eigenschaften/Funktionen des virtuellen Flugsimulators, die Sie am wenigsten nützlich oder sogar störend finden:

-
-

Anregungen, Mitteilungen:

Fragebogen nach den Platzrunden im konventionellen Flugsimulator

Mit den folgenden Skalen sollen Sie die **geistigen Anforderungen** der soeben durchgeführten Aufgaben beurteilen.

sehr niedrig	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr hoch

Mit den folgenden Skalen sollen Sie die **körperlichen Anforderungen** der soeben durchgeführten Aufgaben beurteilen.

sehr niedrig	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr hoch

Mit den folgenden Skalen sollen Sie die **zeitlichen Anforderungen** der soeben durchgeführten Aufgaben beurteilen.

sehr niedrig	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr hoch

Mit den folgenden Skalen sollen Sie die **Leistung** beurteilen, die Sie bei den soeben durchgeführten Aufgaben erzielt haben.

Misserfolg	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	perfekter Erfolg

Mit den folgenden Skalen sollen Sie die **Anstrengung** beurteilen, die Sie bei den soeben durchgeführten Aufgaben aufgewendet haben.

sehr gering	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr groß

Mit den folgenden Skalen sollen Sie beurteilen, wie hoch der Grad der **Frustration** war, den Sie bei den soeben durchgeführten Aufgaben empfunden haben.

sehr niedrig	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr hoch

Fragebogen nach den Ausweichszenarien im konventionellen Flugsimulator

Mit den folgenden Skalen sollen Sie die **geistigen Anforderungen** der soeben durchgeführten Aufgaben beurteilen.

sehr niedrig	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr hoch

Mit den folgenden Skalen sollen Sie die **körperlichen Anforderungen** der soeben durchgeführten Aufgaben beurteilen.

sehr niedrig	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr hoch

Mit den folgenden Skalen sollen Sie die **zeitlichen Anforderungen** der soeben durchgeführten Aufgaben beurteilen.

sehr niedrig	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr hoch

Mit den folgenden Skalen sollen Sie die **Leistung** beurteilen, die Sie bei den soeben durchgeführten Aufgaben erzielt haben.

Misserfolg	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	perfekter Erfolg

Mit den folgenden Skalen sollen Sie die **Anstrengung** beurteilen, die Sie bei den soeben durchgeführten Aufgaben aufgewendet haben.

sehr gering	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr groß

Mit den folgenden Skalen sollen Sie beurteilen, wie hoch der Grad der **Frustration** war, den Sie bei den soeben durchgeführten Aufgaben empfunden haben.

sehr niedrig	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	sehr hoch

SSQ nach dem Flug im konventionellen Flugsimulator

Bitte füllen Sie folgenden Fragebogen aus, der verschiedene Symptome beschreibt, die bei der Benutzung von virtuellen Umgebungen auftreten können. Bitte geben Sie an, ob und gegebenenfalls wie stark die folgenden Symptome auf Ihren Zustand zutreffen.

		gar nicht	etwas	mittel	stark
1	Allgemeines Unwohlsein	0	1	2	3
2	Ermüdung	0	1	2	3
3	Kopfschmerzen	0	1	2	3
4	überanstrengte Augen	0	1	2	3
5	Schwierigkeiten scharf zu sehen	0	1	2	3
6	Erhöhte Speichelfluss	0	1	2	3
7	Schwitzen	0	1	2	3
8	Übelkeit	0	1	2	3
9	Konzentrationsschwierigkeiten	0	1	2	3
10	Druckgefühl im Kopf	0	1	2	3
11	verschwommenes Sehen	0	1	2	3
12	Schwindel (Augen offen)	0	1	2	3
13	Schwindel (Augen geschlossen)	0	1	2	3
14	Drehschwindel	0	1	2	3
15	Magen macht sich bemerkbar	0	1	2	3
16	Aufstoßen	0	1	2	3

Bitte bewerten Sie die Realitätsnähe **der konventionellen Flugsimulation**

Gesamteindruck	unrealistisch 1 2 3 4 5 6 realistisch
Visualisierungssystem	unrealistisch 1 2 3 4 5 6 realistisch
Umgebungsdarstellung	unrealistisch 1 2 3 4 5 6 realistisch
Anzeigeinstrumente	unrealistisch 1 2 3 4 5 6 realistisch
Bedienelemente	unrealistisch 1 2 3 4 5 6 realistisch
Andere Flugzeuge	unrealistisch 1 2 3 4 5 6 realistisch
Wie realistisch konnten Sie Entfernungen einschätzen?	unrealistisch 1 2 3 4 5 6 realistisch
Wie realistisch konnten Sie Geschwindigkeiten einschätzen?	unrealistisch 1 2 3 4 5 6 realistisch

Appendix C.

Questionnaires User Study 1

ALICIA

ALICIA All Condition Operations and Innovative Cockpit Infrastructure

EADS

DBS Systems Engineering

Analysis of the Evaluation of the Use of an optimised Overhead-Panel and Head-Up Display-System with regard to Man-Machine-Interaction

#

CREW DATA - DURATION: appr. 1 Minute

AGE
a

WEIGHT
kg

HEIGHT
m

NHF
h

YOD
a

USABILITY OF HEAD-UP DISPLAYS

-- - 0 + ++

COMMENTS

#

NASA-TLX [NASA-Task Load Index]

MENTAL DEMAND

rel. importance 0 1 2 3 4 5

L H

PHYSICAL DEMAND

rel. importance 0 1 2 3 4 5

L H

TEMPORAL DEMAND

rel. importance 0 1 2 3 4 5

L H

PERFORMANCE

rel. importance 0 1 2 3 4 5

GOOD POOR

EFFORT

rel. importance 0 1 2 3 4 5

L H

FRUSTRATION

rel. importance 0 1 2 3 4 5

L H

ALICIA All Condition Operations and Innovative Cockpit Infrastructure

EADS

DBS Systems Engineering

Analysis of the Evaluation of the Use of an optimised Overhead-Panel and Head-Up Display-System with regard to Man-Machine-Interaction

#

SART [Situation Awareness Rating Technique]

INSTABILITY OF SITUATION
How changeable is the situation?
Is the situation very stable and straight forward [LOW]
or is it highly unstable and likely to change suddenly [HIGH]?

L H

COMPLEXITY OF SITUATION
How complicated is the situation?
Is it simple and straightforward [LOW]
or is it complex with many inter-related components [HIGH]?

L H

VARIABILITY OF SITUATION
How many variables are changing in the situation?
Are there very few variables changing [LOW]
or are there are large number of factors varying [HIGH]?

L H

AROUSAL
How aroused are you in the situation?
Do you have a low degree of alertness [LOW]
or are you alert and ready for activity [HIGH]?

L H

CONCENTRATION OF ATTENTION
How much are you concentrating on the situation?
Is your attention elsewhere [LOW]
or are you bringing all your thoughts to bear [HIGH]?

L H

#

SART [Situation Awareness Rating Technique]

DIVISION OF ATTENTION
How much is your attention divided in the situation?
Are you focussed on only one aspect of the situation [LOW]
or concentrating on many aspects [HIGH]?

L H

SPARE MENTAL CAPACITY
How much mental capacity do you have to spare in the situation?
Do you have nothing to spare at all [LOW]
or do you have sufficient to attend to many variables [HIGH]?

L H

INFORMATION QUANTITY
How much information have you gained about the situation?
Did you understood very little [LOW]
or have you received a great deal of knowledge [HIGH]?

L H

INFORMATION QUALITY
How good is the information you have gained about the situation?
Is it a new situation [LOW]
or is the knowledge communicated very useful [HIGH]?

L H

FAMILIARITY WITH SITUATION
How familiar are you with the situation?
Is it a new situation [LOW]
or do you have a great deal of relevant experience [HIGH]?

L H

Appendix D.

Questionnaires User Study 2

Zum Abschluss machen Sie bitte ein paar Angaben zu Ihrer Person.

Selbstverständlich werden alle Angeben vertraulich, personenungebunden und anonymisiert behandelt.

Bitte geben Sie Ihr Alter an: ________ Jahre

Ihre Körpergröße beträgt: ________ cm

Tragen sie eine Brille/Kontaktlinsen? nein ☐ ja ☐

Haben Sie eine Pilotenlizenz? nein ☐ ja ☐

Wenn ja, welche Lizenz haben Sie? ____________________

Wenn Sie Pilot sind, geben Sie bitte die Anzahl Ihrer Flugstunden an. ________ Stunden

Seit wann haben Sie einen Pilotenschein? ________

Wie oft machen Sie Computerspiele?

- nie ☐
- ca. fünfmal pro Monat ☐ Seit wie vielen Jahren? ________
- ca. dreimal pro Woche ☐ Seit wie vielen Jahren? ________
- nahezu täglich ☐ Seit wie vielen Jahren? ________

Welche Art von Comperspielen spielen Sie häufig?
(Mehrfachnennungen möglich)

Namen der häufigsten Spiele:

- Ego-Shooter ☐ ________
- Adventure ☐ ________
- Strategiespiele ☐ ________
- Rollenspiele ☐ ________
- Jump-'n'-run ☐ ________
- Flugsimulationen ☐ ________
- Rennspiele ☐ ________
- Sportspiele ☐ ________

Möchten Sie Anmerkungen zum Versuch machen?

____

____

____

____

____

____

____

Vielen Dank für Ihre Teilnahme!
Sie haben uns sehr geholfen.

Appendix E.

Questionnaires User Study 3

ACROSS

ID:

Age:

Gender:

Pilot Licence:

Flight Hours:

Type Ratings:

Opinion towards the use of Touchscreens in aircraft cockpits:

--	-	0	+	++

Additional remarks:

ACROSS

ID:

NASA-TLX:

Weight	ACROSS LEGACY
1 | 2 | 3 | 4 | 5	**Mental Demand** — How mentally demanding was the task? Very Low … Very High
1 | 2 | 3 | 4 | 5	**Physical Demand** — How physically demanding was the task? Very Low … Very High
1 | 2 | 3 | 4 | 5	**Temporal Demand** — How hurried or rushed was the pace of the task? Very Low … Very High
1 | 2 | 3 | 4 | 5	**Performance** — How successful were you in accomplishing what you were asked to do? Perfect … Failure
1 | 2 | 3 | 4 | 5	**Effort** — How hard did you have to work to accomplish your level of performance? Very Low … Very High
1 | 2 | 3 | 4 | 5	**Frustration** — How insecure, discouraged, irritated, stressed, and annoyed wereyou? Very Low … Very High

SART:

Instability of Situation

How changeable is the situation? Is the Situation highly unstable and likely to change suddenly (High) or is it very stable and straightforward (Low)?

Low High

Complexity of Situation

How complicated is the situation? Is it complex with many interrelated components (High) or is it simple and straightforward (Low)?

Low High

Variability of Situation

How many variables are changing within the situation? Are there a large numbers of factors varying (High) or are there very few variables changing (Low)?

Low High

Arousal

How aroused are you in the situation? Are you alert and ready for activity (High) or do you have a low degree of alertness (Low)?

Low High

Concentration of Attention

How much are you concentrating on the situation? Are you concentrating on many aspects of the situation (High) or focused on only one (Low)?

Low High

Division of Attention

How much is your attention divided in the situation? Are you concentrating on many aspects of the situation (High) or focused on only one (Low)?

Low High

Spare Mental Capacity

How much mental capacity do you have to spare in the situation? Do you have sufficient to attend to many variables (High) or nothing to spare at all (Low)?

Low High

Information Quantity

How much information have you gained about the situation? Have you received and understood a great deal of knowledge (High) or very little (Low)?

Low High

Information Quality

How good is the information you have gained about the situation? The communicated knowledge was useful (High) or not useful (Low)?

Low High

Familiarity with Situation

How familiar are you with the situation? Do you have a great deal of relevant experience (High) or is it a new situation (Low)?

Low High

Appendix F.

Questionnaires User Study 4

ACROSS

ID:

Participant Information Sheet

Title of Project: Advanced Cockpit for Reduction Of StreSs and workload (ACROSS)

Experiment Purpose & Procedure

The purpose of this experiment is to evaluate a virtual reality flight simulator, a head worn display device and a novel radio management panel.

The experiment consists of 4 parts during which you will be asked to perform an approach to Clermont Ferrand (LFLC), and an approach to Munich (EDDM). In addition you will be asked to assess a radio management panel and the symbology and behaviour of a head worn display device.

After each experiment, you will be asked to complete questionnaires.

Please note that none of the tasks is a test of your personal intelligence or ability. The objective is to test the usability of our research systems.

Your participation in this experiment is entirely voluntary and you are free to withdraw from the experiment at any time.

Confidentiality

The following data will be recorded: Telemetry data of the flight simulator, head and hand movements, Voice (e.g. ATC communication).

All data will be coded so that your anonymity will be protected in any research papers and presentations that result from this work.

The recording of video and audio material during the experiment is for the sole purpose of analysis by the research team. This material will be used by Airbus Group Innovations and ACROSS consortium partner researchers only. The images recorded will not be used in any publication or presentation without your consent. The research team will obtain a separate release statement if required.

(If data is to be recorded that would identify the participant, for example photographs, audio or video, and if there is any intention to use this material in any publication or presentation, a separate release statement should be obtained after the recording has been made).

Finding out about results

If interested, you can find out the result of the study by contacting the researcher Matthias Oberhauser , after date 31.12.2015. The phone number is +49 (0) 89 607 2 12 36 and the email address is Matthias.oberhauser@airbus.com

Version: Date:

ACROSS

ID:

Participant Consent Form

Title of Project: Advanced Cockpit for Reduction Of StreSs and workload (ACROSS)

Name of Principle Researcher: Matthias Oberhauser

Please initial box

1. I confirm that I have read the information sheet for the above study and have had the opportunity to ask questions and have had these answered satisfactorily. ☐

2. I understand that my participation is voluntary and I am free to withdraw consent at any time, without giving a reason, without my legal rights being affected. ☐

3. I understand that data collected during the study may be looked at by responsible & authorised personnel from the ACROSS consortium. I give permission for these individuals to have access to my anonymized data. ☐

4. I understand that data collected may be looked at by responsible representatives from the organiser (Airbus Group Innovations) and the ACROSS consortium for the purposes of monitoring and auditing to ensure that the study is being conducted properly. I give permission for these individuals to have access to relevant information. ☐

5. I agree to take part in the above study. ☐

Name of participant	Signature of participant	Date

Name of Principle Researcher	Signature of Principle Researcher	Date

When completed: 1 copy for participant; 1 copy (original) for researcher file

Version: Date:

ACROSS

ID:

With this questionnaire we want to capture your background and your experience as a pilot that is relevant for us within the context of this study.

1. Gender Male ☐ Female ☐

2. Age 20-30 ☐ 31-40 ☐ 41-50 ☐ 51-60 ☐ 61+ ☐

3. Which type ratings do you hold and which types of aircraft are you currently flying?

4. How many total flying hours do you approximately have?

5. Do you have experience in flying with Head Up Displays (HUD) in aircrafts?

☐ No ☐ Yes, in a simulator ☐ Yes, in a civil aircraft ☐ Yes, in a military aircraft

If Yes (simulator/civil/military): How many hours have you approximately flown with a HUD?

6. Do you have experience in flying with Head Mounted/Head Worn Displays in aircrafts?

☐ No ☐ Yes, in a simulator ☐ Yes, in a civil aircraft ☐ Yes, in a military aircraft

If Yes (simulator/civil/military): How many hours have you approximately flown with a HUD?

7. Do you wear/need corrective/ophthalmic glasses or contact lenses? ☐ Yes ☐ No

If Yes: What is your lens prescription?

ACROSS

ID:

Simulator Sickness Questionnaire:

		None	Slight	Moderate	Severe
1	General discomfort	0	1	2	3
2	Fatigue	0	1	2	3
3	Headache	0	1	2	3
4	Eye strain	0	1	2	3
5	Difficulty focusing	0	1	2	3
6	Salivation increasing	0	1	2	3
7	Sweating	0	1	2	3
8	Nausea	0	1	2	3
9	Difficulty concentrating	0	1	2	3
10	« Fullness of the Head »	0	1	2	3
11	Blurred vision	0	1	2	3
12	Dizziness with eyes open	0	1	2	3
13	Dizziness with eyes closed	0	1	2	3
14	*Vertigo	0	1	2	3
15	**Stomach awareness	0	1	2	3
16	Burping	0	1	2	3

* Vertigo is experienced as loss of orientation with respect to vertical upright.

** Stomach awareness is usually used to indicate a feeling of discomfort which is just short of nausea.

ACROSS

ID:

Head Worn Display (HWD) Questionnaire:

The questionnaire consists of three parts. In the first part of the questionnaire we would like to know your general experience with the Head Worn Display device with regards to technical parameters and usability. The second part of the questionnaire is focused on the Head Worn Display symbology where we want to know how you experienced the display symbology and the different symbology Reference modes during the part task evaluation. In the third part of the questionnaire we ask you about your experience in the context of the parallel runway approach finalized by two general questions regarding the HWD implementation.

Part 1 General Experience with the Head Worn Display device:

Technical characteristics of HWD device:

Please tick the appropriate box, choosing the statement which best represents your opinion.

	Strongly Agree	Somewhat Agree	Undecided	Somewhat Disagree	Strongly Disagree
The field of view of the display is sufficient.	☐	☐	☐	☐	☐
The brightness of the display is sufficient.	☐	☐	☐	☐	☐
The contrast of the display is sufficient.	☐	☐	☐	☐	☐
The weight of the device is acceptable.	☐	☐	☐	☐	☐
The information of the display was well readable.	☐	☐	☐	☐	☐
The delay of Head Tracking is acceptable.	☐	☐	☐	☐	☐

Usability:

What are possible advantages of this kind of system from your point of view?

What are possible drawbacks of this kind of system from your point of view?

Would you like to use this system? ☐Yes ☐No

If yes why? If not why?

Do you think such a system could replace a HUD? ☐Yes ☐No

If yes why? If not why?

ACROSS

ID:

Part 2 Head Worn Display symbology and reference modes during part task evaluation:

Symbology reference modes:

Please order the symbology reference modes into your preferable order: 1 (best) to 3 (worst)

___ Mode A (Aircraft reference mode)

___ Mode B (Flight path vector reference mode)

___ Mode C (Head reference mode)

	Mode	Strongly Agree	Somewhat Agree	Undecided	Somewhat Disagree	Strongly Disagree
The display was too cluttered.	A	☐	☐	☐	☐	☐
	B	☐	☐	☐	☐	☐
	C	☐	☐	☐	☐	☐
	Mode					
The amount and density of the information in the display was appropriate.	A	☐	☐	☐	☐	☐
	B	☐	☐	☐	☐	☐
	C	☐	☐	☐	☐	☐
	Mode					
The information displayed when moving the head to the sides was beneficial.	A	☐	☐	☐	☐	☐
	B	☐	☐	☐	☐	☐
	C	☐	☐	☐	☐	☐
It should be possible to switch between the different modes.		☐	☐	☐	☐	☐

Why do you prefer Mode (A, B, C)?

In which flight phases or flight situations do you think Mode (A, B, C) could be beneficial?

Additional remarks/comments:

Part 3 Head Worn Display symbology during parallel runway approach:

Please tick the appropriate box, choosing the statement which best represents your opinion.

	Strongly Agree	Somewhat Agree	Undecided	Somewhat Disagree	Strongly Disagree
Information in the display provided support during the approach.	☐	☐	☐	☐	☐
I was always aware of the position of the other aircraft.	☐	☐	☐	☐	☐
The square indicating the aircraft position was useful.	☐	☐	☐	☐	☐
It is important to have information about the altitude of the other aircraft.	☐	☐	☐	☐	☐
It is important to have information about the distance of the other aircraft.	☐	☐	☐	☐	☐
It is important to have information about the flight path of the other aircraft.	☐	☐	☐	☐	☐
As soon as the other aircraft entered the critical zone (no transgression zone) I was aware of it.	☐	☐	☐	☐	☐
The additional field of regard of the HWD was useful.	☐	☐	☐	☐	☐
The information was presented in an intuitive manner.	☐	☐	☐	☐	☐

What information about the other aircraft would you like to have in addition?

What else could be done to improve the Head Worn Display?

Bibliography

Adelstein, Bernard, Alan Hobbs, John O'Hara, and Cynthia Null (2006). *Design, Development, Testing, and Evaluation: Human Factors Engineering*. Ed. by Nasa Langley Research Center. Hampton, VA, USA (cit. on p. 11).

Advanced Realtime Tracking GmbH (2016). *System user manual: ARTtrack, TRACKPACK & DTrack*. Ed. by Advanced Realtime Tracking GmbH. Weilheim i. OB, Germany (cit. on pp. 29, 53, 59).

Airbus S.A.S. (2014). *Commercial Aviation Accidents 1958-2014: A Statistical Analysis*. Ed. by Airbus S.A.S. Blagnac, France (cit. on p. 5).

Alexander, Amy L., Tad Brunyé, Jason Sidman, and Shawn A. Weil (2005). "From gaming to training: A review of studies on fidelity, immersion, presence, and buy-in and their effects on transfer in pc-based simulations and games." In: *Proceedings from the 2005 interservice/industry training, simulation, and education conference (I/ITSEC)*, pp. 1–15 (cit. on pp. 24, 27).

ALICIA (2014). *All Condition Operations and Innovative Cockpit Infrastructure*. Ed. by Deep Blue consulting and research. URL: http://www.alicia-project.eu/ (visited on 02/14/2018) (cit. on pp. 42, 108).

Arbon, Edward, Allen Mears, Leonard Wojcik, and Tania Calhoun (1991). *Head-up Guidance System Technology - A Powerful Tool for Accident Prevention*. Ed. by Flight Safety Foundation. Alexandria, VA, USA (cit. on p. 109).

Arthur III, Jarvis J., Lawrence Prinzel III, Kevin Shelton, Lynda J. Kramer, Steven P. Williams, Randall E. Bailey, and Robert M. Norman (2007). "Design and testing of an unlimited field-of-regard synthetic vision head-worn display for commercial aircraft surface operations." In: *SPIE Proceedings Vol. 6559*. Ed. by Jacques G. Verly and Jeff J. Guell. Bellingham, WA, USA: SPIE. DOI: 10.1117/12.719695 (cit. on p. 136).

Aslandere, Turgay I. (2013). "Interaction Methods in a Generic Virtual Reality Flight Simulator." Master's thesis. Munich, Germany: Technical University of Munich (cit. on pp. 55, 58, 59).

Aslandere, Turgay I., Daniel Dreyer, and Frieder Pankratz (2015). "Virtual hand-button interaction in a generic virtual reality flight simulator." In: *2015 IEEE Aerospace Conference*. Big Sky, MT, USA, pp. 1–8. DOI: 10.1109/AERO.2015.7118876 (cit. on pp. 16, 79).

Aslandere, Turgay I., Daniel Dreyer, Frieder Pantkratz, and Rene Schubotz (2014). "A Generic Virtual Reality Flight Simulator." In: *Virtuelle und Erweiterte Realität, 11. Workshop der GI-Fachgruppe VR/AR*. Ed. by Gabriel Zachmann, René Weller, and André Hinkenjann. Aachen, Germany: Shaker Verlag, pp. 1–13. ISBN: 978-3-8440-3054-9 (cit. on p. 16).

Assenheim, Harry M. (1992). "Peripheral Vision Displays." In: *Electro-Optical Displays*. Ed. by Mohammad A. Karim. New York, NY, USA: Marcel Dekker, pp. 311–336. ISBN: 0-8247-8695-5 (cit. on p. 119).

Avila, Lisa and Mike Bailey (2014). "Virtual Reality for the Masses." In: *IEEE Computer Graphics and Applications* 34.5, pp. 103–104. DOI: 10.1109/MCG.2014.103 (cit. on p. 13).

Babka, David W. (2011). "Flight Testing in a Simulation Based Environment." Senior project. San Luis Obispo, CA, USA: California Polytechnic University (cit. on p. 25).

Bailey, Randall E., Jarvis J. Arthur III, Steven P. Williams, and Lynda J. Kramer (2005). "Latency in visionic systems: Test methods and requirements." In: *Workshop on Toward Recommended Methods for Testing and Evaluation of EV and ESV Based Visionic Devices* (cit. on p. 85).

Bandow, Daniel (2006). *Head-Up-Guidance-Systeme und Mensch-Maschine-Interaktion*. Bericht aus dem Institut für Arbeitswissenschaft der TU Darmstadt. Stuttgart, Germany: Ergonomia-Verlag. ISBN: 978-3-935089-86-9 (cit. on pp. 109, 118, 119).

Barbé, Jérôme, Marion Wolff, and Régis Mollard (2013). "Human Centered Design Approach to Integrate Touch Screen in Future Aircraft Cockpits." In: *Human-Computer Interaction. Interaction Modalities and Techniques*. Ed. by Masaaki Kurosu. Berlin, Heidelberg, Germany: Springer, pp. 429–438. ISBN: 978-3-642-39330-3. DOI: 10.1007/978-3-642-39330-3_45 (cit. on p. 66).

Barbour, Sandy and Ben Supnik (2005). *XPLM Display APIs: Theory of Operation*. n.p. URL: http://www.xsquawkbox.net/xpsdk/docs/XPLMDisplay.html (visited on 08/26/2016) (cit. on p. 50).

Barco (2016). *F35 series: High performance DLP projector for professional applications: Product Specifications*. Ed. by Barco SAS. n.p. (cit. on p. 30).

Bauer, Marcus (2009). "Flugpfadrekonstruktion auf Grundlage von Zeugenaussagen mit Hilfe der Virtual-Reality Technologie." Dissertation. Darmstadt, Germany: Technical University of Darmstadt. URL: http://tuprints.ulb.tu-darmstadt.de/1894/ (visited on 10/13/2016) (cit. on p. 16).

Bauer, Marcus and Uwe Klingauf (2008). "Virtual-Reality as a Future Training Medium for Civilian Flight Procedure Training." In: *AIAA Modeling and Simulation Technologies Conference and Exhibit*. Ed. by American Institute of Aeronautics and Astronautics. Guidance, Navigation, and Control and Co-located Conferences. Reston, VA, USA: American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2008-7030 (cit. on pp. 16, 79, 157).

Beard, Steven D., Scott E. Reardon, Eric L. Tobias, and Bimal L. Aponso (2013). "Simulation System Fidelity Assessment at the Vertical Motion Simulator." In: *American Helicopter Society (AHS) Annual Forum and Technology Display*. Ed. by American Helicopter Society. Fairfax, VA, USA, pp. 21–23 (cit. on p. 10).

Berndt, Jon and Agostino De Marco (2009). "Progress On and Usage of the Open Source Flight Dynamics Model Software Library, JSBSim." In: *AIAA Modeling and Simulation Technologies Conference*. Ed. by American Institute of Aeronautics and Astronautics. Guidance, Navigation, and Control and Co-located Conferences. Reston, VA, USA: American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2009-5699 (cit. on p. 25).

Bittner, Kurt and Ian Spence (2003). *Use case modeling*. Boston, MA, USA: Addison Wesley. ISBN: 9780201709131 (cit. on pp. 19, 21).

Bokranz, Rainer and Kurt Landau (1991). *Einführung in die Arbeitswissenschaft: Analyse und Gestaltung von Arbeitssystemen*. Uni-Taschenbücher. Stuttgart, Germany: E. Ulmer. ISBN: 9783800126279 (cit. on p. 6).

Booch, Grady, Ivar Jacobson, and James Rumbaugh (2004). *The Unified Modeling Language reference manual*. 2. ed. The Addison-Wesley object technology series. Boston, MA, USA: Addison-Wesley. ISBN: 9780321245625 (cit. on p. 19).
Bowman, Douglas A. and Larry F. Hodges (1999). "Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments." In: *Journal of Visual Languages & Computing* 10.1, pp. 37–53. ISSN: 1045-926X. DOI: 10.1006/jvlc.1998.0111 (cit. on p. 17).
Bowman, Douglas A., Ernst Kruijff, Joseph J. Laviola Jr., and Ivan Poupyrev (2005). *3D user interfaces: Theory and practice*. Boston, MA, USA: Addison-Wesley. ISBN: 0201758679 (cit. on p. 57).
Bowman, Douglas A. and Ryan P. McMahan (2007). "Virtual reality: how much immersion is enough?" In: *Computer* 40.7, pp. 36–43. ISSN: 0018-9162 (cit. on p. 15).
Brandt, Thomas, Johannes Dichgans, and E. Koenig (1973). "Differential effects of central versus peripheral vision on egocentric and exocentric motion perception." In: *Experimental Brain Research* 16.5, pp. 476–491. ISSN: 0014-4819 (cit. on p. 32).
Broderick, Damien (1982). *The Judas mandala*. New York, NY, USA: Pocket Books. ISBN: 9780671450328 (cit. on p. 12).
Brooks, Frederick P. (1999). "What's Real About Virtual Reality?" In: *IEEE Computer Graphics and Applications* 19.6, pp. 16–27. ISSN: 02721716. DOI: 10.1109/38.799723 (cit. on pp. 13, 16).
Bürki-Cohen, Judith, Nancy N. Soja, and Thomas Longridge (1998). "Simulator platform motion-the need revisited." In: *The International Journal of Aviation Psychology* 8.3, pp. 293–317. ISSN: 1050-8414 (cit. on p. 154).
Cakmakci, Ozan and Jannick P. Rolland (2006). "Head-worn displays: a review." In: *Journal of Display Technology* 2.3, pp. 199–216. ISSN: 1551-319X. DOI: 10.1109/JDT.2006.879846 (cit. on p. 14).
Carmack, John (2013). *Latency Mitigation Strategies*. URL: https://web.archive.org/web/20140719053303/http://www.altdev.co/2013/02/22/latency-mitigation-strategies/ (visited on 02/14/2018) (cit. on p. 105).
Clamann, Michael and David B. Kaber (2004). "Applicability of usability evaluation techniques to aviation systems." In: *The International Journal of Aviation Psychology* 14.4, pp. 395–420. ISSN: 1050-8414 (cit. on p. 1).
Cousins, Steve, Brian Gerkey, Ken Conley, and Willow Garage (2010). "Sharing Software with ROS: ROS Topics." In: *IEEE Robotics & Automation Magazine* 17.2, pp. 19–20. DOI: 10.1109/MRA.2010.936956 (cit. on p. 38).
Cozzi, Patrick and Kevin Ring (2011). *3D engine design for virtual globes*. An A K Peters book. Boca Raton, FL, USA: CRC Press. ISBN: 9781322614274 (cit. on p. 25).
Craighead, Jeff, Robin Murphy, and Jenny Burke (2007). "A Survey of Commercial & Open Source Unmanned Vehicle Simulators." In: *IEEE International Conference on Robotics and Automation*. Ed. by Paolo Fiorini. New York, NY, USA, pp. 852–857. ISBN: 1-4244-0601-3 (cit. on p. 23).
Crawford, Jennifer and Andrew Neal (2006). "A Review of the Perceptual and Cognitive Issues Associated With the Use of Head-Up Displays in Commercial Aviation." In: *The International Journal of Aviation Psychology* 16.1, pp. 1–19. ISSN: 1050-8414. DOI: 10.1207/s15327108ijap1601_1 (cit. on p. 118).
Creighton, Ryan H. (2010). *Unity 3D game development by example: Beginner's guide*. Birmingham, UK: Packt Publishing. ISBN: 1849690545 (cit. on p. 59).

Creswell, John W. (2013). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. 4. ed. Thousand Oaks, CA, USA: SAGE Publications, Inc. ISBN: 9781483321479 (cit. on p. 7).

Crick, Christopher, Graylin Jay, Sarah Osentoski, Benjamin Pitzer, and Odest Chadwicke Jenkins (2011). "Rosbridge: Ros for non-ros users." In: *Robotics Research*. Ed. by Christensen Henrik and Khatib Oussama. Springer Tracts in Advanced Robotics. Cham, Switzerland: Springer International Publishing. ISBN: 978-3-319-29362-2 (cit. on pp. 37, 60).

Crockford, Douglas (2006). *The application/json media type for javascript object notation (json): RFC 4627*. Ed. by Network Working Group. URL: https://tools.ietf.org/html/rfc4627 (visited on 02/14/2018) (cit. on p. 51).

Cruz-Neira, Carolina, Sandin J. Daniel, and DeFanti A. Thomas (1993). "Surround-screen projection-based virtual reality: the design and implementation of the CAVE." In: *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. Anaheim, CA, USA: ACM, pp. 135–142. ISBN: 0-89791-601-8. DOI: 10.1145/166117.166134 (cit. on p. 15).

Curtis, Michael T., Florian Jentsch, and John A. Wise (2010). "Aviation Displays." In: *Human factors in aviation*. Ed. by Eduardo Salas and Daniel E. Maurino. Amsterdam and Boston: Academic Press/Elsevier, pp. 349–478. ISBN: 978-0123745187 (cit. on p. 5).

Da Silva, Christian (2010). "Eurocopter Virtual Environment Cybercopter." In: *Rotor Journal* 2010.85, p. 14. URL: http://www.airbushelicopters.com/w1/jrotor/85/files/docs/all.pdf (visited on 02/14/2018) (cit. on p. 16).

Data Reality (2016). *Virtual reality headsets comparison*. Ed. by Fabian Rohr. URL: http://data-reality.com/virtual-reality-headsets-comparison/ (visited on 03/14/2016) (cit. on p. 31).

Deaton, John E. and Jeffrey G. Morrison (2009). "A Framework for the Effective Practice of Human Factors, or "What Your Mentor Never Told You about a Career in Human Factors. . ."." In: *Handbook of Aviation Human Factors*. Ed. by John A. Wise, David V. Hopkin, and Daniel J. Garland. London, UK: CRC Press, pp. 15–31. ISBN: 9781420063578 (cit. on p. 107).

Degani, Asaf and Earl L. Wiener (1990). *Human Factors of Flight-Deck Checklists: The Normal Checklist*. Ed. by National Aeronautics and Space Administration. Moffett Field, CA, USA (cit. on pp. 6, 124).

Diebel, James (2006). "Representing attitude: Euler angles, unit quaternions, and rotation vectors." In: *Matrix* 58.15-16, pp. 1–35 (cit. on p. 54).

Domino, George and Marla L. Domino (2006). *Psychological testing: An introduction*. 2nd ed. Cambridge and New York: Cambridge University Press. ISBN: 978-0521861816 (cit. on p. 51).

Doolen, Toni, Roberto Nicolade, and Ken Funk (2000). *Improper Checklist Use as a Factor in Aircraft Accidents and Incidents*. URL: http://flightdeck.ie.orst.edu/ElectronicChecklist/HTML/accidents.html#study1 (visited on 02/27/2014) (cit. on p. 124).

Dörr, Kai-Uwe (2004). *Aufbau und Evaluation eines immersiven computerbasierten Trainingssystems in der Pilotenausbildung*. Bericht aus dem Fachgebiet Flugsysteme und Regelungstechnik der TU Darmstadt. Stuttgart, Germany: Ergonomia-Verlag. ISBN: 9783935089760 (cit. on p. 79).

Dörr, Kai-Uwe, Jens Schiefele, and W. Kubbat (2001). "Virtual Cockpit Simulation for Pilot Training." In: *What is essential for virtual reality systems to meet military human performance goals?* Ed. by North Atlantic Treaty Organization. RTO meeting proceedings. Neuilly-sur-Seine, France: RTO, Research and Technology Organization, pp. 11-1–11-7. ISBN: 9789283710578 (cit. on p. 16).
Dreyer, Daniel (2011). "Visuelle Anmutung von Flugzeugkabinen und Sichtbarkeitsuntersuchungen im Flugzeugcockpit (EADS)." In: *Virtuelle Techniken im industriellen Umfeld: Das AVILUS-Projekt - Technologien und Anwendungen*. Ed. by W. Schreiber and P. Zimmermann. Springer Berlin Heidelberg, pp. 152–159. ISBN: 9783642206368 (cit. on p. 16).
Dreyer, Daniel and Axel Hillebrand (2010). "Steigerung der Durchführungsobjektivität bei Probandenbefragungen in Virtual Reality." In: *Innovative Interaktionstechnologien für Mensch-Maschine-Schnittstellen*. Ed. by Morten Grandt and Anna Bauch. Bonn, Germany, pp. 205–214 (cit. on p. 75).
Dreyer, Daniel and Matthias Oberhauser (2016a). "Beyond the Push-Button: A study on system management operations in the flight deck." In: *Proceedings of the International Conference on Human Computer Interaction in Aerospace '16*. Ed. by Guy A. Boy. New York, NY, USA.: ACM (cit. on pp. 107, 124–126, 134, 149).
Dreyer, Daniel and Matthias Oberhauser (2016b). "Method and device for testing a device operated in an aircraft." European Patent EP3067874A1 (cit. on p. 154).
Dreyer, Daniel and Matthias Oberhauser (2016c). "Selection unit to select or control different states or functions of an aircraft system." European Patent EP3018571A1 (cit. on p. 125).
Dreyer, Daniel, Matthias Oberhauser, and Daniel Bandow (2014). "HUD Symbology evaluation in a Virtual Reality Flight Simulation." In: *Proceedings of the International Conference on Human-Computer Interaction in Aerospace '14*. New York, NY, USA: ACM, pp. 9–14. ISBN: 978-1-4503-2560-8. DOI: 10.1145/2669592.2669652 (cit. on pp. xvii, 52, 107–109, 124).
Dreyer, Daniel, Matthias Oberhauser, Andreas Prücklmeier, and Sebastien Mamessier (2017). "Device and Method for Testing Function or Use of a Head Worn See Through Augmented Reality Device." European Patent 15198726.0 - 1562 (cit. on p. 142).
Dreyer, Daniel, Matthias Oberhauser, and Philipp Schmidt (2015). "Fire extinguishing system for an aircraft." European Patent EP20150161127 (cit. on p. 125).
Elder, Stuart and Andrew Vakaloudis (2015). "Towards uniformity for smart glasses devices: An assessment of function as the driver for standardisation." In: *2015 IEEE International Symposium on Technology and Society*. New York, NY: IEEE, pp. 1–7. DOI: 10.1109/ISTAS.2015.7439424 (cit. on p. 137).
Endsley, Mica R. (1988). "Design and Evaluation for Situation Awareness Enhancement." In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 32.2, pp. 97–101. ISSN: 1541-9312. DOI: 10.1177/154193128803200221 (cit. on p. 6).
Endsley, Mica R. (1995). "Toward a Theory of Situation Awareness in Dynamic Systems." In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 37.1, pp. 32–64. ISSN: 0018-7208. DOI: 10.1518/001872095779049543 (cit. on p. 6).
Endsley, Mica R., Stephen J. Selcon, Thomas D. Hardiman, and Darryl G. Croft (1998). "A Comparative Analysis of Sagat and Sart for Evaluations of Situation Aware-

ness." In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 42.1, pp. 82–86. ISSN: 1541-9312. DOI: 10.1177/154193129804200119 (cit. on p. 7).

Ergoneers GmbH (2011a). *D-Lab & D-Lab Control: Plan. Measure. Analyse: User Manual*. Geretsried, Germany (cit. on pp. 72, 115).

Ergoneers GmbH (2011b). *Dikablis - The Eye Tracking System: User Manual Dikablis Software: For Dikablis Software Version 2.0*. Geretsried, Germany (cit. on pp. 72, 73).

Ernst, Johannes M., Hans-Ullrich Doehler, and Sven Schmerwitz (2016). "A concept for a virtual flight deck shown on an HMD." In: *Proceedings SPIE 9839*. Ed. by Jack Sanders-Reed and Jarvis J. Arthur. Bellingham, WA, USA. DOI: 10.1117/12.2224933 (cit. on p. 136).

European Aviation Safety Agency (2012). *Certification Specifications for Aeroplane Flight Simulation Training Devices* (cit. on pp. 154, 156).

Fadden, Steven, Patricia M. Ververs, and Christopher D. Wickens (2001). "Pathway HUDs: Are They Viable?" In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 43.2, pp. 173–193. ISSN: 0018-7208. DOI: 10.1518/001872001775900841 (cit. on p. 119).

Fette, Ian and Alexey Melnikov (2011). *The websocket protocol: RFC 6455*. Ed. by Internet Engineering Task Force. DOI: 10.17487/RFC6455 (cit. on p. 37).

Fischer, Edith, Richard F. Haines, and Tonu A. Price (1980). *Cognitive issues in head-up displays: NASA Technical Paper 1711*. Ed. by NASA Scientific and Technical Information Branch. Hampton, VA, USA (cit. on p. 119).

Foote, Tully (2013). "tf: The transform library." In: *IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*. New York, NY, USA: IEEE. DOI: 10.1109/TePRA.2013.6556373 (cit. on p. 54).

Gopher, Daniel and Emanuel Donchin (1986). "Workload: An examination of the concept." In: *Handbook of perception and human performance, Vol. 2: Cognitive processes and performance*. Ed. by Kenneth R. Boff, Lloyd Kaufman, and James P. Thomas. Oxford, England: John Wiley & Sons, pp. 1–49. ISBN: 0-471-82956-0 (cit. on p. 6).

Goutal, Laurent (2000). "Ergonomics assessment for aircraft cockpit using the virtual mock-up." In: *Ergonomic software tools in product and workplace design*. Ed. by Kurt Landau. Stuttgart: Verlag Ergon, pp. 173–183. ISBN: 3932160118 (cit. on p. 16).

Häkinnen, Jukka, Monika Pölönen, Jari Takatalo, and Göte Nyman (2006). "Simulator sickness in virtual display gaming: a comparison of stereoscopic and non-stereoscopic situations." In: *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*. New York, NY, USA: ACM, pp. 227–230. ISBN: 1-59593-390-5. DOI: 10.1145/1152215.1152263 (cit. on p. 56).

Hamburger, Ellis (2014). *An Oculus Rift in Every Home: Cory Ondrejka on Facebook's $2 billion bet*. Ed. by The Verge. URL: http://www.theverge.com/a/virtual-reality/qa_fb (visited on 02/14/2018) (cit. on p. 154).

Harris, Don, ed. (2004). *Human Factors for Civil Flight Deck Design*. Farnham, UK: Ashgate. ISBN: 9780754613800. URL: http://books.google.de/books?id=LvhHXKQtPjMC (cit. on p. 5).

Hart, Sandra G. (2006). "NASA-task load index (NASA-TLX); 20 years later." In: *Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting*. Vol. 9. Thousand Oaks, CA, USA: Sage Publications (cit. on pp. 6, 9).

Hart, Sandra G. and Lowell E. Staveland (1988). "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research." In: *Advances in psychology* 52, pp. 139–183. ISSN: 0166-4115 (cit. on pp. 6, 8).

Heilig, Morton L. (1962). "Sensorama Simulator." US Patent No. 3,050,870, Washington, DC: U.S. Patent and Trademark Office. URL: http://www.google.com/patents/US3050870 (cit. on pp. 13, 156).

Helmreich, Robert L. and H. Clayton Foushee (1993). "Why crew resource management? Empirical and theoretical bases of human factors training in aviation." In: *Cockpit resource management*. Ed. by Earl L. Wiener, Barbara G. Kanki, and Robert L. Helmreich. San Diego, CA, US: Academic Press, pp. 3–45. ISBN: 0-12-750025-1 (cit. on p. 64).

Hicks, Mark and Gabrielle de Brito (1999). "Civil aircraft warning systems: who's calling the shots?" In: *Air & Space Europe* 1.1, pp. 45–47. ISSN: 12900958. DOI: 10.1016/S1290-0958(99)80037-9 (cit. on p. 5).

Hillebrand, Axel (2013). "ATTENDO: Method for the Assessment of Visual Attention Allocation in Two-dimensional Spaces." In: *Proceedings of the Human Factors and Ergonomics Society 2013 Annual Meeting*. Thousand Oaks, CA, USA: SAGE Publications, Inc, pp. 2047–2051. ISBN: 978-0-945289-43-2. DOI: 10.1177/1541931213571457 (cit. on p. 8).

Hillebrand, Axel, Matthias Oberhauser, Daniel Dreyer, and Daniel Meister (2014). "Modulare Erweiterbarkeit von Flugsimulationen für Human-Factors-Experimente." In: *3. Interdisziplinärer Workshop Kognitive Systeme: Mensch, Teams, Systeme und Automaten* (cit. on pp. 8, 74).

Hillebrand, Axel, Eva Wahrenberg, and Dietrich Manzey (2012). "A new method to assess pilots' allocation of visual attention using a head-up display." In: *Proceedings HFES Europe Chapter Conference*. Ed. by Dick de Waard. Groningen, Netherlands, pp. 179–192. ISBN: 978-0-945289-44-9 (cit. on p. 8).

Holzinger, Andreas (2005). "Usability Engineering Methods For Software Developers." In: *Communications of the ACM* 48.1, pp. 71–74. DOI: 10.1145/1039539.1039541 (cit. on p. 127).

Hosman, Ruud and Henk Stassen (1999). "Pilot's perception in the control of aircraft motions." In: *Control Engineering Practice* 7.11, pp. 1421–1428. ISSN: 0967-0661. DOI: 10.1016/S0967-0661(99)00111-2 (cit. on p. 104).

Hüsgen, Silke and Uwe Klingauf (2005). "Interaktionsmechanismen zur Verbesserung der Mensch-Maschine-Schnittstelle am Beispiel eines virtuellen Flugsimulators." In: *Deutscher Luft- und Raumfahrtkongress* (cit. on p. 16).

IEEE Computer Society (2005). *IEEE standard for software verification and validation*. New York, NY, USA (cit. on p. 76).

Jacob, Robert J. K. and Keith S. Karn (2003). "Eye tracking in human-computer interaction and usability research: Ready to deliver the promises." In: *The Mind's Eye*. Ed. by Ralph Radach, Jukka Hyona, and Heiner Deubel. Vol. 2. Amsterdam, Netherlands: North Holland Press and Elsevier, pp. 573–605. ISBN: 978-0-444-51020-4 (cit. on p. 7).

Jarvis, Peter, Bernard Lalonde, and Daniel Spira (2008). "Flight Simulator Modeling and Validation Approaches and Pilot-in-the-Loop Fidelity." In: *AIAA Modeling and Simulation Technologies Conference and Exhibit*. Ed. by American Institute of Aeronautics and Astronautics. Guidance, Navigation, and Control and Co-located

Conferences. Reston, VA, USA: American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2008-6684 (cit. on pp. 26, 156).

Jorna, Peter G.A.M. and Piet J. Hoogeboom (2004). "Evaluating the flight deck." In: *Human Factors for Civil Flight Deck Design*. Ed. by Don Harris. Farnham, UK: Ashgate, pp. 235–275. ISBN: 9780754613800 (cit. on pp. 7, 9, 150).

Joyce, Richard D. and Stephen K. Robinson (2015). "The Rapidly Reconfigurable Research Cockpit." In: *AIAA Modeling and Simulation Technologies Conference*. AIAA Aviation. Reston, VA, USA: American Institute of Aeronautics and Astronautics. ISBN: 978-1-62410-355-1. DOI: 10.2514/6.2015-2336 (cit. on p. 16).

Just, Marcel A. and Patricia A. Carpenter (1980). "A theory of reading: From eye fixations to comprehension." In: *Psychological Review* 87.4, pp. 329–354. ISSN: 1939-1471(Electronic);0033-295X(Print) (cit. on p. 7).

Käfer, Tobias, Andreas Harth, and Sebastien Mamessier (2016). "Towards declarative programming and querying in a distributed Cyber-Physical System: The i-VISION case." In: *2016 2nd International Workshop on Modelling, Analysis, and Control of Complex CPS (CPS Data)*, pp. 1–6. ISBN: 978-1-5090-1154-4. DOI: 10.1109/CPSData.2016.7496418 (cit. on p. 151).

Kaiser, Mary K. and Jeffery A. Schroeder (2002). "Flights of Fancy: The Art and Science of Flight Simulation." In: *Principles and Practices of Aviation Psychology*. Ed. by Pamela S. Tsang and Michael A. Vidulich. Mahwah, NJ, USA: Lawrence Erlbaum, pp. 435–471. ISBN: 0805833900 (cit. on p. 2).

Kasarskis, Peter, Jennifer Stehwien, Joey Hickox, Anthony Aretz, and Chris Wickens (2001). "Comparison of expert and novice scan behaviors during VFR flight." In: *Proceedings of the 11th International Symposium on Aviation Psychology* (cit. on p. 123).

Kathy H. Abbott (2014). "Human Factors Engineering and Flight Deck Design." In: *Digital avionics handbook*. Ed. by Spitzer Carry, Ferrell Uma, and Ferrell Thomas. CRC Press, pp. 15-1–15-15. ISBN: 9781439868980 (cit. on pp. 11, 12).

Kennedy, Robert S., Norman E. Lane, Kevin S. Berbaum, and Michael G. Lilienthal (1993). "Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness." In: *The International Journal of Aviation Psychology* 3.3, pp. 203–220. ISSN: 1050-8414. DOI: 10.1207/s15327108ijap0303_3 (cit. on pp. 14, 90).

Kolasinski, Eugenia M. (1995). *Simulator Sickness in Virtual Environments: Technical Report 1027*. Ed. by United States Army Research Institute for the Behavioral and Social Sciences. Alexandria, VA, USA (cit. on p. 13).

Koonce, Jefferson and Anthony Debons (2009). "A Historical Overview of Human Factors in Aviation." In: *Handbook of Aviation Human Factors*. Ed. by John A. Wise, David V. Hopkin, and Daniel J. Garland. London, UK: CRC Press, pp. 1-1–1-11. ISBN: 9781420063578. DOI: 10.1201/b10401-3 (cit. on pp. 5, 9).

Krueger, Myron W. (1991). *Artificial reality II*. 2nd edition. Reading, UK: Addison-Wesley. ISBN: 9780201522600 (cit. on p. 12).

Kushner, David (2011). *The Making of Arduino: How five friends engineered a small circuit board that's taking the DIY world by storm: IEEE Spectrum*. URL: http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino (visited on 02/14/2018) (cit. on p. 91).

Lane, Nicholas D., Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T. Campbell (2010). "A survey of mobile phone sensing." In: *IEEE Communications Magazine* 48.9, pp. 140–150. ISSN: 0163-6804. DOI: 10.1109/MCOM.2010.5560598 (cit. on p. 15).

Lange, Christian (2005). “The development and usage of Dikablis (Digital wireless gaze tracking system).” In: *Abstracts of the 13th European Conference on Eye Movements 2005*. Ed. by Marina Groner, Rudolf Groner, R. Müri, Kazuo Koga, Simon Raess, and P. Suri. Bern, Switzerland, p. 50 (cit. on p. 72).

Larimer, James O., Michael P. Prevost, Aries R. Arditi, Steven Azueta, James R. Bergen, and Jeffrey Lubin (1991). “Human visual performance model for crewstation design.” In: *Large Screen Projection, Avionic, and Helmet-Mounted Displays*. Ed. by Harry M. Assenheim, Richard A. Flasck, Thomas M. Lippert, and Jerry Bentz. Vol. 1456, pp. 196–210. DOI: 10.1117/12.45429 (cit. on p. 16).

Le Ngoc, Luan and Roy S. Kalawsky (2013). “Visual Circuit Flying with Augmented Head-tracking on Limited Field of View Flight Training Devices.” In: *Proceedings of the AIAA Modeling and Simulation Technologies Conference 2013*. Reston, VA, USA: American Institute of Aeronautics and Astronautics, pp. 1–16. DOI: 10.2514/6.2013-5226 (cit. on p. 86).

Lee, Jonathan and Nien-Lin Xue (1999). “Analyzing User Requirements by Use Cases: A Goal-Driven Approach.” In: *IEEE Software* 16.4, pp. 92–101 (cit. on p. 19).

Li, Guangyan and Peter Buckle (1999). “Current techniques for assessing physical exposure to work-related musculoskeletal risks, with emphasis on posture-based methods.” In: *Ergonomics* 42.5, pp. 674–695. ISSN: 0014-0139. DOI: 10.1080/001401399185388 (cit. on pp. 9, 76).

Liesecke, Stephan (2013). “Eye-Tracking in Virtual Reality.” Master’s thesis. Munich, Germany: Bundeswehr University Munich (cit. on pp. 31, 72, 73).

Liesecke, Stephan, Daniel Dreyer, and Franz Kocvara (2015). “Hmd apparatus with adjustable eye tracking device.” US Patent US20150009574A1 (cit. on p. 72).

Link, Edwin Albert (1931). “Combination training device for student aviators and entertainment apparatus.” US Patent US1825462A (cit. on p. 9).

Lisker, Bernardo (2014). *Airport Capacity - Into the Future*. Global Aviation Cooperation Symposium (GACS), Montreal, Canada (cit. on p. 139).

Liu, Dahai, Nikolas D Macchiarella, and Dennis A Vincenzi (2009). “Simulation Fidelity.” In: *Human factors in simulation and training*. Ed. by Dennis A. Vincenzi, John A. Wise, Mustapha Mouloua, and Peter A. Hancock. Boca Raton, FL, USA: CRC Press, pp. 61–73. ISBN: 1420072846 (cit. on pp. 10, 79).

Long, Jeffry and Christopher D. Wickens (1994). “Conformal versus nonconformal symbology and the head-up display.” In: *Proc. SPIE 2218*. Ed. by Ronald J. Lewandowski, Stephens Wendell, and Loran A. Haworth. DOI: 10.1117/12.177380. URL: http://dx.doi.org/10.1117/12.177380 (cit. on p. 137).

Luczak, Holger (1997). *Arbeitswissenschaft: Analyse und Gestaltung der Arbeit*. 2nd edition. Springer-Lehrbuch. Berlin: Springer. ISBN: 978-3540591382 (cit. on p. 118).

Lyons, Franklin A. (2016). “Mobile head mounted display.” US Patent D751072 (cit. on p. 157).

MacKenzie, Scott I. and Colin Ware (1993). “Lag as a determinant of human performance in interactive systems.” In: *Proceedings of the INTERACT ’93 and CHI ’93 Conference on Human Factors in Computing Systems*. Ed. by Bert Arnold, Gerrit van der Veer, and Ted White. New York, NY, USA: ACM, pp. 488–493. ISBN: 0-89791-575-5. DOI: 10.1145/169059.169431 (cit. on p. 94).

Malcolm, Richard (1984). “Pilot disorientation and the use of a peripheral vision display.” In: *Aviation, Space, and Environmental Medicine* 55.3, pp. 231–238. ISSN: 0095-6562(Print) (cit. on p. 119).

Mamessier, Sebastien, Daniel Dreyer, and Matthias Oberhauser (2014). "Calibration of Online Situation Awareness Assessment Systems Using Virtual Reality." In: *Digital Human Modeling. Applications in Health, Safety, Ergonomics and Risk Management: 5th International Conference, DHM 2014, Held as Part of HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014. Proceedings*. Ed. by Vincent G. Duffy. Cham: Springer International Publishing, pp. 124–135. ISBN: 978-3-319-07725-3. DOI: 10.1007/978-3-319-07725-3_12 (cit. on p. 151).

Mamessier, Sebastien, Daniel Dreyer, and Matthias Oberhauser (2015). "Semantically Integrated Human Factors Engineering." In: *Digital Human Modeling. Proceedings, Part II*. Ed. by G. Vincent Duffy. Cham, Switzerland: Springer International Publishing, pp. 284–294. ISBN: 978-3-319-21070-4. DOI: 10.1007/978-3-319-21070-4_29 (cit. on pp. xvii, 39, 47, 157).

Mamessier, Sebastien and Karen M. Feigh (2016). "HTA-Based Tracking of Pilot Actions in the Cockpit." In: *Advances in Physical Ergonomics and Human Factors: Proceedings of the AHFE 2016 International Conference on Physical Ergonomics and Human Factors, July 27-31, 2016, Walt Disney World®, Florida, USA*. Ed. by Ravindra Goonetilleke and Waldemar Karwowski. Cham: Springer International Publishing, pp. 93–103. ISBN: 978-3-319-41694-6. DOI: 10.1007/978-3-319-41694-6_10 (cit. on p. 151).

McCarty, Dean W., Steven Sheasby, Philip Amburn, Martin R. Stytz, and Chip Switzer (1994). "A virtual cockpit for a distributed interactive simulation." In: *IEEE Computer Graphics and Applications* 14.1, pp. 49–54. ISSN: 0272-1716. DOI: 10.1109/38.250919 (cit. on p. 16).

McLellan, Hilary (1996). "Virtual Realities." In: *Handbook of research for educational communications*. Ed. by David H. Jonassen. Boston, MA, USA: Kluwer-Nijhoff Publishing, pp. 457–487 (cit. on p. 13).

Moir, Ian, Allan G. Seabridge, and Malcolm Jukes (2006). *Military avionics systems*. Chichester, UK: John Wiley & Sons. ISBN: 978-0-470-01632-9 (cit. on pp. 108, 136).

Moroney, William F. and Michael G. Lilienthal (2009). "Human Factors in Aviation and Training: An Overview." In: *Human factors in simulation and training*. Ed. by Dennis A. Vincenzi, John A. Wise, Mustapha Mouloua, and Peter A. Hancock. Boca Raton, FL, USA: CRC Press, pp. 3–26. ISBN: 1420072846 (cit. on p. 10).

Moroney, William F. and Brian W. Moroney (2009). "Flight simulation." In: *Handbook of Aviation Human Factors*. Ed. by John A. Wise, David V. Hopkin, and Daniel J. Garland. London, UK: CRC Press, pp. 355–388. ISBN: 9781420063578 (cit. on pp. 1, 151).

Mühlbauer, Frank (2010). "Konzeption und Entwurf einer Cockpitkernarchitektur mit neuen Technologien." Master's Thesis. Dresden: TU Dresden (cit. on pp. 42, 43).

Mulder, Lambertus J. M., Dick De Ward, and Karela Brookhuis (2004). "Estimating Mental Effort Using Heart Rate and Heart Rate Variability." In: *Handbook of Human Factors and Ergonomics Methods*. Ed. by Neville A. Stanton, Alan Hedge, Karel Brookhuis, Eduardo Salas, and Hal W. Hendrick. London, UK: Taylor & Francis, pp. 20-1–20-8. ISBN: 9780415287005. DOI: 10.1201/9780203489925.ch20 (cit. on p. 9).

Murphy, Joel and Yury Gitman (2016). *The Pulse Sensor Amped*. URL: http://pulsesensor.com (visited on 02/14/2018) (cit. on p. 74).

National Transportation Safety Board, ed. (2010). *Loss of Thrust in Both Engines After Encountering a Flock of Birds and Subsequent Ditching on the Hudson River: US Airways*

Flight 1549: Airbus A320–214, N106US, Weehawken, New Jersey, January 15, 2009. Washington, D.C., USA (cit. on p. 125).

Noyes, Jan M., Andrew F. Starr, and Clive R. Frankish (1996). “User involvement in the early stages of the development of an aircraft warning system.” In: *Behaviour & information technology* 15.2, pp. 67–75. ISSN: 0144-929X. DOI: 10.1080/014492996120274 (cit. on p. 11).

Oberhauser, Matthias and Daniel Dreyer (2017). “A virtual reality flight simulator for human factors engineering.” In: *Cognition, Technology & Work* 19.2, pp. 263–277. ISSN: 1435-5566. DOI: 10.1007/s10111-017-0421-7 (cit. on pp. xvii, 107).

Oberhauser, Matthias, Daniel Dreyer, Reinhard Braunstingl, and Ioana Koglbauer (2017a). “Pilots’ Flight Performance in a Virtual Reality Flight Simulator.” In: *Proceedings of the 32nd Conference of the European Association of Aviation Psychology*. Ed. by Michaela Schwarz and Julia Harfmann. Groningen, NL: European Association for Aviation Psychology, pp. 322–335. ISBN: 978-90-815253-5-0 (cit. on pp. 81, 83, 84, 87, 99, 100).

Oberhauser, Matthias, Daniel Dreyer, Reinhard Braunstingl, and Ioana Koglbauer (2017b). “Pilots’ Interaction with Hardware Controls in a Virtual Reality Flight Simulator.” In: *Proceedings of the 32nd Conference of the European Association of Aviation Psychology*. Ed. by Michaela Schwarz and Julia Harfmann. Groningen, NL: European Association for Aviation Psychology, pp. 565–575. ISBN: 978-90-815253-5-0 (cit. on pp. 81, 84, 85, 87–89, 96).

Oberhauser, Matthias, Daniel Dreyer, Reinhard Braunstingl, and Ioana Koglbauer (2018). “What’s Real About Virtual Reality Flight Simulation? Comparing the Fidelity of a Virtual Reality with a Conventional Flight Simulation Environment.” In: *Aviation Psychology and Applied Human Factors* (In Print) (cit. on p. 81).

Oberhauser, Matthias, Daniel Dreyer, Thomas Convard, and Sebastien Mamessier (2016). “Rapid Integration and Evaluation of Functional HMI Components in a Virtual Reality Aircraft Cockpit.” In: *Advances in Ergonomics in Design*. Ed. by Francisco Rebelo and Marcelo Soares. Cham, Switzerland: Springer International Publishing, pp. 17–24. ISBN: 978-3-319-41983-1. DOI: 10.1007/978-3-319-41983-1_2 (cit. on pp. xvii, 12, 66, 68, 152).

Oberhauser, Matthias, Daniel Dreyer, and Primož Kovačič (2015). “Rapid Prototyping in einem funktionalen, virtuellen Flugzeugcockpit.” In: *12. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung*. Ed. by Jürgen Gausemeier, Michael Grafe, and Friedhelm Meyer auf der Heide. Paderborn, Germany, pp. 9–20. ISBN: 978-3-942647-61-8 (cit. on pp. 44, 61).

Oberhauser, Matthias, Daniel Dreyer, Sebastien Mamessier, Thomas Convard, Daniel Bandow, and Axel Hillebrand (2015). “Bridging the Gap Between Desktop Research and Full Flight Simulators for Human Factors Research.” In: *Engineering Psychology and Cognitive Ergonomics*. Ed. by Don Harris. Vol. 9174. Lecture Notes in Computer Science. Cham, Switzerland: Springer International Publishing, pp. 460–471. ISBN: 978-3-319-20372-0. DOI: 10.1007/978-3-319-20373-7_44 (cit. on pp. xvii, 12, 13, 75, 107, 115, 118, 119, 150).

Occipital (2016). *Skanect: 3D Scanning: Fast, Easy and Low-Cost*. URL: http://skanect.occipital.com (visited on 02/14/2018) (cit. on p. 70).

Page, Ray L. (2000). “Brief history of flight simulation.” In: *SimTecT Proceedings 2000*. Ed. by Simulation Industry Association of Australia. Lindfield, Australia, pp. 11–17 (cit. on pp. 9, 10).

Parasuraman, Raja and Victor Riley (1997). "Humans and Automation: Use, Misuse, Disuse, Abuse." In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 39.2, pp. 230–253. ISSN: 0018-7208. DOI: 10.1518/001872097778543886 (cit. on p. 6).

Perfect, Philip, Mark D. White, Gareth D. Padfield, and Arthur W. Gubbels (2013). "Rotorcraft simulation fidelity: new methods for quantification and assessment." In: *The Aeronautical Journal* 117.1188 (cit. on p. 79).

Perry, Alexander R. (2004). "The flightgear flight simulator." In: *Proceedings of the FREENIX Track*. Berkeley, CA, USA (cit. on p. 25).

Persiani, Franco, Luca Piancastelli, and Alfredo Liverani (1997). "Virtual Flight Simulator." In: *Proceedings of the 10th ADM International Conference on Design Tools and Methods in Industrial Engineering*, pp. 693–701 (cit. on p. 16).

Poole, Alex and Linden J. Ball (2006). "Eye Tracking in HCI and Usability Research." In: *Encyclopedia of Human Computer Interaction*. Ed. by Claude Ghaoui. Hershey, PA, USA: IGI Global, pp. 211–219. ISBN: 9781591405627. DOI: 10.4018/978-1-59140-562-7.ch034 (cit. on pp. 7, 72).

Posner, Michael I. (1980). "Orienting of attention." In: *Quarterly journal of experimental psychology* 32.1, pp. 3–25. ISSN: 0033-555X (cit. on p. 7).

Prasad, J.V.R., Daniel P. Schrage, William Lewis, Daniel Wolfe, C.M. McKeithan, and C. Gardner (1991). *Development and Evaluation of Performance and Handling Qualities Critera for Low Cost Real Time Rotorcraft Simulators: Phase 1 Methodology Development*. Atlanta, GA, USA (cit. on p. 10).

Pritchett, Amy, B. Carpenter, Koji Asari, James K. Kuchar, and R John Hansman (1995). "Issues in airborne systems for closely-spaced parallel runway operations: Digital Avionics Systems Conference, 1995., 14th DASC." In: *Digital Avionics Systems Conference*, pp. 140–145. DOI: 10.1109/DASC.1995.482820 (cit. on p. 139).

Prothero, J. D. and H. G. Hoffman (1995). *Widening the field-of-view increases the sense of presence in immersive virtual environments: Human Interface Technology Laboratory Technical Report*. Ed. by University of Washington. Seattle, WA, USA (cit. on p. 14).

Quigley, Morgan, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng (2009). "ROS: an open-source Robot Operating System." In: *ICRA workshop on open source software* 3.3.2, p. 5 (cit. on p. 35).

Reason, James (1990). "The Contribution of Latent Human Failures to the Breakdown of Complex Systems." In: *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 327.1241, pp. 475–484. ISSN: 00804622 (cit. on p. 5).

Reason, James (2000). "Human error: models and management." In: *BMJ : British Medical Journal* 320.7237, pp. 768–770. ISSN: 1468-5833 (cit. on p. 5).

Rehmann, Albert J. (1995). *A Handbook of Flight Simulation Fidelity Requirements for Human Factors Research: DOT/FAA/CT-TN95/46*. Ed. by U.S. Department of Transportation Federal Aviation Administration. Springfield, VA, USA (cit. on pp. 10, 11).

Reibold, Holger (2011). *X-Plane kompakt: So lernen Sie den Umgang mit dem beliebten Flugsimulator ... schnell und einfach*. Professionelles Computer- und Internet-Know-How. Saarbrücken, Germany: Brain Media. ISBN: 9783939316671 (cit. on p. 25).

Reichelt, Stephan, Ralf Häussler, Gerald Fütterer, and Norbert Leister (2010). "Depth cues in human visual perception and their realization in 3D displays." In: *Three-dimensional imaging, visualization, and display 2010 and Display technologies and applications for defense, security, and avionics IV*. Ed. by John Tudor Thomas, Andrew

Malloy, Bahram Javidi, and Jung-Young Son. Proceedings of SPIE. Bellingham, WA, USA: SPIE. ISBN: 9780819481542. DOI: 10.1117/12.850094 (cit. on p. 55).
Reuzeau, Florence and René Nibbelke (2004). "Flight deck design process." In: *Human Factors for Civil Flight Deck Design*. Ed. by Don Harris. Farnham, UK: Ashgate, pp. 33–55. ISBN: 9780754613800 (cit. on pp. 11, 114).
Reveley, Mary, Jeffrey Briggs, Megan Thomas, Joni Evans, and Sharon Jones (2009). "An Examination of Aviation Accidents and Incidents Related to Integrated Vehicle Health Management." In: *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*. Aviation Technology, Integration, and Operations (ATIO) Conferences. American Institute of Aeronautics and Astronautics. ISBN: 978-1-60086-977-8. DOI: 10.2514/6.2009-7124 (cit. on p. 124).
Rheingold, Howard (1991). *Virtual Reality*. New York, NY, USA: Summit Books. ISBN: 9780671693633 (cit. on p. 12).
Richardson, Tristan, Quentin Stafford-Fraser, Kenneth R. Wood, and Andy Hopper (1998). "Virtual network computing." In: *IEEE Internet Computing* 2.1, pp. 33–38. ISSN: 1089-7801. DOI: 10.1109/4236.656066 (cit. on p. 68).
Richardson, Tristan and Kenneth R. Wood (1998). *The RFB Protocol: Version 3.8*. n.p. (cit. on p. 68).
Robertson, Adi and Michael Zelenko (2014). *Voices from the past: An oral history of a technology whose time has come again*. Ed. by The Verge. URL: http://www.theverge.com/a/virtual-reality/oral_history (visited on 03/02/2016) (cit. on p. 13).
Robinson, Andrew, Katerina Mania, and Philippe Perey (2004). "Flight simulation: research challenges and user assessments of fidelity." In: *Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*. Singapore: ACM, pp. 261–268. ISBN: 1-58113-884-9. DOI: 10.1145/1044588.1044644 (cit. on p. 10).
Rockwell Collins (2016). *Head-Up Guidance System™ (HGS) for midsize and light business aircraft: HGS-3500 white paper*. Ed. by Rockwell Collins. Cedar Rapids, IA, USA (cit. on p. 109).
Rodgers, Suzanne H. (2004). "Muscle Fatigue Assessment: Functional Job Analysis Technique." In: *Handbook of Human Factors and Ergonomics Methods*. Ed. by Neville A. Stanton, Alan Hedge, Karel Brookhuis, Eduardo Salas, and Hal W. Hendrick. London, UK: Taylor & Francis, pp. 12-1–12-10. ISBN: 9780415287005 (cit. on p. 131).
Rolfe, John M. and Ken J. Staples (1986). *Flight simulation*. 1st Ed. Cambridge aerospace series. Cambridge [Cambridgeshire] and New York: Cambridge University Press. ISBN: 9780521306492 (cit. on p. 10).
Rollon, Javier (2008). *A possible way to make an real hud in x-plane: still not tested...* URL: http://forums.xplanefreeware.net/topic/6976-a-possible-way-to-make-an-real-hud-in-x-plane/ (visited on 10/23/2016) (cit. on p. 110).
Rönnau, Arne (2016). *Was Bratwurst-Bot und autonome Autos gemeinsam haben: Robot Operating System: golem.de*. URL: http://www.golem.de/news/robot-operating-system-was-bratwurst-bot-und-autonome-autos-gemeinsam-haben-1612-124079.html (visited on 02/14/2018) (cit. on p. 159).
Rosero, J. A., J. A. Ortega, E. Aldabas, and L. Romeral (2007). "Moving towards a more electric aircraft." In: *IEEE Aerospace and Electronic Systems Magazine* 22.3, pp. 3–9. ISSN: 0885-8985. DOI: 10.1109/MAES.2007.340500 (cit. on p. 124).

Rude, Christoph (2014). *Head Mounted Displays & Datenbrillen: Einsatz und Systeme*. Ed. by Virtual Dimension Center VDC. URL: http://www.vdc-fellbach.de/files/Whitepaper/2014_VDC-Whitepaper_Head_Mounted_Displays_&_Datenbrillen.pdf (visited on 02/14/2018) (cit. on pp. 30, 31).
Russel, Ben (2016). *Gizmo64: A plugin for the X-Plane flight simulator.* URL: http://www.gizmo64.com (visited on 02/14/2018) (cit. on p. 60).
Salmon, Paul, Neville Stanton, Guy Walker, and Damian Green (2006). "Situation awareness measurement: A review of applicability for C4i environments." In: *Applied ergonomics* 37.2, pp. 225–238. ISSN: 0003-6870. DOI: 10.1016/j.apergo.2005.02.001 (cit. on pp. 8, 86).
Sarter, Nadine B. and David D. Woods (1991). "Situation awareness: A critical but ill-defined phenomenon." In: *The International Journal of Aviation Psychology* 1.1, pp. 45–57. ISSN: 1050-8414 (cit. on p. 6).
Sarter, Nadine B. and David D. Woods (1995). "How in the World Did We Ever Get into That Mode? Mode Error and Awareness in Supervisory Control." In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 37.1, pp. 5–19. ISSN: 0018-7208. DOI: 10.1518/001872095779049516 (cit. on p. 6).
Schmidt, Philipp (2011). "Konzeption eines Overhead-Panels." Master's Thesis. Munich, Germany: Bundeswehr University Munich (cit. on pp. 59, 125).
Selcon, Stephen J. and R. M. Taylor (1990). "Evaluation of the situational awareness rating technique (SART) as a tool for aircrew systems design." In: *Situational Awareness in Aerospace Operations (AGARD-CP-478)*. Neuilly Sur Seine, France: NATO–AGARD (cit. on p. 7).
Service de l'Information Aeronautique (2016). *Carte d'Aerodrome: Clermont Ferrand Auvergne*. Ed. by Service de l'Information Aeronautique. Mérignac, France (cit. on p. 113).
Shadish, William R., Thomas D. Cook, and Donald T. Campbell (2002). *Experimental and Quasi-experimental Designs for Generalized Causal Inference*. Boston, MA, USA: Houghton Mifflin. ISBN: 9780395615560 (cit. on p. 118).
Shappell, Scott, Cristy Detwiler, Kali Holcomb, Carla Hackworth, Albert Boquet, and Douglas A. Wiegmann (2007). "Human error and commercial aviation accidents: an analysis using the human factors analysis and classification system." In: *Human factors* 49.2, pp. 227–242. ISSN: 0018-7208 (cit. on p. 5).
Shaw, Stephen (2007). *Airline Marketing and Management*. 6th ed. Aldershot, UK: Ashgate Publishing. ISBN: 978-0-7546-4819-2 (cit. on p. 1).
Society of Automotive Engineers International (2001). *Aerospace Recommended Practices for Transport Category Airplane HUD Systems, ARP 5288*. Warrendale, PA, USA (cit. on p. 85).
Southard, David A. (1993). "Transformations for stereoscopic visual simulation." In: *Computers & Graphics* 16.4, pp. 401–410. ISSN: 0097-8493 (cit. on p. 55).
Stanney, Kay M., Ronald R. Mourant, and Robert S. Kennedy (1998). "Human factors issues in virtual environments: A review of the literature." In: *Presence* 7.4, pp. 327–351. ISSN: 1054-7460 (cit. on p. 14).
Steuer, Jonathan, Frank Biocca, and Mark R. Levy (1993). "Defining virtual reality: Dimensions determining telepresence." In: *Communication in the age of virtual reality*. Ed. by Frank Biocca and Mark R. Levy. Abingdon-on-Thames, UK: Routledge, pp. 33–56. ISBN: 978-0805815504 (cit. on p. 12).

Stock, Michael and Jim Deas (2009). *Fly-By-Wire for Experimental Aircraft? A Vision based on CANaerospace/AGATE Data Bus Technology*. Ed. by Stock flight systems and JAD Systems. n.p. (cit. on p. 47).

Stone, Robert J. (2000). "Haptic feedback: a brief history from telepresence to virtual reality." In: *Haptic Human-Computer Interaction: First International Workshop*. Ed. by Stephen Brewster and Roderick Murray-Smith. Berlin, Heidelberg, Germany: Springer Berlin Heidelberg, pp. 1–16. ISBN: 978-3-540-44589-0. DOI: 10.1007/3-540-44589-7_1 (cit. on p. 62).

Sutherland, Ivan E. (1965). "The ultimate display." In: *Proceedings of the International Federation of Information Processing (IFIP)*. London, UK: Macmillan and Co (cit. on p. 14).

Sutherland, Ivan E. (1968). "A head-mounted three dimensional display." In: *Proceedings of the 1968 fall joint computer conference*. New York, NY, US: ACM, pp. 757–764. DOI: 10.1145/1476589.1476686 (cit. on pp. 13, 14, 55).

Taymans, Wim, Steve Baker, Andy Wingo, Rondald S. Bultje, and Stefan Kost (2013). *GStreamer Application Development Manual* (cit. on p. 68).

Unity Technologies (2016). *Unity Documentation: Box Collider*. URL: https://docs.unity3d.com/Manual/class-BoxCollider.html (visited on 02/14/2018) (cit. on p. 59).

Vandel, Robert and Earl F Weener (2009). *Head-Up Guidance System Technology - A Clear Path To Increase Flight Safety: Special Report*. Ed. by Flight Safety Foundation. Alexandria, VA, USA (cit. on p. 109).

Vasigh, Bijan, Reza Taleghani, and Darryl Jenkins (2012). *Aircraft finance: Strategies for managing capital costs in a turbulent industry*. Fort Lauderdale, FL, USA: J. Ross Pub. ISBN: 978-1-60427-071-6 (cit. on p. 1).

Veltman, J. A. and A. W. K. Gaillard (1998). "Physiological workload reactions to increasing levels of task difficulty." In: *Ergonomics* 41.5, pp. 656–669. ISSN: 0014-0139. DOI: 10.1080/001401398186829 (cit. on p. 9).

VRNerds (2016). *VR Brillen Vergleich*. Ed. by Nico Uthe. URL: http://www.vrnerds.de/vr-brillen-vergleich/ (visited on 03/14/2016) (cit. on p. 31).

De Waard, Dick and Ben Lewis-Evans (2014). "Self-report scales alone cannot capture mental workload." In: *Cognition, Technology & Work* 16.3, pp. 303–305. ISSN: 1435-5566. DOI: 10.1007/s10111-014-0277-z (cit. on p. 9).

Wickens, Christopher D. (2002). "Situation Awareness and Workload in Aviation." In: *Current Directions in Psychological Science* 11.4, pp. 128–133. DOI: 10.1111/1467-8721.00184 (cit. on p. 6).

Wickens, Christopher D. (2008). "Situation awareness: review of Mica Endsley's 1995 articles on situation awareness theory and measurement." In: *Human factors* 50.3, pp. 397–403. ISSN: 0018-7208 (cit. on p. 6).

Wickens, Christopher D. and Jeffry Long (1995). "Object versus space-based models of visual attention: Implications for the design of head-up displays." In: *Journal of Experimental Psychology: Applied* 1.3, pp. 179–193. ISSN: 1939-2192(Electronic);1076-898X(Print) (cit. on p. 119).

Wickens, Christopher D., Patricia M. Ververs, and Steven Fadden (2004). "Head-up displays." In: *Human Factors for Civil Flight Deck Design*. Ed. by Don Harris. Farnham, UK: Ashgate, pp. 69–103. ISBN: 9780754613800 (cit. on pp. 108, 137).

Wiener, Earl L. (1989). *Human factors of advanced technology (glass cockpit) transport aircraft*. Moffett Field, CA, USA (cit. on p. 66).

Williams, Bruce (2012). *Scenario-based training with X-plane and Microsoft flight simulator: Using PC-based flight simulations based on FAA and industry training standards*. Indianapolis, IN, USA: Wiley. ISBN: 978-1118105023 (cit. on p. 51).

De Winter, Joost (2014). "Controversy in human factors constructs and the explosive use of the NASA-TLX: a measurement perspective." In: *Cognition, Technology & Work* 16.3, pp. 289–297. ISSN: 1435-5566. DOI: 10.1007/s10111-014-0275-1 (cit. on p. 9).

Woo, Mason, Jackie Neider, Tom Davis, and Dave Shreiner (1999). *OpenGL Programming Guide: The Official Guide to Learning OpenGL: Version 4.3*. 8th ed. Boston, MA, USA: Addison-Wesley Longman Publishing. ISBN: 0201604582 (cit. on p. 50).

Yavrucuk, Ilkay, Eser Kubali, Onur Tarimci, and Deniz Yilmaz (2009). "A Low Cost Flight Simulator Using Virtual Reality Tools." In: *Procedings of the AIAA Modeling and Simulation Technologies Conference*. Ed. by American Institute of Aeronautics and Astronautics. Reston, VA, USA: American Institute of Aeronautics and Astronautics, pp. 1–9. ISBN: 978-1-62410-161-8. DOI: 10.2514/6.2009-5832 (cit. on p. 16).