*aerospace*

MDPI

*Article*

# Safety Aspects of Supporting Apron Controllers with Automatic Speech Recognition and Understanding Integrated into an Advanced Surface Movement Guidance and Control System

**Matthias Kleinert [1,*], Oliver Ohneiser [1], Hartmut Helmke [1], Shruthi Shetty [1], Heiko Ehr [1], Mathias Maier [2], Susanne Schacht [2] and Hanno Wiese [3]**

[1] German Aerospace Center (DLR), Institute of Flight Guidance, Lilienthalplatz 7, 38108 Braunschweig, Germany; oliver.ohneiser@dlr.de (O.O.); hartmut.helmke@dlr.de (H.H.); shruthi.shetty@dlr.de (S.S.); heiko.ehr@dlr.de (H.E.)
[2] ATRiCS Advanced Traffic Solutions GmbH, Am Flughafen 7, 79108 Freiburg im Breisgau, Germany; mathias.maier@atrics.com (M.M.); susanne.schacht@atrics.com (S.S.)
[3] Fraport AG, Frankfurt Airport Services Worldwide, 60547 Frankfurt am Main, Germany; h.wiese@fraport.de
* Correspondence: matthias.kleinert@dlr.de; Tel.: +49-531-295-2567

**Abstract:** The information air traffic controllers (ATCos) communicate via radio telephony is valuable for digital assistants to provide additional safety. Yet, ATCos have to enter this information manually. Assistant-based speech recognition (ABSR) has proven to be a lightweight technology that automatically extracts and successfully feeds the content of ATC communication into digital systems without additional human effort. This article explains how ABSR can be integrated into an advanced surface movement guidance and control system (A-SMGCS). The described validations were performed in the complex apron simulation training environment of Frankfurt Airport with 14 apron controllers in a human-in-the-loop simulation in summer 2022. The integration significantly reduces the workload of controllers and increases safety as well as overall performance. Based on a word error rate of 3.1%, the command recognition rate was 91.8% with a callsign recognition rate of 97.4%. This performance was enabled by the integration of A-SMGCS and ABSR: the command recognition rate improves by more than 15% absolute by considering A-SMGCS data in ABSR.

**Keywords:** air traffic controller; simulation pilot; workload; assistant-based speech recognition; automatic speech recognition and understanding; apron control; STARFiSH



**Citation:** Kleinert, M.; Ohneiser, O.; Helmke, H.; Shetty, S.; Ehr, H.; Maier, M.; Schacht, S.; Wiese, H. Safety Aspects of Supporting Apron Controllers with Automatic Speech Recognition and Understanding Integrated into an Advanced Surface Movement Guidance and Control System. *Aerospace* **2023**, *10*, 596. https://doi.org/10.3390/aerospace10070596

Academic Editor: Michael Schultz

Received: 16 May 2023
Accepted: 19 June 2023
Published: 29 June 2023



## 1. Introduction

This article is an extended version of [1].

In air traffic control (ATC), there is a permanent need to increase the efficiency of handling air and ground traffic. This need exists especially at highly frequented airports such as Frankfurt. However, increasing efficiency must never come at the expense of safety. An important approach to increase efficiency and safety on the ground is by supporting ground traffic controllers' decision making through digitization and automation with new digital assistant systems that are integrated in or interoperate with advanced surface movement guidance and control systems (A-SMGCS) [2]. Currently, A-SMGCS already monitor data from different sensors and are designed to enable controllers to guide traffic more safely without reducing the capacity of traffic guidance.

### 1.1. Motivation

The most advanced digital assistants in apron control today already have access to a large number of sensors. Together with manual inputs from the controller, the digital assistants are able to detect potentially dangerous situations and warn the controller about

them. In contrast, voice communication between controllers and pilots, one of the most important sources of information in ATC, has not yet been used by these assistants. Whenever information from voice communication needs to be digitized, controllers are burdened with the additional task of entering this information into the ATC system. Research results show that up to one third of controllers' working time is spent on these manual entries [3]. This leads to a reduction in overall efficiency as controllers spend less time optimizing traffic flow [4]. The amount of time spent on manual inputs will even increase in the coming years as future regulations require more alerting functions to be implemented, hence more inputs are needed, especially in apron control, e.g., Commission Implementing Regulation (EU) 2021/116 [5].

Assistant-based speech recognition (ABSR) has already shown in the past that it is possible to significantly reduce manual input from controllers by recognizing and understanding the controller–pilot communication and automatically providing the required inputs into digital assistants [6]. ABSR technology has been continuously developed in several projects, and possible fields of application have been identified in the context of research prototypes. So far, there has been no initial integration into a commercial system to demonstrate that ABSR can also meet the corresponding requirements for safety and usability in a system network commonly used in aviation. In the German Federal Ministry of Education and Research funded project STARFiSH (Safety and Artificial Intelligence Speech Recognition), a powerful artificial intelligence (AI)-based speech recognition system was integrated into a modern A-SMGCS for apron control [1]. The solution was supposed to reduce the additional workload of controllers as much as possible by using speech recognition and understanding capabilities. At the same time, the solution was supposed to be objectively safe and be rated as reliable, easy to use, and safe by the controllers.

### *1.2. Related Work*

This section outlines related work for automatic speech recognition (Section 1.2.1) and understanding applications (Section 1.2.2) in air traffic control and closes with related work on how both are used to automatically fill digital flight strips or radar labels with voice information from the controller–pilot communication in Section 1.2.3.

#### 1.2.1. Early Work on Speech Recognition in Air Traffic Control

Speech Recognition in general has a long history of development. It started in 1952 when Davis, Biddulph, and Balashek of Bell Laboratories built a digit recognition system for a single speaker called "Audrey" [7]. Over the last 70 years, technological advances have led to dramatic improvements in the field of speech recognition. An overview of the first four decades is provided by, e.g., Juang and Rabiner [8]. Connolly from FAA was one of the first to describe the steps of using automatic speech recognition (ASR) in the air traffic management (ATM) domain [9]. In the late 1980s, a first approach to incorporate speech technologies in ATC training was reported [10]. Such developments led to enhanced ASR systems, which are used in ATC training simulators to replace expensive simulation pilots, e.g., FAA [11,12], DLR [13], MITRE [14], and DFS [15].

The challenges with ASR in ATC today go beyond basic training scenarios, where often standard procedures and ICAO phraseology [16] are followed very closely. Modern ASR applications have to recognize experienced controllers, who more often make deviations from the mentioned standards. Furthermore, applications with ASR also go beyond just the scope of simulation and training. ASR is for example used to obtain more objective feedback concerning controllers' workload [17,18]. A good overview of the integration of ASR in ATC is provided in the two papers of Nguyen and Holone [19,20].

#### 1.2.2. Speech Recognition and Understanding Applications in Air Traffic Control

In the recent past, research projects developed prototypical applications with speech recognition and understanding for all ATC domains from en route [21], via approach [4], to tower and ground [22]. These prototypes support controllers in maintaining aircraft

radar labels [23] and flight strips [22] to reduce workload, recognize and highlight aircraft callsigns [24], build safety nets for tower control [25], or even offer automatic readback error detection with reports to controllers [26,27]. The systems have matured to recognize words and meanings of real-life controller and pilot utterances even beyond the simulated environments [28]. The rules of how a speech understanding system can annotate the meaning conveyed with ATC radio transmissions are defined in an ontology that was agreed between major European air traffic management stakeholders in 2018 [29]. With this ontology, different word sequences can be mapped to unique word sequence meanings, e.g., the word sequences "lufthansa zero seven tango taxi via november eight and november to stand victor one five eight" and "zero seven tango via november eight november victor one five eight" both correspond to the same annotation in the ontology "DLH07T TAXI VIA N8 N, DLH07T TAXI TO V158".

#### 1.2.3. Related Work for Pre-Filling Flight Strips and Radar Labels

The information which air traffic controllers communicate via radio telephony is valuable for digital assistants to provide additional safety. Yet, controllers are usually burdened with entering this information manually. Assistant-based speech recognition (ABSR) has been shown to be a lightweight technology that automatically extracts ATC communication content without additional human workload and that successfully feeds digital systems [6]. DLR, together with Austro Control, DFS, and other European air navigation service providers, has demonstrated that pre-filling radar labels supported by automatic speech recognition and understanding reduces air traffic controllers' workload [3] and increases flight efficiency with respect to flight time and kerosene consumption. Fuel burn can be reduced by 60 L per aircraft in the approach phase [4]. DLR, Austro Control, Thales, and the air navigation service provider of Czech Republic have redesigned this exercise with a commercial off-the-shelf speech recognizer and an industrial radar screen. The exercise results clearly showed that speech recognition, i.e., obtaining the sequence of words from a voice signal, is not enough [30]. Speech understanding is needed for providing information for flight strips and radar labels.

Recently DLR and Austro Control analyzed the safety aspects of using speech recognition and understanding for pre-filling radar label contents. They investigated how many of the verbally spoken approach controller commands, with and without speech recognition, were finally entered into the ATC system and how many errors were made, not recognized, or not corrected by the air traffic controllers. Despite manual corrections of commands even with speech recognition and understanding support, about 4% of the spoken commands were still not correctly entered into the system. However, this result, which is initially alarming from a safety point of view, is quickly put into perspective, when considering that roughly 10% of the verbally spoken commands are incorrectly or not entered at all into the system, if no speech recognition and understanding support is available. More details are provided in [23]. The results show that speech recognition and understanding [31] is far from being perfect, but a system without speech recognition and understanding seems to be even further away.

One of the main input sources for this paper, which describes the results or transforming the support tool for approach controllers to apron controllers, were two studies of DLR: one from 2015 for the Dusseldorf approach [4] and a recent one for the Vienna approach control [23]. It was expected that the good results of command recognition in the approach area will translate one-to-one to the correctness and completeness of inputs in the apron area. Previous projects have already taken first steps towards using speech recognition and understanding in a tower or apron environment, which included for example the prediction of potential controller commands [32]. The actual use of speech recognition and understanding was then further investigated in a multiple remote tower setup [33]. In the process, relevant information for digital flight strips was automatically derived and entered from the given verbal commands. The tower environment already covered

many of the command types relevant for ground/apron traffic such as taxi, hold short, and pushback instructions.

### *1.3. Paper Structure*

Section 2 summarizes the use case of supporting apron controllers, the iterative software development approach and introduces the Software Failure Modes, Effects, and Criticality Analysis. Section 3 describes the final version of the evaluation system. Section 4 explains the validation of the developed application. Section 5 presents the validation results before Sections 6 and 7 finalize the paper with discussions and conclusions.

## 2. Materials and Methods

### *2.1. Application Use Case of Supporting Apron Controllers*

Initially, the apron controller, shown as "ATCo" in Figure 1, issues a command to the pilot by radio. Without ABSR (Figure 1, left), the controller enters this command into the A-SMGCS manually either before, afterwards, or in parallel to the radio call so that the system can provide automation functions. With ABSR (Figure 1, right), an ABSR-system automatically generates, based on the radio call, a data packet including metadata from the command, which is sent to the A-SMGCS. The A-SMGCS executes valid commands and highlights the changes together with the associated aircraft symbol. No system interaction by the controller is required unless an error has occurred. If the automatic speech recognition fails, the controller needs to manually correct or enter the command in the same way as without the ABSR system.

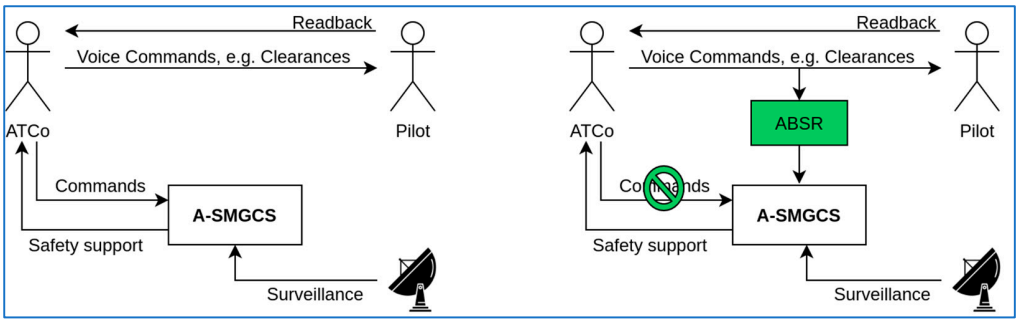


**Figure 1.** Interaction between human operators and digital systems in ground control without an ABSR component (**left**) and with an ABSR component (**right**).

The automatic recognition of voice commands issued by controllers to pilots should provide a solution to the problem that controllers are less able to keep an eye on traffic when they enter the information of the radio call that is necessary for modern A-SMGCS support functions. This includes, for example, entering taxi routes so that compliance can be monitored automatically.

From a technical point of view, the sequence of actions is:

1. Commands given via voice by the controller to the pilot are recorded as an audio data stream (A/D conversion of utterances).
2. The audio stream is divided into sections by detecting individual transmissions in the audio data.
3. Speech-to-text (S2T) transformation is applied on the resulting audio sections. S2T is based on neural networks trained with freely available data as well as with domain-specific recorded audio data for the target environment.
4. Relevant ATC concepts are automatically extracted from the S2T transcription using rule-based algorithms on a previously defined ontology and traffic data fed from the A-SMGCS.
5. High-level system commands are generated from the extracted ATC instructions using rules algorithmically interpreted from operational necessities according to the current traffic situation and fed into the system.

6. The changes to the system state resulting from the high-level system commands are presented to the human operators.
7. Human operators can correct or undo the automatic inputs.

We explain these steps by an example:

1. The apron controller is continuously speaking to the pilots with some gaps in between, e.g., " ... to seven five seven from the left ... lufthansa four two two good morning behind opposite air france three twenty one continue november eight lima hold short lima six ... austrian one foxtrot behind the passing". The gaps occur either because no further action is required or due to the verbal response of the (simulation) pilot, which is not available to the ABSR.
2. The audio stream sections are detected, and one continuous transmission could then be "lufthansa four two two good morning behind opposite air france three twenty one continue november eight lima hold short lima six".
3. Let us assume that the result of S2T contains some errors and results in the word sequence: "lufthansa four **to** two good morning behind opposite air **frans** three twenty one continue november eight lima **holding** short lima six" (errors marked in bold).
4. The relevant ATC instruction, being extracted by ABSR even with the errors from S2T, would be:
    a. DLH422 GREETING;
    b. DLH422 GIVE_WAY AFR A321 OPPOSITE;
    c. DLH422 CONTINUE TAXI;
    d. DLH422 TAXI VIA N8 L;
    e. DLH422 HOLD_SHORT L6.
5. The GREETING is ignored by the A-SMGCS. For the GIVE_WAY instruction the A-SMGCS may find out that the A321 from the opposite is the callsign AFR2AD. A symbol is generated in the human machine interface (HMI) of the apron controllers (and the simulation pilots), showing that DLH422 is waiting until the AFR2AD has passed. The continue statement is executed after the give way situation is resolved. The route along the taxiways N8 and L is shown. A hold short (stop) is displayed before taxiway L6.
6. In summary, the following visual output is shown to the apron controller:
    a. The aircraft symbol of DLH422 is highlighted;
    b. A GIVE_WAY symbol between the two aircrafts;
    c. The taxi route via N8 and L;
    d. A HOLD_SHORT symbol (stop) at L6.
7. The apron controller can accept or reject all three above options or can change some or all of them.

For the controller, almost all processing steps are invisible. Technology remains in the background. From the human operator's point of view, the sequence of actions is like this:

1. The callsign addressed in the controller's radio call is highlighted at the corresponding aircraft symbol in the A-SMGCS (DLH422 in the above example).
2. Once the commands to the pilot are fully uttered, they are converted into corresponding system commands that would otherwise have to be manually entered, e.g., a taxi route.
3. The result of the command input is displayed to the controller (and/or simulation pilot) in the A-SMGCS. Wherever possible, the visualization corresponds to the same visualization that would have resulted from a manual entry.
4. Special case: If an error in the data processing causes the wrong command to be sent and therefore the wrong effects (or none) to be displayed, the human operator must manually correct the command or enter it into the system. Depending on the type of command, dedicated buttons are offered for this purpose.

### 2.2. Application Development

The solution was created in four main iterations following the spiral model of software development [34]. For this purpose, an ABSR system was integrated with the A-SMGCS system TowerPad™ (see Figure 1) by iterating the following steps:

- Technical and operational requirements were determined;
- Software and interfaces were developed, implemented, and tested;
- Progress was validated by users in realistic operational scenarios in Fraport's training simulator;
- Results were analyzed to derive new requirements.

In the end, the system was intensively validated in realistic simulations with apron controllers and evaluated based on recorded data and defined metrics. The safety aspects detailed in the next subsection were addressed and focused on in the third iteration.

### 2.3. Safety Considerations

In aviation, a system can only be approved for operation if its impact on safety has been thoroughly assessed. This is even more important if it uses technology that is new and for which the currently available safety assessments are not necessarily suitable. For the use of artificial intelligence-based methods, discussions are taking place in the community regarding how the safety of AI methods can be verified or demonstrated. These discussions happen independent from air traffic management application areas.

However, there is a way out of the dilemma of the lack of approved testing and verification methods, which we saw in the STARFiSH project as a possibility to safely operate a system with AI-based speech recognition and understanding. If the AI system can be encapsulated in such a way that safety-critical outputs cannot have an immediate impact on real-world operation and must always be approved by the user of the system prior to implementation, safety will be verified during operation. However, manual checking of commands means additional effort that one does not want to impose on users for system inputs that cannot have any safety-critical consequences. Thus, it was important to identify which commands have effects that are safety-critical from an operational view.

In order to determine, which system inputs are safety-critical in this sense and which are not, a safety analysis based on the classification in Figure 2, must be performed that first determines, independently of the solution, which system inputs are potentially safety-relevant because they can endanger operational safety. For this analysis, we followed the EUROCONTROL "safety assessment methodology" (SAM) and applied the SFMECA methodology that is at the core of the "functional hazard analysis" (FHA).

Rule-based classification and execution based on safety criticality:

| Rules | **Recognition good** | **Recognition bad** |
|---|---|---|
| **Command non-critical** | Implement command | Implement but offer undo |
| **Command critical** | Implement but warn | Ask ATCO for help |

**Figure 2.** Naïve safety classification of ATC commands regarding safety criticality and recognition quality as suggested before execution of the project.

SFMECA (Software Failure Modes, Effects, and Criticality Analysis) [35] is a formalized method of risk assessment and subsequent identification of mitigation measures. It

is a bottom-up method that analyzes so-called failure modes and their effects to identify (hidden) hazards at the system level. Using the standardized structure and presentation of the process and results specified by the SFMECA, a team of experts used predefined and individual "failure modes" to analyze which safety-relevant effects could be caused by the software and what their causes were for the functional requirements in the project. The requirements were grouped according to features and pre-filtered according to the evaluation dimensions:

- Safety-criticality;
- Criticality for the work of the controller;
- Risk due to potential software development errors.

Then the error cases (in categories "functionality", "timing", "sequencing", and "data, error handling") were quantitatively evaluated with their respective "root cause", checking for 26 common causes plus specific functional errors, e.g., "misdetection of callsign under own jurisdiction".

The evaluation criteria were the severity of the effects, their probability of occurrence, and the probability of timely detection of the error case. Each of these criteria was evaluated in ten gradations for each failure mode and its cause, and a risk priority number (RPN) was calculated. For sufficiently high RPNs, the SFMECA provides steps to be defined on how to reduce the risk with mitigation actions. In the project, the mitigation action envisaged was to let the controller decide on such commands instead of executing the commands directly.

Section 5.7.1 presents the results of the SFMECA and even shows that in our use case, the distinction into good and bad recognitions is not needed.

## 3. Description of Evaluation System

The final validation trials were conducted on five consecutive days in the apron simulator in Frankfurt in summer 2022. All necessary data were recorded, subsequently processed, evaluated, and documented along the agreed validation concept.

For the trials, an evaluation system was created that allowed us to test the hypotheses set from the project description and to adapt them to the experience gained. The following section describes the final evaluation system as it was integrated into the Fraport apron simulator.

### *3.1. Technical Integration into the Simulator*

While the validation system was necessary to perform the final validation trials, its design was developed in iterations. From the start of the project and as a very basic technical integration, it was used to test the planned system's architecture and functionality. It was also used to record speech and validation data necessary for the iterative improvement of artificial intelligence (AI)-based speech recognition and understanding during each training session. The actual speech data were recorded, and the A-SMGCS position data, flight plan data, and commands entered by the simulation pilot were logged. Additionally, the recorded speech data were transcribed (a word-for-word transcript of the uttered speech) and annotated (information on the contained commands in the defined ontology). The recorded data were used to train the speech recognition models and adapt the algorithms for speech understanding and callsign prediction. The data used for training and adaptation contained 19 h of audio data without silence from 14,567 single utterances, aligned with corresponding transcriptions. Furthermore, around 8.5 h with respect to 7132 utterances of the transcribed data were annotated in the defined ontology. Both the transcription and annotation processes are based on automatic pre-transcription/pre-annotations generated by the speech recognition and understanding components in the quality available within the different iterations. The manual verification and correction of the pre-transcripts were executed by a human expert from Fraport who is familiar with the airport layout, the procedures, and so on. The pre-annotations were verified and corrected by experts from the DLR, which are familiar with the defined ontology and its components.

The part of the Fraport simulator that was used in the project consists of a simulation room for the apron controllers and a control room for the simulation pilots (see Figure 3).

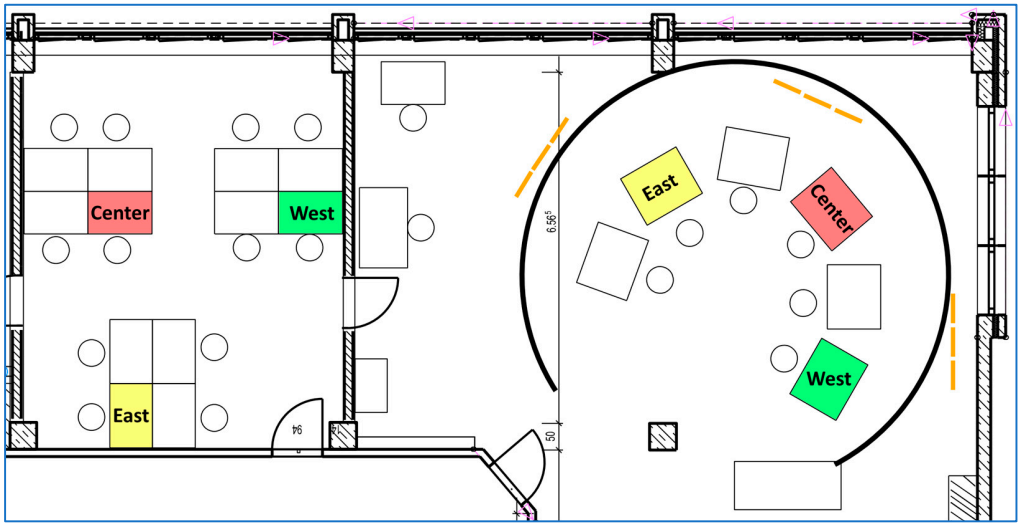


**Figure 3.** Simulation rooms for controllers and simulation pilots at Fraport. The left part shows the working positions of the three simulation pilots. The right part shows the simulation environment for the three apron controllers.

In operational mode, there are two different workstations. The Movement Controller (MC) workstation guides the aircraft. English is spoken on the flight frequency. In Frankfurt, three Movement workstations are usually manned, named East, Center, and West (see Figures 3 and 4). In addition to the Movement workstations, there are Operational Safety Controllers (OSC). These workstations guide the tugs and assign the follow-me vehicles. German is spoken on these frequencies. For training, usually the MC and two OSC are assigned and split in two different rooms. It was decided to not use OSC during simulation, so that five instead of three simulation days were possible with the same effort of the involved apron controllers. Everything was located in one simulation room.

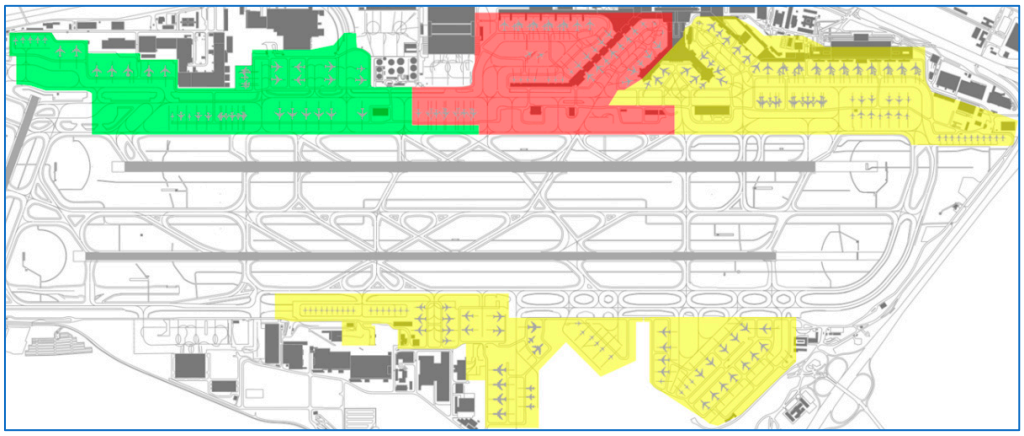

**Figure 4.** Areas of responsibility for the East (yellow), Center (red), and West (green) workstations for the Frankfurt apron control.

In the simulator environment, simulation pilots act as counterparts for the controllers. They sit in the simulation pilot room (left part in Figure 3). The task of the simulation pilots is to move the aircraft as instructed by the controller and to provide readback of uttered commands. The simulation pilot is in control of the same aircraft as the controller, and, therefore, controls several aircraft. The simulation pilots, like the controllers, are assigned to designated work areas (East, Center, and West). Thus, the controller always talks to the

same simulation pilot during a simulation session and vice versa. Three MC workstations and three active simulation pilot positions were evaluated through ABSR support.

To use ABSR in the simulator like in real operations, various data had to be exchanged between the simulator software (ATRiCS AVATOR™), the A-SMGCS (ATRiCS TowerPad™), and DLR's ABSR system (see Figure 5).

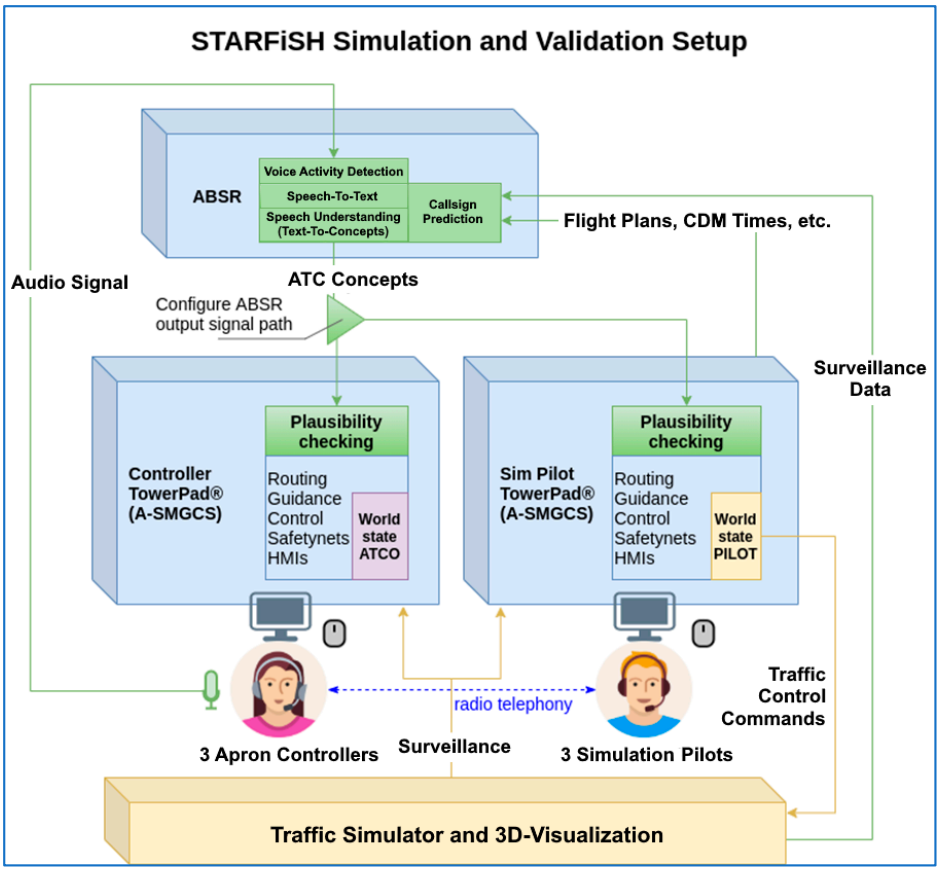


**Figure 5.** Simulation setup for validation trials with ABSR in an A-SGMCS system.

The surveillance data in Asterix CAT 20 format as well as flight plans and Collaborative Decision Making (CMD) Times were sent to the ABSR system and evaluated by the callsign prediction module. The audio recordings were first processed by the voice activity detection, then by the speech-to-text component, and finally by the speech understanding component. These results (ATC concepts) were forwarded as commands to the A-SMGCS and simulation pilot workstations. The latter control the traffic and radar simulator and visualization. This was performed by means of a transmission control protocol/internet protocol (TCP-IP) connection. Another interface was used to transmit flight plan data from the simulator to the ABSR system. The main interface was from the ABSR system to the A-SMGCS. Here, the recognized commands were passed to the A-SMGCS for visual display on the simulation pilot workstation and on the apron controller workstation, respectively. The interfaces and software programs as well as traffic scenarios were successfully tested in the first iteration of the project.

After the first iteration, new functions were integrated and tested in the simulator and the ABSR system in short intervals. In this way, it was possible to quickly check whether the interaction of the software worked and whether the adjustments represented an improvement or had no significant or even negative effects and should, therefore, be removed again.

Similar to the development process, an iterative approach was also taken to evaluate the results. An evaluation basis was defined and tested during the iterations and continuously improved. During these tests, it was determined that an objective measure of the

cognitive load placed on controllers by system inputs was needed. Using eye-tracking sensors would have been one way to measure how often the controller's gaze is on the implemented ABSR output, e.g., to provide manual input and to determine how often the simulated traffic can be observed from an outside view. However, after further experiments, the decision was made to use a much less complex measurement method by means of secondary tasks for the participating controllers, see Section 4.4.

In the simulations, different traffic situations were used as scenarios. Scenarios from 30 min to 60 min were tested, as well as high, medium, and low traffic. After various tests, the length of 30 min and very high traffic density seemed to be most suitable to validate or falsify the validation hypotheses in the final validation trials. Two scenarios were created for the final validation trial. One scenario included runway operating direction 25, another one, operating direction 07. The two different operating directions indicate the direction in which the parallel runway system in Frankfurt is used, i.e., the direction in which aircraft take off and land. The direction depends on the weather, in particular on the wind, since landings should be made against the wind direction if possible. During operating direction 25, the runways 25 left (25 L) and 25 right (25 R) were used for inbounds/arrivals. The runways 18 and 25 center (25 C) were used for outbounds/departures. During operating direction 07, inbounds used 07 L and 07 R, whereas outbounds used 07 C and 18, i.e., in both scenarios, two inbound and two outbound runways were in use. On the ground, the operating directions affect the taxi guidance since the aircraft are then guided on other taxiways to the stand or runway. Accordingly, the ABSR and the integration of the systems could be tested in different situations. Consequently, the results should be more general and transferable to other traffic scenarios and other airports.

### *3.2. Assistant-Based Speech Recognition*

The core of the ABSR system implemented in the STARFiSH project mainly consists of three modules (see Figure 6), which perform the conversion of the audio signal into recognized word sequences (speech recognition), the prediction of the relevant callsigns (callsign prediction), and the extraction of the semantic meaning of apron controller commands (speech understanding).

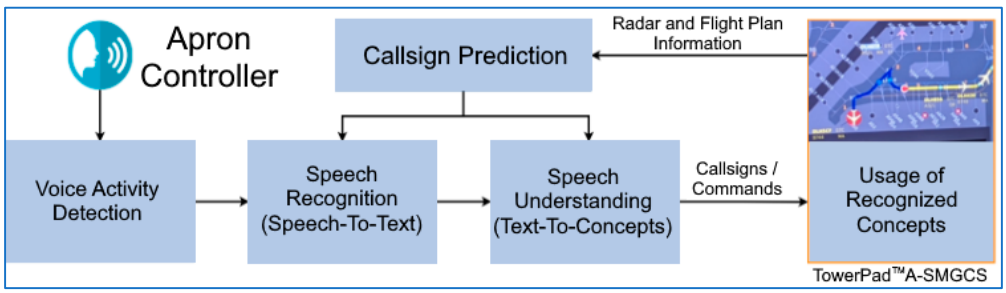


**Figure 6.** Modules of assistant-based speech recognition.

The only mandatory input signal to the system is the voice radio of the apron controller. To improve the recognition quality of the ABSR system, radar and flight plan information is also provided by the A-SMGCS. These data allow the generation of relevant contextual information, such as the list of aircraft callsigns that are currently relevant for operations per area of responsibility, that can be directly integrated into the recognition process of the ABSR system. In addition to the three central modules, for technical reasons, see Section 3.2.1, the project also had to implement and integrate a voice activity detection for the ABSR system, which determines when a controller's radio transmission starts and when it ends. Figure 6 provides an overview of the interfaces between the core modules. The following sections describe each module in more detail.

#### 3.2.1. Voice Activity Detection

The goal of the ABSR system in STARFiSH is to recognize and understand the uttered commands of apron controllers. Since the audio signal is transmitted as a continuous stream of data from the voice communication system to the ABSR system, even when there is no speech at all, the system needs a way to detect the points in time a dedicated radio transmission has started and ended. Therefore, a signal is required that indicates the beginning and end of a radio message to the ABSR system. The most precise signal for this purpose would be the so-called push-to-talk (PTT) signal. This signal is triggered by controllers each time they push or release the button on the microphone they are using to start or end a radio transmission to a pilot. However, for technical reasons, the PTT signal could not be accessed for use in this project. To compensate for this problem, STARFiSH uses voice activity detection, i.e., the acoustic signal is analyzed to determine when a transmission begins and ends. The start and end of radio transmissions are detected based on the duration of previously detected silence states and a probability of reaching the end of the voice signal. Five predefined rules for detecting the end of a segment online from Kaldi have been considered without further adaptations [36].

#### 3.2.2. Speech Recognition, i.e., Speech-to-Text (Transcriptions)

As soon as the voice activity detection detects the beginning of a radio transmission, the audio signal is forwarded to the S2T component, and the recognition process immediately starts converting the audio signal into word sequences. This means that the speech recognition system starts the recognition process as soon as the apron controller begins the radio transmission. The system then continuously provides intermediate recognitions until the controller reaches the end of the radio transmission. For example, a controller might say the following:

> *"lufthansa three charlie foxtrot taxi alfa six two alfa via november one one november november eight at november eight give way to the company A three twenty from the right".*

Let us assume that this sentence could be recognized and output by the S2T component in the following increments:

1. *"lufthansa three charlie";*
2. *"lufthansa three charlie foxtrot taxi alfa six two alfa via";*
3. *"lufthansa three charlie foxtrot taxi alfa six two alfa via november one one november november eight";*
4. *"lufthansa three charlie foxtrot taxi alfa six two alfa via november one one november november eight at november eight give way to";*
5. *"lufthansa three charlie foxtrot taxi alfa six two alfa via november one one november november eight at november eight give way to the company A three twenty from the right".*

The speech recognition engine is implemented as a hybrid deep neural network combined with a hidden Markov model (HMM). It is combined with a convolutional neural network factorized time delayed neural network (CNN-TDNNF) with six convolution layers and fifteen factorized time-delay neural networks. Overall, the model has around 31 M trainable parameters. The whole model is trained with a so-called "lattice-free maximum mutual information" as an objective function. The system follows the standard chain LF-MMI training recipe [37] of Kaldi [38], which uses high-resolution "Mel frequency cepstral coefficients" and i-vectors as input features. A typical 3-gram language model was trained and adapted using domain-specific data.

Starting from a base model, the speech recognition engine was continuously improved with new training data during the course of this project. An integration of context knowledge from callsign predictions was also implemented and contributes to the improvement of the recognition performance.

3.2.3. Speech Understanding

When a word sequence is transmitted from the S2T component, it is analyzed by the speech understanding module and converted into relevant ATC concepts, as originally defined in an ontology [29]. According to this ontology, the above word sequence "*lufthansa three charlie foxtrot taxi alfa six two alfa* via *november one one november november eight at november eight give way to company A three twenty from the right*" is transformed into the following commands:

- DLH3CF TAXI TO A62A;
- DLH3CF TAXI VIA N11 N N8;
- DLH3CF GIVE_WAY DLH A320 RIGHT WHEN AT N8.

In total, this radio transmission contains three commands. Here, the pilot of the aircraft with the callsign DLH3CF was instructed to taxi to parking position A62A via the taxiways N11, N, and N8. When arriving at taxiway N8, the pilot of the aircraft must give way to a Lufthansa (DLH), which is from the same company as the pilot addressed, has the aircraft type A320, and is coming from the right (RIGHT), before being allowed to continue taxiing.

In the case of intermediate detections from the speech recognition engine, the speech understanding module is able to provide early recognition of the callsign or, if required, early recognition of subsequent commands. The speech understanding implementation is based on a rule-based algorithm that identifies the relevant parts step by step and converts them into ATC commands. For more information, see [39].

The speech understanding module does not only convert the word sequences into ATC concepts but also makes an initial decision as to whether the extracted commands might be erroneous. Potentially erroneous commands are caused either by an erroneous interpretation of the rule-based algorithm, by an already erroneous word sequence due to a misrecognition of the speech recognition engine, or by misleading formulations of the controller. The decision, whether a command could be erroneous, is based on simple heuristic rules that determine which commands can occur together in a radio transmission. Here are some of the rules, explained by examples:

- It is logically not possible that an aircraft is instructed in a single radio transmission to taxi to two different target positions, e.g., a "TAXI TO" to two different parking positions, runways, or both in one transmission is impossible. Therefore, the module would automatically discard all "TAXI TO" commands within the transmission. Of course, with more information, it might be possible in some cases to determine which of the target positions is the correct one and only neglect one of the "TAXI TO" clearances, but that would require quite complex knowledge about the airport infrastructure to be implemented within the speech understanding component. The target application, on the other hand, which receives information from speech understanding, usually already has the required knowledge about the airport and therefore is more suitable to handle this task.
- A similar example would be a "TURN LEFT" and a "TURN RIGHT" command within one transmission and no other command in between, which is also impossible and would therefore be neglected for the same reasons.
- A less obvious example is the recognition of a "PUSHBACK" and a "TAXI TO" command in one transmission. Theoretically this might seem possible, but also these commands do not appear together and if they do, the error is usually a wrongly extracted "TAXI TO". Therefore, the heuristic says to always neglect the "TAXI TO" in this case.

However, the examples above show also that erroneous commands can only be detected if the error case is predefined. Therefore, confidence measures have furthermore been implemented for speech understanding output and are used to reduce possible false recognitions. These confidence measures can also be applied to the error cases listed above instead of neglecting the erroneous commands, but this requires that the application receiving the information is able to implement it. This means that the application then has

to determine which command to neglect or not. In the end, all errors that are not detected, either by speech understanding or by the application, have to be handled manually by the apron controller in charge.

Analogous to speech recognition, speech understanding was also continuously developed and adapted based on new information. Just as in speech recognition, an integration of context knowledge from callsign prediction takes place and contributes to the improvement of recognition performance.

#### 3.2.4. Callsign Prediction

Callsign prediction receives both radar and flight plan information from the A-SMGCS. The module uses these data to determine which callsigns may be part of a radio transmission in the near future. The radar information is used in the first step to obtain an overview of the available callsigns in the airport area. However, since many aircraft are in the airport area, but not all will be actively participating in taxiing traffic in the near future, the module also uses flight plan information dynamically provided by the A-SMGCS to determine more precisely which of the available callsigns will be addressed in the near future. For this purpose, the responsible controller position, the target startup approval time (TSAT), the actual take off time (ATOT), the actual landing time (ALDT), and the actual in block time (AIBT) are extracted from the flight plan. All relevant callsigns are forwarded to the speech recognition module (callsign boosting) and the speech understanding module to include the callsigns in the process of recognition and understanding. More information on the technique of callsign boosting, used within the speech recognition module to enhance recognition, can be found here [40,41]. The integration of callsign predictions in the speech understanding module transforms the callsigns into possible word sequences and calculates the closest match to the recognized word sequence based on the Levenshtein distance [42] to determine the correct callsign.

#### 3.2.5. Concept Interpretation

The final stage of integrating an ABSR system into an A-SMGCS represents the testing for operational plausibility, interpretation, and implementation of the extracted concepts or commands. Figure 7 shows the running integration of ABSR into the A-SMGCS at one of the simulation pilot stations.

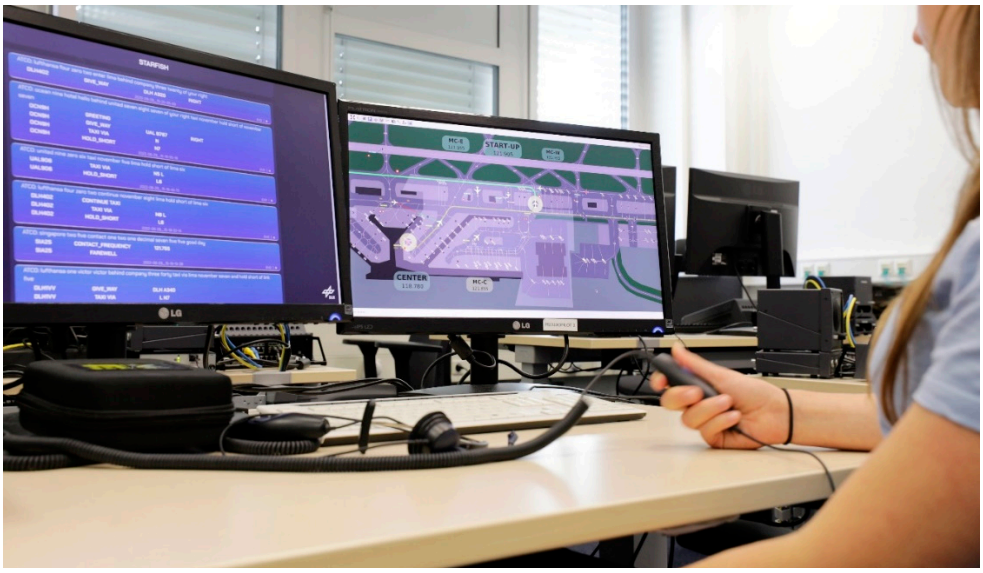


**Figure 7.** ABSR output log on the left screen and airport map for simulation pilot on the right (photo © Fraport).

Testing and interpretation are necessary prior to implementation for two reasons:

- Controller instructions via voice convey exactly the information that is necessary and sufficient for the addressed pilot in the current traffic situation. Globally, however, these instructions can be ambiguous. It is, therefore, necessary for an information technology system to unambiguously identify the addressed pilot and to make assumptions about his/her contextual knowledge in order to be able to exclude ambiguities from this perspective. A GIVE_WAY command from the right could identify several aircraft that approach from the right at the same time or consecutive taxiway crossings. The system has to determine the correct one that is implied from the traffic context.
- The extracted concepts may be erroneous. Either the controller has made a mistake, so that the verbal instruction does not correspond to what would be advised in the current traffic situation, or errors have occurred in the recording of the speech, the pause recognition, the conversion to text, or the speech understanding, so that the extracted concept is erroneous and should not be implemented.

These two sources of error cannot be distinguished. The task of this module is to admit only those commands into the assistance system that are plausible and fit into the current traffic context. If inappropriate commands are delivered by the ABSR system, the user must be given the opportunity to manually correct the error. This is technically implemented by the following steps, which are detailed in Appendix B:

1. Preprocessing;
2. Highlight the aircraft symbol on the basis of the recognized callsign;
3. Trigger multiple actions based on a single command;
4. Discard commands incompatible with the traffic situation;
5. Correctly interpret context-dependent commands;
6. Complete incomplete commands from the current traffic situation;
7. Convert commands;
8. Deal with detected errors;
9. Deal with undetected errors and identify error sources.

### *3.3. Usability Considerations*

A key element to the successful implementation of automation features is the user interface. Since automation reduces the necessary interactions with the system, users may miss automatically executed actions. It is thus essential that users are still able to see all system states that are relevant for safe operations. In addition, it must be possible to quickly analyze and correct errors in the event of automation failure. This is a necessary requirement of operational safety, especially in aviation, where errors can lead to accidents.

In the STARFiSH project, the automation functions as well as the user interface were first implemented in a purely functional way and then analyzed in operation with the end users to iteratively overcome the challenges. This involved asking questions such as: Is the right information available, and is the right information being perceived? Is the user interface not overloaded with information, i.e., is important information also perceived more easily than less important information? Are delays sufficiently low?

Using various elicitation techniques (observation, brainstorming, and interviews), user requirements were thus determined in an iterative manner, and new target formulations were achieved. In total, 30 users participated in 29 evaluation days, distributed over the four iterations in the course of this project.

#### 3.3.1. Visualization of the Automation Actions (Feedback)

In iteration 1, the main focus was to be able to check the technical integration of the systems, i.e., to investigate the question of whether recognized commands reach the human operator and are available in time from an operational point of view. Therefore, the following were implemented first:

- Display the recognized transcriptions and the resulting annotations (ATC commands) on the side of the ABSR output log, in order to be able to compare the output of the ABSR system with the received data on the TowerPad™.
- Log data at the interfaces of the ABSR system and on the working position computers of the controllers and simulation pilots, in order to be able to analyze, after the simulation runs, whether the correct commands arrive in time.
- Log the commands provided by the ABSR system in chronological order on the working position computers to give users and researchers a way to observe and verify the results of the speech inputs independently of the implementation of the commands.

During preliminary testing, it became apparent that displaying each recognized command in a table to support easy troubleshooting did not add value to the operational users of the system but was distracting. Therefore, the user interface was designed so that commands generate specific visual feedback which is integrated into the workflow. In terms of position and design, this resulted in symbolic displays specifically adapted to the command or very compact dialogs. Although the display as a table was still extended for troubleshooting, it was no longer visible at the controllers' working positions during the trials and was positioned on a second screen outside the focus of the users at the simulation pilot workstations. It was only used for evaluation and development.

Starting with iteration 3, commands were directly translated into visible actions of the system. For some actions, it was possible to use the same visual feedback to the human operator that is used for manual input, for example:

- Change a route;
- HOLD_SHORT command;
- GIVE_WAY command.

There are fundamental advantages to displaying the same feedback in the HMI regardless of the input method (by speech recognition and understanding, or by mouse or touch gesture), as there is less need for training. On the other hand, users should be able to identify if the source of a change in the user interface is the speech recognition and understanding component. This was implemented by the following user interface features:

- Highlighting of the addressed aircraft symbols without disturbing user touch or mouse input, executed in parallel, additionally multi-highlighting when several commands are executed in quick succession.
- Feedback for changes, which are scarcely visible when executed manually, such as the transfer of an aircraft to another working position.

#### 3.3.2. Manual Error Correction

The simulation experiments showed that for some actions, an undo is ambiguous and not without side effects, e.g., when changing a taxi route. For these actions, it was easier for human operators to select the desired function directly without prior "undo", thereby implicitly overriding the wrong action.

## 4. Validation Trials

This section presents the preparation and results of the validation trials. All simulations took place in Fraport's training simulator, which had been retrofitted for the experiments and tests.

### *4.1. Pre-Simulations*

During the pre-simulations, the individual parts of the system and their integration were tested, and exemplary evaluations of the simulation runs were carried out in order to determine methods for the final validation trial. The basic structure, i.e., the architecture, remained constant after the initial integration tests.

Controllers and pilots, in their corresponding positions, speak to each other on the same radio frequency. The ABSR system operates on the voice recordings of the controllers,

and the command implementation takes place independently in the two instances of the A-SMGCS for the controllers and simulation pilots, respectively (see Figure 5). For both groups, ABSR support is enabled either for all working positions or for none.

The simulations in the first iterations served the dual purpose of obtaining feedback from the human operators and testing the technical integration. In later iterations, the validation methods themselves were tested as well, i.e., exemplary evaluations of the simulation runs were performed. For example, the hypotheses regarding the reduction of taxi times were discarded, since they did not differ significantly.

It was also explored what the scope of traffic should/should not be, and which additional tasks are suitable to challenge the attention of the users without tying up the support team too much.

### 4.2. Validation Plan

Four different combinations of the ABSR support were investigated, as shown in Table 1:

**Table 1.** Different combinations of ABSR support investigated during validation trials.

| Condition Name | Operational Conditions |
| --- | --- |
| NO | No ABSR support; manual input, i.e., the baseline scenario.<br>The controllers manually enter the spoken commands via mouse into the controller's HMI of the TowerPad™.<br>Simulation pilots control taxi traffic at their working positions by manual input via mouse and keyboard. This corresponds to the established mode of operation without ABSR. |
| JC | Use of ABSR support just for controllers, i.e., automatic command recognition support for controllers plus manual correction, if ABSR fails.<br>Commands spoken by the controller are processed by the ABSR system and transmitted to the controller's working position, where they are automatically entered for the controller. The controller receives feedback on the recognized commands via the controller's HMI and can correct errors via a mouse.<br>No support by ABSR for the simulation pilots. |
| JP | Use of ABSR support just for simulation pilots, i.e., automatic command recognition and control for pilots.<br>The commands spoken by the controllers are processed by the ABSR system, transmitted to the working position of the responsible simulation pilot, and automatically executed as control commands for the simulation pilot. The simulation pilot receives feedback on the recognized commands via the simulation pilot's HMI and can correct errors via a mouse and keyboard.<br>No support by ABSR for the controllers. |
| CP | Use of ABSR support for both controllers and simulation pilots, as described individually for JC and JP conditions. |

#### 4.2.1. Validation Hypotheses

The following hypotheses were tested during the final validation trials:

**H1.** *(H-C-less_input): Automatic documentation (conditions JC and CP) reduces the total number of manual inputs to guide taxiing traffic at the controller's working position compared to full manual input (conditions NO and JP).*

**H2.** *(H-P-less_input): Automatic command recognition for simulation pilots (conditions JP and CP) reduces the total number of manual inputs to guide the taxiing traffic of simulation pilots compared to full manual input (conditions NO and JC).*

**H3.** *(H-C-more_cog_res): Automatic documentation (conditions JC and CP) increases the controller's free cognitive resources compared to full manual input (conditions JP and NO).*

**H4.** *(H-C-less_workload): Automatic documentation (conditions JC and CP) reduces the workload of the controller compared to full manual input (conditions JP and NO).*

**H5.** *(H-C-sit_aw_ok): Automatic documentation (conditions JC and CP) does not limit the controller's situational awareness compared to full manual input (conditions JP and NO).*

**H6.** *(H-C-conf): The controller's confidence in command entry automation (conditions JC and CP) is above average.*

**H7.** *(H-P-conf): The simulation pilot's confidence in command entry automation (conditions JP and CP) is above average.*

**H8.** *(H-E-CmdRR): The command extraction rate (JC, JP, and CP) in the apron environment is comparable to the quality already achieved in the approach domain by ABSR (command extraction rate for simulation-relevant commands >90%).*

**H9.** *(H-E-CmdER): The command extraction error rate (conditions JC, JP, and CP) in the apron environment is comparable to the quality already achieved in the approach domain by ABSR (command extraction error rate for simulation-relevant commands <5%).*

**H10.** *(H-E-CsgRR): The callsign extraction rate (conditions JC, JP, and CP) in the apron environment is comparable to the quality already achieved in the approach domain by ABSR (>97%).*

**H11.** *(H-E-CsgER): The callsign extraction error rate (conditions JC, JP, and CP) in the apron environment is comparable to the quality already achieved in the approach domain by ABSR* ***H11.*** *(callsign extraction error rate <2%.)*

#### 4.2.2. Independent Variables

The independent variables (IV) of the final validation trials were as follows:

1. (IV-Input): Documentation on the controller's HMI by ABSR vs. manual input (JC and CP vs. JP and NO).
2. (IV-Control): Control of the simulation by ABSR for the controller's utterances vs. full manual input (JP and CP vs. JC and NO).

#### 4.2.3. Dependent Variables

The dependent variables of the final validation trials are listed in Appendix A. The respective results are each compared between the different operational conditions within a scenario.

### *4.3. Execution of the Final Validation Trials*

The final validation trials took place from 27th of June to 1st of July 2022 in the apron simulator in Frankfurt. The number of simultaneously active users was three controllers and three simulation pilots. For the final trials, 14 controllers were recruited who had enough experience with the A-SMGCS system (see Figure 8). On each day, a new team of controllers was on site (one controller participated twice). Half of the participants already had their first experience with the system at one of the many pre-simulations. The other half had their first contact with the ABSR system during the final trials.

Two different traffic scenarios were prepared for the final validation trial: one for runway operating direction (OD) 25 and one for OD 07. The simulation scenarios generated from these were 30 min long each. Table 2 shows the number of aircraft movements in total and the projected number of aircraft at each of the three working areas: East, Center, and West.

**Table 2.** Number of (#) aircraft in certain areas per operating direction.

| Traffic Scenario | # Aircraft | # Arriving Aircraft | # Departing Aircraft | # Expected Aircraft East | # Expected Aircraft Center | # Expected Aircraft West |
|---|---|---|---|---|---|---|
| OD25 | 106 | 46 | 60 | 59 | 61 | 63 |
| OD07 | 106 | 46 | 60 | 57 | 45 | 59 |

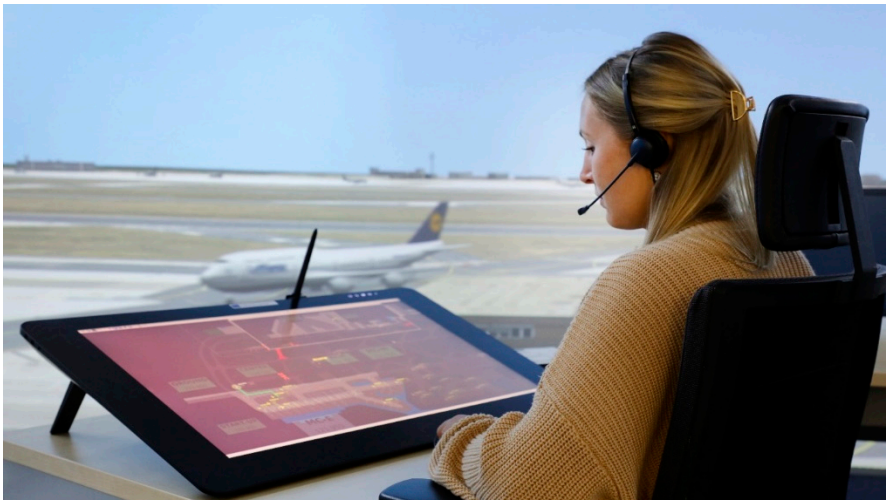

**Figure 8.** Photo (© Fraport) of final validation trial setup with A-SMGCS in front of the controller.

In order to generate a heavy workload, the amount of traffic in the scenarios were increased compared to usual traffic at Frankfurt Airport so that the controllers were as busy as possible all the time. The heaviest workload with respect to radio frequency usage and number of commands was expected at the Center position, followed by the East working position. At the West working position, the load was expected to be lower, even if the numbers in the Table 2 suggest otherwise. This is partially due to the type of movement (pushback-aircraft must be pushed from the parking position with the tug, etc.) and the size of the area. In the West, there were significantly fewer pushbacks, because there were less nose-in positions (so most aircraft could leave the parking stand forward under their own power). The number of commands and utterances per position for all runs are shown in Table 3.

**Table 3.** Number of commands (# Cmds) and utterances (# Utterances) at each of the three working positions.

| | # Cmds | % of All | # Utterances | % of All |
|---|---|---|---|---|
| Center | 5858 | 38% | 2437 | 38% |
| West | 4376 | 28% | 1654 | 26% |
| East | 5235 | 34% | 2262 | 36% |

The realistic maximum traffic volume in real operations was 106 per 60 min for 2019. This amount was used in the simulation trials for half an hour. This increase compensated for the fact that there were no tows on the apron and other secondary activities that would otherwise occur in reality and that could not all be represented in the simulator.

Each simulation day began with a briefing of the controllers and simulation pilots involved. In this briefing, the controllers and simulation pilots were educated on the concept of ABSR and its interaction with the controller input interface and the simulation pilot's working station. They were explained how to make manual entries in the systems and when these are required (when no ABSR support is active for the respective station, or a correction is necessary).

In addition, the controllers and simulation pilots were informed about the schedule, and the questionnaires were introduced. Afterwards, training for controllers and simulation pilots took place, in which all operational conditions were explained and tried out. During the training run, the controllers also exercised the secondary task for measuring mental load after a short introduction on how to perform it. This secondary task is discussed in more detail in Section 4.4.

The teams of three controllers and three simulation pilots remained at their working positions throughout the different simulation runs and operational conditions (OC). On each day, there were different runs for each of the two operating directions OD07 and OD25. The teams were always the same for the same OD.

When evaluating the influence of ABSR support on the work of the controller, the ABSR system was always active for the simulation pilots, i.e., only altered from on to off and vice versa for the controllers. In addition, two simulation runs were carried out in OD25, in which the ABSR system was always active on the controller's side and a change from on to off and vice versa for the simulation pilots. The influence of the ABSR system on the simulation pilot's activity was analyzed, too. Thus, six runs were performed per day. After each run, the controllers filled out a questionnaire. At the end of the day, an additional questionnaire was filled out, and the impressions, comments, and hints of the controllers and simulation pilots were recorded in a non-formal debriefing session with all participants.

In the runs, in which the ABSR system was alternately on or off for the controller, the secondary task was performed by the controllers. The task started 10 min after the start of the run and stopped 10 min later. This was performed simultaneously for all three working positions. The simulation runs with the different operational conditions and simulation scenarios were determined as follows in Table 4.

**Table 4.** Simulation runs with different operational conditions and simulation scenarios.

| Simulation Run Name with OD | Operational Conditions and Traffic Scenario |
|---|---|
| T | Training of fully manual input and ABSR-supported input with manual corrections at controllers' and simulation pilots' working positions. |
| CP25 | ABSR support for controllers and simulation pilots. |
| JC25 | ABSR support just for controllers. |
| JP25 | ABSR support just for simulation pilots. |
| NO25 | No ABSR support. |
| CP07 | ABSR support for controllers and simulation pilots. |
| JP07 | ABSR support just for simulation pilots. |

Table 5 below shows the order of the training and experimental runs for each controller-team, consisting of three persons. The order of the runs was changed each day to reduce (in the mean of the evaluation) learning effects that may occur during the day.

**Table 5.** Simulation runs per controller team and day.

| Team 1<br>Day 1 | Team 2<br>Day 2 | Team 3<br>Day 3 | Team 4<br>Day 4 | Team 5<br>Day 5 |
|---|---|---|---|---|
| T | T | T | T | T |
| NO25 | CP25 | JP25 | JP25 | JP07 |
| JC25 | JP25 | NO25 | CP25 | CP07 |
| JP25 | JC25 | CP25 | JP07 | CP25 |
| CP25 | NO25 | JC25 | CP07 | JP25 |
| JP07 | CP07 | JP07 | NO25 | JC25 |
| CP07 | JP07 | CP07 | JC25 | NO25 |

### *4.4. Objective Workload Measurement by a Secondary Task*

The questionnaires reflect the subjective experiences of the controllers, which one might argue to be the most important measure for most operationally deployed systems.

Nevertheless, we wanted to obtain more objective data that would confirm or reject our hypothesis that the proposed system reduces workload.

To measure mental load, we used a secondary task that required similar skills to the main task (controlling traffic), namely mental focus, English language proficiency, color recognition, and quick orientation on the user interface, and yet that was simple enough to be performed in parallel with the main task.

In the pre-simulations, subjects were asked to sort decks of playing cards as a secondary task to measure free mental capacity and were then made to answer questions about missing cards ("which 1–4 cards were missing?"), as was described in [3]. However, the use of this task required too much manual effort and, therefore, ran the risk of introducing errors in data recording and execution, so we chose a largely automated approach for the final validation trials using the application described below. This greatly reduced the physical and mental workload of the simulation support team and the susceptibility to errors.

For the secondary task, 10 min after the start of each simulation run, each controller (and additionally once in parallel with the simulation pilots) was asked to complete as many Stroop tasks [43] as possible in the following 10 min in addition to their main task. For this purpose, a tablet PC (6x Samsung A8) was provided to the controllers that ran an application for executing consecutive Stroop tasks [44]. The application recorded the execution time and duration of each task as well as its correctness. A high number of correctly executed Stroop tasks in the application suggests an available mental capacity that is not needed for the main task.

The atomic Stroop task is the following: when the start button is pressed, a word for a color is displayed, but in a different color to the color that the word stands for. The task for the user is to select the right button with the color word that matches the display color from a set of seven buttons, all labelled with a color word in black. The order of these buttons changes in a pseudo-random way at each repetition of the task. In Figure 9, the color word "ORANGE" is displayed in blue, so the button to press is the one labelled "BLUE".

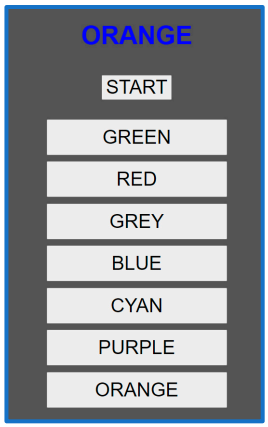


**Figure 9.** Example Stroop task.

## 5. Validation Results

### *5.1. Speech Recognition and Understanding Performance*

Section 5.1.1 focuses on speech recognition, and Section 5.1.2 focuses on speech understanding performance results.

#### 5.1.1. Speech-to-Text Accuracy (Speech Recognition)

A first indication for the quality of the ABSR system is provided by the so-called word error rate (WER) of the S2T component. The WER is calculated based on the Levenshtein distance [42] between the word sequence recognized by the S2T component and the actual spoken word sequence (gold transcription). This involves counting, in the recognized

word sequence, how many words of the actual word sequence have been substituted (S), deleted (D), or additionally inserted (I). All three components are then added and divided by the number of words of the actual word sequence (N). Table 6 shows the WER of the developed S2T component in the final validation trials based on the verbal utterances of 14 apron controllers.

**Table 6.** Word error rate of recognized word sequences from the S2T component.

| Recognition Mode | WER | |
|---|---|---|
| Offline (PTT signal simulated) | 3.1% | Male: 3.3%<br>Female: 2.6% |
| Online (voice activity detection) | 5.0% | Male: 5.5%<br>Female: 3.7% |

WER was evaluated for two different modes. Online recognition measures what recognition performance the S2T component achieved during the final validation trials in summer 2022. This means that these results contain a certain number of errors that are not induced by the S2T component but by voice activity detection (VAD), due to the missing PTT signal. In order to determine how large the influence of VAD is and what improvement can be expected by accessing the PTT signal, the offline recognition after summer 2022 was used to subsequently evaluate what the system would have recognized if the audio stream had been perfectly split by PTT. It can be seen that offline recognition again brings a significant improvement over online recognition, with a WER of 3.1% compared to 5.0%.

It is also interesting to observe that the average WER of female apron controllers (2.6% and 3.7%, respectively) was better than those of male apron controllers (3.3% and 5.5%, respectively). On the other hand, out of the total 14 apron controllers, only four were female. Performing unpaired *t*-tests with the 24 runs with female apron controllers versus the 62 runs of male controllers provides very statistically significant results, with a *p*-value of 0.02%.

The question of what WER is good enough for the intended purpose often arises. This question cannot be answered in a general way, because in the end, it is irrelevant how many words are recognized correctly. What is important is the ability of the system to extract the meaning behind the recognized words and finally to implement it appropriately in the application. Some errors on the word level can change the meaning of an utterance, while others have no influence at all. Therefore, it is not possible to define a general threshold for the WER, but a low WER allows conclusions on the quality of the implemented ABSR system.

#### 5.1.2. Text-to-Concept Accuracy (Speech Understanding)

The performance of speech understanding is evaluated by comparing the commands automatically extracted by the system with the correct commands manually created and verified by human experts (gold annotations). The evaluation is based on three metrics: command recognition rate, command error rate, and command rejection rate. The command recognition rate is defined as the number of correctly recognized commands divided by the total number of commands actually given. A command is considered correctly recognized if and only if all elements of a command such as command type, callsign, value, unit, qualifier, condition, etc., as defined in the ontology, are correctly recognized. Command error rate is the percentage of incorrectly extracted commands divided by the total number of commands actually given. Command rejection rate is the percentage of actual commands given that were not extracted at all or were rejected by the system for some reason. Table 7 below shows the metrics defined above with an example. The example also illustrates that the sum of the recognition, error, and rejection rates can exceed 100%.

**Table 7.** Example for speech understanding metrics.

| Actual Commands | Recognized Commands | Contribution to Metric |
|---|---|---|
| DLH695 TURN RIGHT | DLH695 TURN LEFT | ⊖ |
| DLH695 TAXI VIA N10 N | DLH695 TAXI VIA N10 N | ⊕ |
| | DLH695 TAXI TO V162 | ⊖ |
| AUA1F PUSHBACK | AUA1F NO_CONCEPT | ○ |
| CCA644 NO_CONCEPT | CCA644 NO_CONCEPT | ⊕ |
| Recognition Rate (⊕) = 2/4 = 50% | Error Rate (⊖) = 2/4 = 50% | Rejection Rate (○) = 1/4 = 25% |

In a similar manner to the extraction rates for commands, separately, the extraction rates for callsigns are determined. Again, there is a callsign recognition rate, error rate, and rejection rate. For each utterance, each callsign is considered only once, unless several different callsigns are extracted from the same utterance ("break break" utterances). Therefore, in the above example from Table 7, three callsigns are considered. Detailed information on the defined metrics can be found in [45]. Table 8 illustrates the performance of speech understanding based on the above-explained metrics, i.e., it contains the number of radio telephony utterances and commands as well as the recognition, error, and rejection rates for full commands and callsigns.

**Table 8.** Recognition (RecR), error (ErrR), and rejection (RejR) rate for commands (Cmds) and callsigns (Csgn) in [%] and absolute numbers of utterances (# Utterances) and commands (# Commands).

| Recognition Mode | # Utterances | # Commands | Cmds [%] | | | Csgn [%] | | |
|---|---|---|---|---|---|---|---|---|
| | | | RecR | ErrR | RejR | RecR | ErrR | RejR |
| Offline (PTT signal simulated) | 5495 | 13,251 | 91.8 | 3.2 | 5.4 | 97.4 | 1.3 | 1.3 |
| Online (voice activity detection) | 5432 | 13,168 | 88.7 | 4.3 | 7.5 | 95.2 | 2.3 | 2.4 |
| Offline (no callsign prediction used) | | | 76.3 | 10.5 | 13.7 | 81.1 | 9.6 | 9.3 |
| Delta to context | | | 15.5 | −7.3 | −8.3 | 16.3 | −8.3 | −8.0 |

"Delta to context" is the difference of row 2 "Offline (PTT…)" and row 4 "(Offline no callsign…)".

Recognition rates of 91.8% and 88.7% are obtained when speech understanding is applied offline (simulated PTT) and online (VAD), respectively. The improvement in the speech understanding result for offline recognition comes from the better word-level recognition and the fact that the offline data does not include radio transmissions that were incorrectly split by VAD, allowing for a better interpretation of the content. Similarly, the recognition of aircraft callsigns in offline recognition is also better than in online recognition, with recognition rates of 97.4% and 95.2%, respectively. The last two rows of Table 8 show the influence of the predicted callsigns on the recognition performance of ABSR. With context information available, the recognition rate increases by 15.5% overall and 16.3% on the callsign level.

Table 9 shows the rates of offline recognition for different command types. The table lists only the most common or important command types that are relevant to the application.

Thus, speech understanding has error rates below 4% and recognition rates in the range of roughly 87% to 98%, depending on the command type with the exception of the GIVE_WAY command. The reason for the worse results of the GIVE_WAY command extraction, marked in red in Table 9, is its very complex nature, i.e., it can be given/uttered in many different ways, not all of which were modeled so far.

**Table 9.** Recognition (RecR), error (ErrR), and rejection (RejR) rate for specific command types in [%] and absolute number of commands (# Cmds) of that type.

| Command Type | # Cmds | RecR | ErrR | RejR |
|---|---|---|---|---|
| TAXI VIA | 2922 | 86.9 | 3.9 | 9.1 |
| HOLD_SHORT | 1837 | 89.3 | 0.8 | 9.9 |
| TAXI TO | 1406 | 89.0 | 1.1 | 9.9 |
| CONTACT_FREQUENCY | 1387 | 95.7 | 0.7 | 3.6 |
| CONTINUE TAXI | 1102 | 95.4 | 0.0 | 4.6 |
| GIVE_WAY | 728 | 69.6 | 10.2 | 20.3 |
| CONTACT | 672 | 98.4 | 0.3 | 1.3 |
| PUSHBACK | 663 | 92.3 | 1.2 | 6.5 |
| TURN | 359 | 89.2 | 3.9 | 6.9 |
| HOLD_POSITION | 223 | 93.4 | 0.0 | 6.6 |

Worst ErrR overall marked in red.

### *5.2. Interaction Count*

To determine the amount of manual HMI interactions needed at the A-SMGCS with and without ABSR support, we recorded the HMI interactions at each position and per simulation run, counted them, and categorized them into 48 different task types, such as "edit route", "clear pushback", and "select label". Expectations were, of course, that the number of interactions would be significantly lower when ABSR was available. However, it was apparent from the pre-simulations that the controllers would not make all the required inputs without ABSR support, nor would they correct every error made by the ABSR system, because not performing an input has no direct consequences for the controller as long as the simulation pilot still follows the voice instructions, because the pilots control the simulation.

Therefore, the numbers of the simulation pilots are more meaningful, since here, any omitted input or correction leads to a delay or incorrect behavior in the simulator. On the other hand, not all interactions of the simulation pilots can be replaced by the ABSR input, because the pilot initiates the communication with the controller and needs information for this, which is only available when selecting the aircraft symbol via a mouse click.

Therefore, even for a perfect ABSR, the total number of interactions can never be zero and without ABSR, the number of interactions is higher for the simulation pilots, and the reduction in interactions for the simulation pilots is also not as extreme as for the controllers. Figure 10 shows the remaining portion of manual actions needed for the controllers and simulation pilots when being supported by ABSR for the most frequent interactions. A strong reduction of workload is apparent for both.

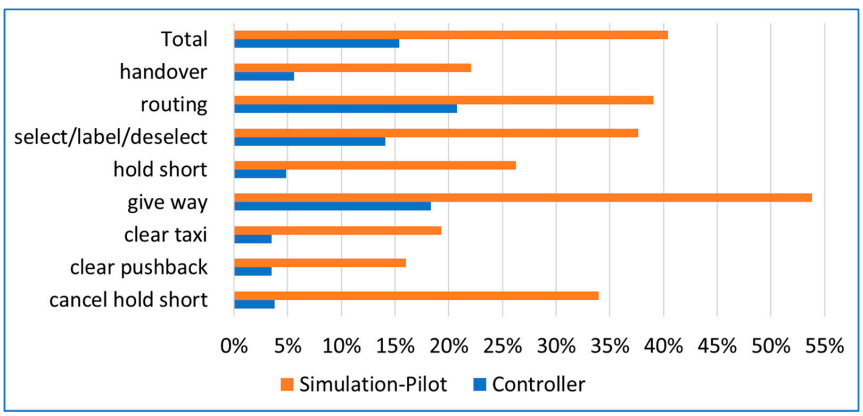


**Figure 10.** Portion of remaining manual interactions with the HMI (by type) of the controllers and simulation pilots with ABSR support compared to runs without ABSR support.

### *5.3. Workload, NASA TLX*

The NASA Task Load Index (TLX) has been used for decades in different variants to assess perceived workload in six different aspects [46]. We used a simple unweighted questionnaire procedure in which a mark between 0 (very low) and 20 (very high) is to be entered for each aspect. "The Task", here, means the task of the apron control in the simulator operating the A-SMGCS.

This questionnaire was completed by the users at the controllers' working positions after each simulation run. The scores were aggregated by position, OD, and by use of the ABSR (or not). The six questions are as follows:

- Mental Demand: How mentally demanding was the task?
- Physical Demand: How physically demanding was the task?
- Temporal Demand: How hurried or rushed was the pace of the task?
- Performance: How successful were you in accomplishing what you were asked to do?
- Effort: How hard did you have to work to accomplish your level of performance?
- Frustration: How insecure, discouraged, irritated, stressed, and annoyed were you?

The question about the subjects' own performance (see list above) requires a reversal of the scale values: "how successful?" suggests a high value if the subjects were satisfied with their performance. This is also how the controllers expressed themselves in the feedback rounds, but this was not consistently evident in the questionnaires; in some cases, conspicuously low values were given here, although the values for the other aspects were also very low. Unfortunately, we have to assume that not all controllers understood this question correctly or answered it as expected, and we therefore excluded the performance aspect from the evaluation.

Table 10 below shows the average values by working position and overall, separately for the baseline runs (without ABSR) and the solution runs (with ABSR). Columns "$\alpha$" show the statistical significance of a *t*-test.

**Table 10.** NASA TLX questionnaire results of the controllers on perceived workload.

| | West | | | Center | | | East | | | All | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Workload** | **Base** | **Sol** | $\alpha$ | **Base** | **Sol** | $\alpha$ | **Base** | **Sol** | $\alpha$ | **Base** | **Sol** | $\alpha$ |
| Mental Demand [MD] | 7.9 | 6.4 | 14.5% | 14.4 | 12.4 | 1.1% | 14.7 | 13.8 | 10.3% | 12.3 | 10.8 | 1.3% |
| Physical Demand [PD] | 7.6 | 3.3 | 0.4% | 9.4 | 5.6 | 0.3% | 10.6 | 7.9 | 2.5% | 9.2 | 5.6 | $3 \times 10^{-4}$ |
| Temporal Demand [TD] | 8.1 | 5.9 | 4.4% | 12.8 | 10.7 | 2.4% | 14.1 | 13.1 | 6.2% | 11.7 | 9.9 | 0.3% |
| Effort [EF] | 8.6 | 5.5 | 2.1% | 12.6 | 10.5 | 1.3% | 14.5 | 12.4 | 1.6% | 11.9 | 9.5 | $1 \times 10^{-3}$ |
| Frustration [FR] | 4.0 | 3.4 | 23.9% | 6.8 | 3.5 | 1.0% | 7.2 | 5.0 | 6.1% | 6.0 | 4.0 | 0.2% |
| ALL | 7.2 | 4.9 | 2.5% | 11.2 | 8.5 | 0.2% | 12.2 | 10.4 | 1.4% | 10.2 | 8.0 | $6 \times 10^{-4}$ |

Minimal $\alpha$ values, shaded in green for $0\% \leq \alpha < 5\%$, in light green for $5\% \leq \alpha < 10\%$, and in yellow for the rest ($|\alpha| \geq 10\%$). As the numbers show no evidence that results with ASRU support are worse, no further color coding is needed.

In general, the workload on the working position "West" was significantly lower than on the other two, while it was estimated to be slightly higher on East than on Center. At OD25, the workload is slightly higher at West and at Center, while the workload for East is estimated to be highest at OD07 (not shown in the table). On average, the workload is slightly higher for OD07, and only with regard to the time aspect is OD25 experienced as somewhat more stressful. Thus, although the working position makes a big difference, the OD does not have a noticeable effect on the average values for all positions.

Especially for physical demand, the value even decreases by almost four points. Significance tests (paired *t*-tests) on the data prove that the differences between with and without ABSR are not random. On the West position, the results for mental demand, and frustration does not decrease in a statistically significant way. The overall results with

respect to workload reduction were very, very statistically significant. We obtained an alpha ($p$-value) of 0.06%, i.e., if we would have repeated the experiments with all the 15 participants 1000 times again, only in six cases could we expect that the workload without ABSR support is less than that with ABSR support.

### *5.4. Evaluation of Stroop Tests as Secondary Task*

The results of the secondary task point in the expected direction: at the Center and West positions, subjects were able to perform significantly more tasks correctly in parallel with their work when the ABSR support was active, see Figure 11.

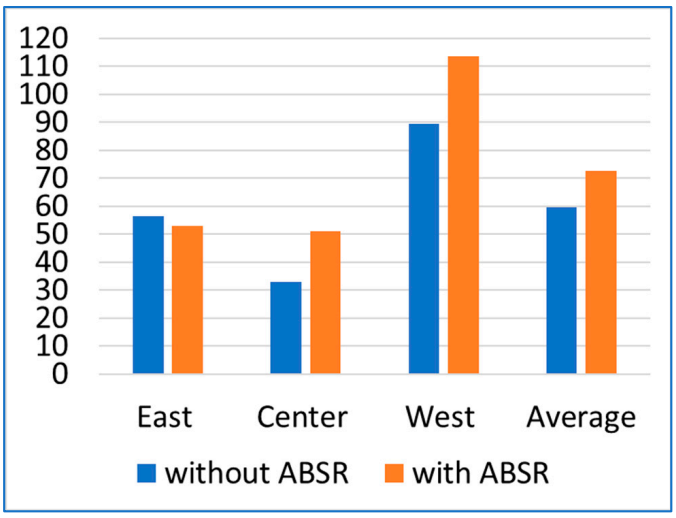


**Figure 11.** Average number of correct Stroop tests per working position with and without ABSR.

From a statistical point of view, the variance at the East working position was too high to make a reliable statement for this position. At Center and West, the figures support the hypothesis, but strictly speaking, this statement is still outside usually required statistical significance due to the relatively small total number of experiments. The qualitative observations made during the simulation runs support our decision to use the application and indicate that the secondary task used here is an objective measure of the cognitive capacity available:

- "If the Stroop tasks are done while R/T [radio telephony] must be done, selecting the correct button takes longer."
- "More complicated routes increase the error rate [in Stroop tasks]."

### *5.5. Situational Awareness, Shape-SASHA*

The questionnaire Situational Awareness for Shape (SASHA) [47] was used to assess the controllers' situational awareness during the simulation runs. The test persons marked their assessment of the aspects on a Likert scale between 0 "never" and 6 "always". The negatively formulated statements (marked with an "*" below) have been inverted for later evaluation, i.e., the averages are calculated as 6 minus the raw average value. The statements in detail were as follows:

- In the previous working period(s), ...
  - I was ahead of the traffic.
  - I started to focus on a single problem or a specific area of the sector.*
  - There was the risk of forgetting something important [...].*
  - I was able to plan and to organize my work as I wanted.
  - I was surprised by an event I did not expect [...].*
  - I had to search for an item of information.*

The subjects filled out this questionnaire after each simulation run. After evaluation, the situational awareness of the controllers is generally found to be good with and without ABSR (see Table 11). The average values over all simulation runs, positions, and aspects are above 4. The most important message is that with ABSR support, situational awareness increases on average over all aspects and at each position.

**Table 11.** Results of SASHA questionnaire with and without ABSR support.

| Situational Awareness | without ABSR (W/C/E) | with ABSR (W/C/E) |
|---|---|---|
| ahead of traffic | 4.4 (5.2/4.0/4.0) | 4.7 (5.4/4.2/4.6) |
| focus single problem | 4.3 (4.9/3.9/4.2) | 4.3 (4.2/4.3/4.5) |
| risk of forgetting | 3.9 (4.8/3.5/3.4) | 4.4 (4.9/4.1/4.3) |
| able to plan | 4.1 (4.7/3.9/3.6) | 4.6 (4.9/4.6/4.3) |
| surprised by event | 4.5 (5.1/4.5/4.0) | 4.9 (5.4/4.7/4.7) |
| search information | 3.9 (4.1/3.8/3.9) | 4.4 (4.5/4.7/4.0) |
| ALL | 4.2 (4.8/3.9/3.8) | 4.6 (4.9/4.4/4.4) |

Further analysis of the values not shown in the above table reveals some differences in situational awareness (SA) related to the working environment. The working position area has an influence on SA, i.e., at the West position, the value was 4.8, while East (4.1) and Center (4.2) have lower values. The OD makes a very small difference, with 4.3 for OD07 and 4.4 for OD25, respectively. However, SA is significantly lower (one whole scale point) at the Center position for OD25 and at the East position for OD07, regardless of the use of ABSR. These two working positions gain half a point with ABSR support, but this is not as clearly reflected at the West position. These results fit the NASA TLX results: where workload is lower, situational awareness is higher.

### *5.6. Confidence in Automation, Shape-SATI*

To elicit trust in the automatic entry of commands into the controllers' or simulation pilots' HMI, we used the SHAPE Automation Trust Index questionnaire, SATI. We had each participant complete this questionnaire once at the end of the simulation day, with the request that they focus on the effects of the ABSR system. This allowed us to evaluate 15 questionnaires from the controllers and 6 from the pilots. The items that could be answered on a 7-point Likert scale from 0 "never" to 6 "always" in detail were as follows:

- In the previous working period(s), I felt that . . .
  - ○ The system was useful.
  - ○ The system was reliable.
  - ○ The system worked accurately.
  - ○ The system was understandable.
  - ○ The system worked robustly (in difficult situations, with invalid inputs, etc.).
  - ○ I was confident when working with the system.

A distinction whether the ABSR system or other automation features of the not completely familiar system triggered trust or distrust could probably not be made completely by the test persons. Hence, the results must be considered under this restriction. The overall impression turned out to be very positive, as shown in Table 12.

The controllers consider the system almost always useful. Regarding accuracy and robustness, the confidence is lowest but still high (>4). The simulation pilots are slightly more skeptical, but overall trust in the system is well above average.

**Table 12.** Results of SATI questionnaire.

| Automation Trust | Controllers (15) | Simulation Pilots (6) |
|---|---|---|
| useful | 5.1 | 4.6 |
| reliable | 4.5 | 4.2 |
| accurate | 4.3 | 4.2 |
| understandable | 4.9 | 5.3 |
| robust | 4.2 | 4.2 |
| confident | 4.8 | 4.3 |
| ALL | 4.6 | 4.5 |

### *5.7. Results with Respect to Safety*

#### 5.7.1. Software Failure Modes, Effects, and Criticality Analysis (SFMECA)

The risk analysis based on the SFMECA had the result that no error case would lead to an increased or unacceptable risk, so that no classification into good and bad recognitions is needed, as mentioned in Section 2.3, although our implementation of speech understanding provides this information. This result was not expected. However, it can be explained as follows: The apron control is not responsible for the runways, i.e., areas are excluded where wrong decisions have particularly severe effects and where the possibility of detecting errors quickly is reduced (due to higher speeds). There was also no indirect risk of causing distractions through false alarms and thus endangering situational awareness, since there were no automatic alarm functions available in the project (with which the controllers were familiar). Based on the safety analysis, it was therefore possible to make the decision to implement all commands directly in the A-SMGCS without exceptions (those that are not identified as nonsensical based on plausibility checks).

For the most critical command "Handover", it was decided to always offer an undo function. A mistakenly executed handover of a flight to another working position would cause all subsequent commands to be discarded: the aircraft would be assigned to another working position and therefore be unavailable for incoming commands at the actual working position. However, the error is very easy to detect with the implemented visualization, and we offered a one-click solution to undo it.

In addition to considering AI ABSR errors, this project also discussed and considered the issues of safety when introducing automation. In addition to the direct effects of automation errors, increased automation can affect safety in the following ways:

- Indirect impact due to automation errors (too many disruptive errors, either due to a lack of recognition or incorrect recognition);
- Lack of visibility of the automation result (a loss of "situational awareness");
- Lack of flexibility (no possibility of correction or override by the user and therefore a loss of control);
- Overconfidence/complacency.

The approaches in this project for addressing these risks were the following:

1. Achieve sufficient recognition rates and sufficiently low recognition error rates to prevent potential overload from occurring in the first place.
2. Make the results visible enough for users to retain situational awareness at all times.
3. Allow human operators to make corrections to automation errors in order to remain in control.
4. Assessments of risk by overconfidence through safety considerations: what can happen if automation errors are not corrected?

This was validated in multiple ways: (1) indirectly, by selecting particularly challenging simulation scenarios that go beyond the usual in terms of traffic density, by evaluating the required number of interactions with the user interface, and by measuring cognitive

load using secondary tasks; and (2) directly through test subjects filling out standardized questionnaires on situational awareness and trust in automation.

#### 5.7.2. Feedback from the Test Subjects on Safety

From simulations at the beginning of the project, in which significantly more errors happened, and significantly fewer commands were available for automatic execution, the following feedback was obtained:

- *Since the speech recognizer still makes mistakes and you have to check if everything is correct whenever you are spoken to, you are less free in your timing. One also expects that, e.g., the callsign is highlighted. If that doesn't happen, you're wondering why it didn't work.*

The feedback on safety became gradually less negative as the project progressed, and the number of errors decreased. At the beginning, there were definitely impairments of a smooth workflow, because the controllers had to wait for the implementation or were inclined to always check the correctness. When most commands worked and the error rate dropped significantly, there were no more comments suggesting a negative impact or reduced safety. This confirms the work on the safety of the overall validation system and the analysis from the safety assessment.

After the simulation runs, subjects were always questioned about safety. The following responses (translated analogously by the authors) are representative of the sentiments:

- *You could always see if there were errors or not.*
- *The delay is fine. You can already talk to the next pilot or you get the indication during the readback. That's sufficient.*
- *The errors were very few. They couldn't put us in critical situations.*
- *Here, the aircraft are controlled very directly because the simulator directly implements commands [with voice recognition enabled] [including errors]. A pilot would not do that. That's why it [emerging situations] would be less critical in real life.*
- *If something takes too long, you leave it out.—If the pilot executes it correctly, it's okay.—If incorrectly detected, the worst thing that can happen is false alarms.*

#### 5.7.3. Summary of All Feedback Collected

Feedback was consistently positive toward the solution with ABSR support. The controllers were mostly surprised that the system worked so well, even though it is still in a research stage. It was emphasized that it made no qualitative difference to the ABSR system (1) whether the controller spoke quickly or slowly, or (2) whether the controller strictly adhered to International Civil Aviation Organization (ICAO) phraseology [16] in his/her speech utterances or deviated from it to a greater or lesser extent, caused by increased traffic density and high radio frequency use. The system was very well received because it did not require controllers to change. The controllers could simply speak as they were accustomed and still the correct action occurred in most cases. The controllers said that it could leave more time to keep an eye on traffic instead of staring at the display.

It was also noted that with ABSR support, the controllers sometimes instruct different taxi routes than when they have to input the route manually: If a route is pre-selected by the system, then it is easier to follow it than to change it manually. But if the controllers can simply use speech to change the route instead of having to enter it manually, then they are more likely to change the route, e.g., to shorten the aircraft's taxiing time.

The controllers as well as the simulation pilots indicated that the workload decreases significantly with ABSR support. The best feedback was for the working position West. Here, almost everything was correctly recognized for everybody. Recognition was also good for the East and Center positions, but there were also minor misrecognitions.

There were hardly any critical voices. Rather, there were suggestions on how to make it even better, e.g., that the recognition should be faster and could be better, so that there would be even fewer false recognitions. The command types "Hold Abeam" and "Pushback Abeam" were, e.g., not implemented within the resources of the STARFiSH project. Over the days, the feedback from the controllers involved was qualitatively repetitive, and

so it became apparent that the different controllers had the same good experiences with the system.

### *5.8. Results with Respect to Validation Hypotheses*

In Section 4.2.1, we formulated the Hypotheses H1 to H11. The results with respect to validation of the hypotheses and falsification were presented in the above sections. This subsection summarizes the results with respect to each hypothesis.

#### 5.8.1. Hypotheses with Respect to "Number of Manual Inputs"

The results with respect to these hypotheses are presented in Section 5.2 in Figure 10. The number of inputs from the simulation pilots (dependent variable, *DV-Input-H-P-less_input*) is reduced by a factor of 2.5, and the number of manual inputs of the apron controllers (dependent variable, *DV-Input-H-C-less_input*) is even reduced by a factor of more than 6. Therefore, we mark the following two hypotheses as validated.

**H1.** *(H-C-less_input): Automatic documentation (conditions JC and CP) reduces the total number of manual inputs to guide taxiing traffic at the controllers' working position compared to full manual input (conditions NO and JP).* **Validated**

**H2.** *(H-P-less_input): Automatic command recognition for the simulation pilots (conditions JP and CP) reduces the total number of manual inputs to guide the taxiing traffic of the simulation pilots compared to full manual input (conditions NO and JC).* **Validated**

#### 5.8.2. Hypothesis with Respect to "Free Cognitive Resources of Apron Controller"

The results with respect to this hypothesis are presented in Section 5.4 in Figure 11. The number of correct Stroop tasks increased for the West and Center positions and did not decrease for the East position. Therefore, we mark the following hypothesis as partially validated.

**H3.** *(H-C-more_cog_res): Automatic documentation (conditions JC and CP) increases the controller's free cognitive resources compared to full manual input (conditions JP and NO).* **Partially Validated**

#### 5.8.3. Hypothesis with Respect to "Apron Controller Workload Reduction"

The results with respect to this hypothesis are presented in Section 5.3 in Table 10. The workload reduced by 2.2 scale units on the 20-unit NASA TLX scale on average. Therefore, we mark the following hypothesis as validated.

**H4.** *(H-C-less_workload): Automatic documentation (conditions JC and CP) reduces the workload of the controller compared to full manual input (conditions JP and NO).* **Validated**

#### 5.8.4. Hypothesis with Respect to "Apron Controller's Situational Awareness"

The results with respect to this hypothesis are presented in Section 5.5 in Table 11. The situational awareness over all three positions and over both operating directions increased from 4.2 to 4.6 (maximum value of 6.0). The lowest effect was measured for the West position with an increase of 0.1 unit points. However, situational awareness was already high without ABSR support at this position (4.8). Therefore, we mark the following hypothesis as validated.

**H5.** *(H-C-sit_aw_ok): Automatic documentation (conditions JC and CP) does not limit the controller's situational awareness compared to full manual input (conditions JP and NO).* **Validated**

#### 5.8.5. Hypotheses with Respect to "Apron Controller's Confidence"

The results with respect to this hypothesis are presented in Section 5.6 in Table 12. The average value for the apron controllers was 4.6 and that for simulation pilots was 4.5. These values are above the average of 3.0. In addition, the lowest individual value for both

(4.2) is far beyond the average of 3.0. Therefore, we mark the following hypotheses both as validated.

**H6.** *(H-C-conf): Controller confidence in command entry automation (conditions JC and CP) is above average.* **Validated**

**H7.** *(H-P-conf): Simulation pilot's confidence in command entry automation (conditions JP and CP) is above average.* **Validated**

#### 5.8.6. Hypotheses with Respect to "Automatic Speech Understanding for Complete Commands"

The results with respect to this hypothesis are presented in Section 5.1.2 in Table 8. Assuming the availability of push-to-talk, we measured an average command recognition rate of 91.2%, which is fully above the threshold of 90%. We obtained 3.2% as the command recognition error rate, which is also better than the threshold of 5%. Therefore, we mark the following hypotheses both as validated.

**H8.** *(H-E-CmdRR): The command extraction rate (JC, JP, and CP) in the apron environment is comparable to the quality already achieved in the approach domain by ABSR (command extraction rate for simulation-relevant commands >90%).* **Validated**

**H9.** *(H-E-CmdER): The command extraction error rate (conditions JC, JP, and CP) in the apron environment is comparable to the quality already achieved in the approach domain by ABSR (command extraction error rate for simulation-relevant commands <5%).* **Validated**

The results with respect to callsign recognition are also presented in Table 8. The callsign recognition rate of 97.4% is better than the threshold of 97%, and the callsign recognition error rate of 1.3% is also better than the threshold of 2%. Therefore, we mark the following hypotheses both as validated.

**H10.** *(H-E-CsgRR): The callsign extraction rate (conditions JC, JP, and CP) in the apron environment is comparable to the quality already achieved in the approach domain by ABSR (>97%).* Validated

**H11.** *(H-E-CsgER): The callsign extraction error rate (conditions JC, JP, and CP) in the apron environment is comparable to the quality already achieved in the approach domain by ABSR (callsign extraction error rate <2%).* Validated

## 6. Discussion

The STARFiSH project was of course subject to some restrictions that determined what was possible to research in the given time and budget. This section, therefore, discusses possibilities for improvements that could be addressed in the future and highlights some aspects that proved to be useful within this project.

The SFMECA (see Sections 2.3 and 5.7) produced no RPNs that mandated mitigation actions. This was due to the environment in which the project was executed: Areas of responsibility for the apron control did not include runways and no automatic alerting functions were implemented at the baseline A-SMGCS. For an environment without these limitations, the SFMECA is expected to produce different results and additional challenges for usability and safety. In addition, while the SFMECA itself was chosen as a proven instrument, there are suggestions for amendments to the methodology which could be used in order to address its specific shortcomings [48].

The results in Sections 5.1.1 and 5.1.2 show that the use of voice activity detection significantly degrades the overall performance. The push-to-talk signal should therefore be used whenever possible. Especially in an operational scenario, voice activity detection should not be considered as an alternative, since the push-to-talk signal is in use anyway, and technical access should not be an issue. Nevertheless, in non-operational scenarios where, for technical reasons, the push-to-talk signal might not be available, more modern approaches to voice activity detection based on neural network architectures could be exploited [49].

In Section 3.2.3, the rule-based algorithm for speech understanding was mentioned. This approach, of course, offers a very precise control about what is extracted and how the extraction itself takes place. The disadvantage of this method is, on the other hand, that every adaptation has to be programmed manually, which can create a lot of effort. Future projects could ease the adaptation process by fine-tuning pre-trained language models such as BERT [50], which could then recognize the different elements of the ontology [51].

The iterative approach taken for the development of the whole system and also for the training and improvement of the speech recognition and understanding modules proved to be very useful throughout this project. The different prototypes made it possible to involve the apron controller (end-users) already at an early stage of development and to incorporate their feedback in future prototypes. That not only improved the system in itself but also made the controllers involved and interested in the system and in what can be achieved with such a technology. The iterative improvement of the speech recognition parts was also useful with respect to the transcription and annotation process of the recorded data. As the recognition performance of these components became better over the iterations, the manual work to correct and verify transcriptions and annotations could be reduced.

One of the next steps should be to move the developed system from the simulation into an operational environment to see how big the difference to real world operations is and what obstacles have to be overcome. A first step could be to run the system in shadow mode so that it does not interfere with the operating systems, but operational experts could monitor how the system would react.

## 7. Conclusions

The STARFiSH project was the first to implement a speech recognition and understanding system for a complex apron environment at Frankfurt Airport. DLR's ABSR system was successfully coupled with the commercial A-SMGCS system from ATRiCS, i.e., a previously prototypical technology from a scientific environment was integrated into a commercial system that is available on the market. The solution was iteratively improved and finally tested in validation trials with 14 different apron controllers in 29 simulation runs in the tower simulator of Fraport. A total of 43 h of validation data (radar, audio, HMI inputs, etc.) were recorded and subsequently analyzed.

A main objective of the STARFiSH project was to prepare the usage of an artificial intelligence-powered speech recognition and understanding system in the safety-critical environment of the ops room at a European major hub airport. The formal method SFMECA (=Software Failure Modes, Effects, and Criticality Analysis) for risk assessment and subsequent identification of mitigation measures was applied with the very encouraging result that no error case would lead to an increased or unacceptable risk. At the same time, it could be shown that such an AI-equipped application can be operated safely in aviation and, moreover, does not have a negative impact on the controllers' situational awareness.

When supported by ABSR, the controllers made more than six times fewer manual entries into the A-SMGCS. This already includes the correction of wrong or missing recognitions from the speech recognition and understanding support. A recognition rate of 91.8% on the command level was observed, i.e., the callsign, the command type, the command values, e.g., taxi routes, and the command conditions were correctly extracted in 91.8% of the cases.

**Author Contributions:** Conceptualization, M.K. and H.H.; methodology, H.H.; software, M.K., H.H., O.O., S.S. (Shruthi Shetty) and H.E.; validation, M.K., H.W., M.M. and S.S. (Susanne Schacht); formal analysis, H.H.; investigation, H.H. and M.K.; resources, H.W. and S.S. (Susanne Schacht); data curation, H.W., M.K., H.H. and O.O.; writing—original draft preparation, M.K., O.O. and H.H.; writing—review and editing, O.O., H.W., M.M., H.E., S.S. (Shruthi Shetty) and S.S. (Susanne Schacht); visualization, H.H., M.K., H.E., O.O., M.M. and S.S. (Susanne Schacht); supervision, M.K.; project administration, M.K.; funding acquisition, M.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** The project STARFiSH was funded by the German Federal Ministry of Education and Research, under support code 01IS20017C.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** We would like to thank the Fraport controllers for their participation in this study.

**Conflicts of Interest:** The authors declare no conflict of interest. The funding sponsors had no role in the design of this study, in the collection, analyses, or interpretation of the data, in the writing of the manuscript, and in the decision to publish the results.

## Appendix A

The following dependent variables of the final validation trials are considered. The respective results of a dependent variable are each compared between the different operational conditions within a scenario.

### *Appendix A.1. DV-Input: Number of Manual Inputs for Control by Controllers/Simulation Pilots*

The manual inputs are counted. Since the inputs are identifiable by type, certain types are highlighted, if necessary, should it be found that some types occur particularly frequently or infrequently. The total count is compared between simulation runs with or without ABSR support.

These dependent variables are used to validate/falsify the following hypotheses:

- DV-Input-H-C-less_input (ABSR for the controllers reduces the number of manual inputs).
- DV-Input-H-P-less_input (ABSR for the simulation pilots reduces the number of manual inputs).

### *Appendix A.2. DV-Cog-Res: Measurement of Cognitive Resources by Secondary Task*

The cognitive resources are measured by means of a secondary task, i.e., a task the test subject (controller or simulation pilot) performs during a scenario in parallel to the main task. This is a secondary task that the subject is only allowed to perform when no mental resources are needed for the main task. The secondary task consists of performing a repeated Stroop test in a web application, see Section 4.4. The number of correctly mastered tests in a given time period is a measure of free cognitive resources. For this purpose, the responses per item are categorized as correct/wrong, and the number per time is plotted as a histogram and compared for simulation runs with and without ABSR support, respectively. These dependent variables are used to validate/falsify the following hypothesis:

- DV-Cog-Res-H-C-more_cog_res (more free cognitive resources of the controller due to ABSR).

### *Appendix A.3. DV-Workload Scoring by NASA TLX*

This dependent variable is used to validate/falsify the following hypothesis:

- DV-Workload-H-C-less_workload (less controller workload due to ABSR).

### *Appendix A.4. DV-Sit-Aw Scoring According to SHAPE-SASHA*

This dependent variable is used to validate/falsify the following hypothesis:

- DV-Sit-Aw-H-C-sit_aw_ok (situational awareness of the controller).

### *Appendix A.5. DV-Trust: Scoring According to SHAPE-SATI*

These dependent variables are used to validate/falsify the following hypotheses:

- DV-Trust-H-C-conf (automation trust of the controller).
- DV-Trust-H-P-conf (automation trust of the simulation pilot).

*Appendix A.6. DV-CmdRR: Command Extraction Rate*

This dependent variable is used to validate/falsify the following hypothesis:

- DV-CmdRR-H-E-CmdRR (comparable command extraction rate as in the approach environment).

*Appendix A.7. DV-CmdER: Command Extraction Error Rate*

This dependent variable is used to validate/falsify the following hypothesis:

- DV-CmdER-H-E-CmdRR (comparable command extraction error rate as in the approach environment).

*Appendix A.8. DV-CsgRR: Callsign Extraction Rate*

This dependent variable is used to validate/falsify the following hypothesis:

- DV-CsgRR-H-E-CsgRR (comparable callsign extraction rate as in the approach environment).

*Appendix A.9. DV-CsgER: Callsign Extraction Error Rate*

This dependent variable is used to validate/falsify the following hypothesis:

- DV-CsgER-H-E-CsgER (comparable callsign extraction error rate as in the approach environment).

## Appendix B

The task of the module "Concept Interpretation" is to transfer only those commands into the assistance system that are plausible and fit into the current traffic context. In the following, we describe the steps already mentioned in Section 3.2.5 in more detail.

*Appendix B.1. Preprocessing*

The modules described in Sections 3.2.2 and 3.2.3 generate data telegrams for the respective assigned working position. These data telegrams contain, among other things, the extracted ATC concepts with the semantics according to the ontology for the annotation of ATC utterances. An example of the logical content of a data telegram is presented in Box A1.

**Box A1.** Logical content example of a data telegram which has to be preprocessed for the A-SMGCS.

```
Sender: MC East
Callsign: DLH4YE
Command: GREETING
Command: TAXI (TO) V106
Command: TAXI (VIA) L
```

The interpretation of such a data telegram requires several checking steps, depending on the commands contained, before one or more inputs can be safely made to the A-SMGCS. Basically, this step checks whether the working position assigned to the sender is authorized to make entries for the callsign or whether another working position is responsible for the aircraft of this callsign.

*Appendix B.2. Highlighting the Aircraft Symbol on the Basis of the Recognized Callsign*

If the basic check of this preprocessing is successful, the corresponding aircraft symbol is highlighted at the assigned working position to inform the human operator for which flight a command has been recognized.

*Appendix B.3. Checking and Interpretation*

Depending on the command type, the following checks and computation steps are performed depending on the characteristics of the command received.

Appendix B.3.1. Triggering Multiple Actions Based on a Single Command

Some commands require that multiple actions are triggered by the same command in the correct order. For example, a TAXI command should trigger a TAXI clearance in the system and create a taxi route. It may also be necessary to modify an existing route and cancel stop instructions in certain situations.

Appendix B.3.2. Discarding Commands Incompatible with the Traffic Situation

It may happen that a command is received that does not make sense in the current traffic situation, e.g., a TAXI command destination for an inbound flight that contains a runway as the destination of the route. If possible, such cases are detected by checking a set of rules. The command is then ignored, and a message is displayed to the human operator. The cause of an incompatible command cannot be determined here. It could be an error of the controller, or it could originate in the ABSR system.

Appendix B.3.3. Correctly Interpret Context-Dependent Commands

Some commands must be interpreted differently depending on the flight plan data or other circumstances identified by the A-SMGCS. For example, if a special routing procedure is set for a flight in the system depending on certain conditions in the database, the system must assign a different route than in the normal case. The same utterance by the controller, therefore, leads to different results in the system depending on the data situation.

Appendix B.3.4. Completing Incomplete Commands from Current Traffic Situation

There are commands that do not contain all the necessary information to be able to implement them directly. For example, the command “GIVE_WAY . . . A320 RIGHT" needs further analysis, assuming that there is more than one A320 aircraft moving at the airport. The transcription part “from the right” can be ambiguous, thus it needs to be determined algorithmically which aircraft is probably correct from the controller’s and pilot’s perspective. In such cases, configuration tables, algorithms, state machines, and rules stored in the code are used to generate the correct command appropriate to the traffic situation.

Appendix B.3.5. Conversion of Commands

Once a command has been established by the previous checks, it can be implemented. This means that the command is entered into the A-SMGCS, i.e., the internal system state is changed to reflect the command. For the controller, this results in visual feedback on the working position. For example, the callsign that the controller addressed is highlighted near the corresponding aircraft symbol on the ground situation display. A change in the route is illustrated by colored lines, and changes in clearances are indicated on the label of the corresponding flight.

Appendix B.3.6. Dealing with Detected Errors

If a command does not pass one of the plausibility checks, an error message is displayed. This gives the human operator the option to check the situation and either ignore it or correct it.

Appendix B.3.7. Undetected Errors and Identification of Error Sources

It is not possible to identify for each command whether it is operationally correct. It is also not possible to determine whether a detected error originates from the ABSR system or was made by the controller. The controller, as the user of the system, must therefore observe the output of the A-SMGCS and anticipate, detect, and correct error situations not detected by the system. In the training of the controllers, this behavior is trained specifically and repeatedly, since errors of the human actors involved (e.g., the pilots) and the electronic systems must always be expected. It is always necessary to make a trade-off between the

recognition rate and the recognition error rate. For example, a 0% error rate can be achieved simply by discarding every recognized command.

## References


1. Kleinert, M.; Shetty, S.; Helmke, H.; Ohneiser, O.; Wiese, H.; Maier, M.; Schacht, S.; Nigmatulina, I.; Sarfjoo, S.S.; Motlicek, P. Apron Controller Support by Integration of Automatic Speech Recognition with an Advanced Surface Movement Guidance and Control System. In Proceedings of the 12th SESAR Innovation Days, Budapest, Hungary, 5–8 December 2022.
2. International Civil Aviation Organization (ICAO). *Advanced Surface Movement Control and Guidance Systems (ASMGCS) Manual, Doc 9830 AN/452*, 1st ed.; International Civil Aviation Organization (ICAO): Montréal, QC, Canada, 2004.
3. Helmke, H.; Ohneiser, O.; Mühlhausen, T.; Wies, M. Reducing Controller Workload with Automatic Speech Recognition. In Proceedings of the 35th Digital Avionics Systems Conference (DASC), Sacramento, CA, USA, 25–29 September 2016.
4. Helmke, H.; Ohneiser, O.; Buxbaum, J.; Kern, C. Increasing ATM Efficiency with Assistant Based Speech Recognition. In Proceedings of the 12th USA/Europe Air Traffic Management Research and Development Seminar (ATM2017), Seattle, WA, USA, 26–30 June 2017.
5. European Commission. *L 36/10*; Commission Implementing Regulation (EU) 2021/116 of 1 February 2021 on the Establishment of the Common Project One Supporting the Implementation of the European Air Traffic Management Master Plan Provided for in Regulation (EC) No 550/2004 of the European Parliament and of the Council, Amending Commission Implementing Regulation (EU) No 409/2013 and Repealing Commission Implementing Regulation (EU) No 716/2014. Official Journal of the European Union: Luxembourg, 1 February 2021.
6. Helmke, H.; Rataj, J.; Mühlhausen, T.; Ohneiser, O.; Ehr, H.; Kleinert, M.; Oualil, Y.; Schulder, M. Assistant-Based Speech Recognition for ATM Applications. In Proceedings of the 11th USA/Europe Air Traffic Management Research and Development Seminar (ATM2015), Lisbon, Portugal, 23–26 June 2015.
7. Davis, K.H.; Biddulph, R.; Balashek, S. Automatic recognition of spoken digits. *J. Acoust. Soc. Am.* **1952**, *24*, 637–642. [CrossRef]
8. Juang, B.H.; Rabiner, L.R. Automatic speech recognition–a brief history of the technology development. *Ga. Inst. Technol. Atlanta Rutgers Univ. Univ. Calif. St. Barbar.* **2005**, *1*, 67.
9. Connolly, D.W. *Voice Data Entry in Air Traffic Control*; Report N93-72621; National Aviation Facilities Experimental Center: Atlantic City, NJ, USA, 1977.
10. Hamel, C.; Kotick, D.; Layton, M. *Microcomputer System Integration for Air Control Training*; Special Report SR89-01; Naval Training Systems Center: Orlando, FL, USA, 1989.
11. FAA. *National Aviation Research Plan (NARP)*; FAA: Washington, DC, USA, 2012.
12. Updegrove, J.A.; Jafer, S. Optimization of Air Traffic Control Training at the Federal Aviation Administration Academy. *Aerospace* **2017**, *4*, 50. [CrossRef]
13. Schäfer, D. Context-Sensitive Speech Recognition in the Air Traffic Control Simulation. Eurocontrol EEC Note No. 02/2001. Ph.D. Thesis, University of Armed Forces, Munich, Germany, 2001.
14. Tarakan, R.; Baldwin, K.; Rozen, R. An automated simulation pilot capability to support advanced air traffic controller training. In Proceedings of the 26th Congress of the International Council of the Aeronautical Sciences, Anchorage, Alaska, 14–19 September 2008.
15. Ciupka, S. Siris big sister captures DFS (original German title: Siris große Schwester erobert die DFS). *Transmission* **2012**, 1.
16. *Doc 4444 ATM/501*; ATM (Air Traffic Management): Procedures for Air Navigation Services. International Civil Aviation Organization (ICAO): Montréal, QC, Canada, 2007.
17. Cordero, J.M.; Dorado, M.; de Pablo, J.M. Automated speech recognition in ATC environment. In Proceedings of the 2nd International Conference on Application and Theory of Automation in Command and Control Systems (ATACCS'12), London, UK, 29–31 May 2012; IRIT Press: Toulouse, France, 2012; pp. 46–53.
18. Cordero, J.M.; Rodríguez, N.; de Pablo, J.M.; Dorado, M. Automated Speech Recognition in Controller Communications applied to Workload Measurement. In Proceedings of the 3rd SESAR Innovation Days, Stockholm, Sweden, 26–28 November 2013.
19. Nguyen, V.N.; Holone, H. N-best list re-ranking using syntactic score: A solution for improving speech recognition accuracy in Air Traffic Control. In Proceedings of the 2016 16th International Conference on Control, Automation and Systems (ICCAS), Gyeongju, Republic of Korea, 16–19 October 2016; pp. 1309–1314.
20. Nguyen, V.N.; Holone, H. N-best list re-ranking using syntactic relatedness and syntactic score: An approach for improving speech recognition accuracy in Air Traffic Control. In Proceedings of the 2016 16th International Conference on Control, Automation and Systems (ICCAS 2016), Gyeongju, Republic of Korea, 16–19 October 2016; pp. 1315–1319.
21. Helmke, H.; Kleinert, M.; Shetty, S.; Ohneiser, O.; Ehr, H.; Arilíusson, H.; Simiganoschi, T.S.; Prasad, A.; Motlicek, P.; Veselý, K.; et al. Readback Error Detection by Automatic Speech Recognition to Increase ATM Safety. In Proceedings of the 14th USA/Europe Air Traffic Management Research and Development Seminar (ATM2021), Virtual, 20–24 September 2021.
22. Ohneiser, O.; Helmke, H.; Shetty, S.; Kleinert, M.; Ehr, H.; Murauskas, Š.; Pagirys, T.; Balogh, G.; Tønnesen, A.; Kis-Pál, G.; et al. Understanding Tower Controller Communication for Support in Air Traffic Control Displays. In Proceedings of the 12th SESAR Innovation Days, Budapest, Hungary, 5–8 December 2022.

23. Helmke, H.; Kleinert, M.; Ahrenhold, N.; Ehr, H.; Mühlhausen, T.; Ohneiser, O.; Motlicek, P.; Prasad, A.; Zuluaga-Gomez, J. Automatic Speech Recognition and Understanding for Radar Label Maintenance Support Increases Safety and Reduces Air Traffic Controllers' Workload. In Proceedings of the 15th USA/Europe Air Traffic Management Research and Development Seminar (ATM2023), Savannah, GA, USA, 5–9 June 2023.
24. García, R.; Albarrán, J.; Fabio, A.; Celorrio, F.; de Oliveira, C.P.; Bárcena, C. Automatic Flight Callsign Identification on a Controller Working Position: Real-Time Simulation and Analysis of Operational Recordings. *Aerospace* **2023**, *10*, 433. [CrossRef]
25. Chen, S.; Kopald, H.D.; Elessawy, A.; Levonian, Z.; Tarakan, R.M. Speech inputs to surface safety logic systems. In Proceedings of the IEEE/AIAA 34th Digital Avionics Systems Conference (DASC), Prague, Czech Republic, 13–17 September 2015.
26. Chen, S.; Kopald, H.D.; Chong, R.; Wei, Y.; Levonian, Z. Read back error detection using automatic speech recognition. In Proceedings of the 12th USA/Europe Air Traffic Management Research and Development Seminar (ATM2017), Seattle, WA, USA, 26–30 June 2017.
27. Helmke, H.; Ondřej, K.; Shetty, S.; Arilíusson, H.; Simiganoschi, T.S.; Kleinert, M.; Ohneiser, O.; Ehr, H.; Zuluaga-Gomez, J.-P.; Smrz, P. Readback Error Detection by Automatic Speech Recognition and Understanding—Results of HAAWAII project for Isavia's Enroute Airspace. In Proceedings of the 12th SESAR Innovation Days, Budapest, Hungary, 5–8 December 2022.
28. Zuluaga-Gomez, J.-P.; Sarfjoo, S.S.; Prasad, A.; Nigmatulina, I.; Motlicek, P.; Ondřej, K.; Ohneiser, O.; Helmke, H. BERTRAFFIC: BERT-based joint Speaker Role and Speaker Change Detection for Air Traffic Control Communications. In Proceedings of the 2022 IEEE Spoken Language Workshop Technology Workshop (SLT 2022), Doha, Qatar, 9–12 January 2023.
29. Helmke, H.; Slotty, M.; Poiger, M.; Herrer, D.F.; Ohneiser, O.; Vink, N.; Cerna, A.; Hartikainen, P.; Josefsson, B.; Langr, D.; et al. Ontology for transcription of ATC speech commands of SESAR 2020 solution PJ.16-04. In Proceedings of the IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), London, UK, 23–27 September 2018.
30. Kleinert, M.; Helmke, H.; Moos, S.; Hlousek, P.; Windisch, C.; Ohneiser, O.; Ehr, H.; Labreuil, A. Reducing Controller Workload by Automatic Speech Recognition Assisted Radar Label Maintenance. In Proceedings of the 9th SESAR Innovation Days, Athens, Greece, 2–6 December 2019.
31. Lin, Y. Spoken Instruction Understanding in Air Traffic Control: Challenge, Technique, and Application. *Aerospace* **2021**, *8*, 65. [CrossRef]
32. Ohneiser, O.; Helmke, H.; Kleinert, M.; Siol, G.; Ehr, H.; Hobein, S.; Predescu, A.-V.; Bauer, J. Tower Controller Command Prediction for Future Speech Recognition Applications. In Proceedings of the 9th SESAR Innovation Days, Athens, Greece, 2–5 December 2019.
33. Ohneiser, O.; Sarfjoo, S.; Helmke, H.; Shetty, S.; Motlicek, P.; Kleinert, M.; Ehr, H.; Murauskas, Š. Robust Command Recognition for Lithuanian Air Traffic Control Tower Utterances. In Proceedings of the InterSpeech 2021, Brno, Czech Republic, 30 August–3 September 2021.
34. Boehm, B. A Spiral Model of Software Development and Enhancement. *IEEE Comput.* **1988**, *21*, 61–72. [CrossRef]
35. Neufelder, A.M. *Effective Application of Software Failure Modes Effects Analysis*; Quanterion Solutions, Incorporated: New York, NY, USA, 2017.
36. Povey, D. Online Endpoint Recognition. 2013. Available online: https://github.com/kaldi-asr/kaldi/blob/master/src/online2/online-endpoint.h (accessed on 15 May 2023).
37. Povey, D.; Peddinti, V.; Galvez, D.; Ghahremani, P.; Manohar, P.; Na, X.; Wang, Y.; Khudanpur, S. Purely sequence-trained neural networks for ASR based on lattice-free MMI. *Interspeech* **2016**, *2016*, 2751–2755.
38. Povey, D.; Ghoshal, A.; Boulianne, G.; Burget, L.; Glembek, O.; Goel, N.; Hannemann, M.; Motlicek, P.; Qian, Y.; Schwarz, P.; et al. The Kaldi Speech Recognition Toolkit. In Proceedings of the IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, IEEE Signal Processing Society, Waikoloa, Big Island, HI, USA, 11–15 December 2011.
39. Kleinert, M.; Helmke, H.; Shetty, S.; Ohneiser, O.; Ehr, H.; Prasad, A.; Motlicek, P.; Harfmann, J. Automated Interpretation of Air Traffic Control Communication: The Journey from Spoken Words to a Deeper Understanding of the Meaning. In Proceedings of the IEEE/AIAA 40th Digital Avionics Systems Conference (DASC), Virtual, 3–7 October 2021.
40. Nigmatulina, I.; Zuluaga-Gomez, J.; Prasad, A.; Sarfjoo, S.S.; Motlicek, P. A two-step approach to leverage contextual data: Speech recognition in air-traffic communications. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022.
41. Zuluaga-Gomez, J.; Nigmatulina, I.; Prasad, A.; Motlicek, P.; Vesely, K.; Kocour, M.; Szöke, I. Contextual Semi-Supervised Learning: An Approach to Leverage Air-Surveillance and Untranscribed ATC Data in ASR Systems. *Interspeech* **2021**, *2021*, 3296–3300.
42. Levenshtein, V.I. Binary codes capable of correcting deletions, insertions, and reversals. *Sov. Phys. Dokl.* **1966**, *10*, 707–710.
43. Stroop, J.R. Studies of interference in serial verbal reactions. *J. Exp. Psychol.* **1935**, *18*, 643–662. [CrossRef]
44. Maier, M. Workload-Gauge. Available online: https://github.com/MathiasMaier/workload-gauge (accessed on 15 May 2023).
45. Helmke, H.; Shetty, S.; Kleinert, M.; Ohneiser, O.; Prasad, A.; Motlicek, P.; Cerna, A.; Windisch, C. Measuring Speech Recognition Understanding Performance in Air Traffic Control Domain Beyond Word Error Rates. In Proceedings of the 11th SESAR Innovation Days, Virtual, 7–9 December 2021.
46. Hart, S.G. NASA-TASK LOAD INDEX (NASA-TLX); 20 years later. In Proceedings of the Human Factors and Ergonomics Society, San Francisco, CA, USA, 16–20 October 2006; Volume 50, pp. 904–908.

47. Dehn, D.M. Assessing the Impact of Automation on the Air Traffic Controller: The SHAPE Questionnaires. *Air Traffic Control Q.* **2008**, *16*, 127–146. [CrossRef]
48. Di Nardo, M.; Murino, T.; Osteria, G.; Santillo, L.C. A New Hybrid Dynamic FMECA with Decision-Making Methodology: A Case Study in an Agri-Food Company. *Appl. Syst. Innov.* **2022**, *5*, 45. [CrossRef]
49. Mihalache, S.; Burileanu, D. Using Voice Activity Detection and Deep Neural Networks with Hybrid Speech Feature Extraction for Deceptive Speech Detection. *Sensors* **2022**, *22*, 1228. [CrossRef] [PubMed]
50. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
51. Zuluaga-Gomez, J.; Vesely, K.; Szöke, I.; Motlicek, P.; Kocour, M.; Rigault, M.; Choukri, K.; Prasad, A.; Sarfjoo, S.S.; Nigmatulina, I.; et al. ATCO2 corpus: A Large-Scale Dataset for Research on Automatic Speech Recognition and Natural Language Understanding of Air Traffic Control Communications. *arXiv* **2022**, arXiv:2211.04054.