Process Model for Integration of Speech Recognition and Understanding in Multiple Remote Tower Control Simulations

Oliver Ohneiser[1,2*], Hartmut Helmke[1], Sebastian Schier-Morgenthal[1], Umair Ahmed[2]

[1]Institute of Flight Guidance, Department Controller Assistance, German Aerospace Center (DLR), Lilienthalplatz 7, 38108 Braunschweig, Germany; *oliver.ohneiser@dlr.de, https://orcid.org/0000-0002-5411-691X

[2]Institute for Informatics, Clausthal University of Technology, 38678 Clausthal-Zellerfeld, Germany

SNE 35(2), 2025, 105-115, DOI: 10.11128/sne.35.tn.10735
Submitted: 2025-05-20; Revised: 2025-05-29
Accepted: 2025-06-10
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Simulations – especially human-in-the-loop real-time simulations – are important in the air traffic control (ATC) domain to train controllers and to test new features for controller working positions. One important reason for such simulations is the measurement of human workload. Verbal communication of aviation operators – contributing to this workload – is a central mean for safety and efficiency of air traffic. Speech recognition and understanding (ASRU) has reached pre-industry level, is about to enter operations, and therefore will become a vital part in training. The technology affects working procedures and reduces controller workload by roughly 20%. Thus, ASRU must be considered in simulations.

This paper describes a process model to integrate ASRU in ATC simulations. The model consists of three steps for efficient integration and adaption of ASRU: (1) collection of in-domain speech data for tuning of acoustic and language models, (2) compilation of configuration files and adaptation of speech understanding algorithms, and (3) manual checking of automatic transcriptions and extracted, semantic meanings of speech utterances.

We evaluate the process using a multiple remote tower environment case study. In this study, recognition error rates for words and callsigns were reduced by a factor of three compared to first simulations and command recognition rates increased from 81% to 92%.

Similarly feasible results are expected for other new ATC simulations with ASRU using the proposed process model.

Introduction

Simulating air traffic control (ATC) is crucial for training of air traffic controllers (ATCos) outside of their operational environment. They can train new or seldom executed operational procedures and test new features for controller working positions before potential deployment. Radio telephony communication between ATCos and cockpit crews is in general a crucial part of these simulations [1].

The transformation of operational, analogue voice signals into spoken words, intended meanings, and further provision of the digitalized ATC commands for downstream applications has been subject to numerous research projects [2]. ASRU is evaluated ready for operational usage and as such is expected to become a standard feature for future training and system development. As ASRU impacts ATC procedures in a way that workload of ATCos is reduced by 20% (cf. [3]) it must become a standard feature of ATC simulators.

Moreover, simulators will benefit from additional available data: Participating simulation staff such as supervisors or simulation pilots can be supported by receiving given ATC commands in real-time, e.g., for assistance, automation, data recording, and analysis.

The major challenge to achieve these benefits is to meet the requirements of the dynamical changing simulation environments including, e.g., airspace characteristics. To cope with these circumstances, we propose an iterative integration approach into ATC simulators with continuous improvement of ASRU.

This approach contains three steps which are repeated multiple times to iteratively improve the steps' quality:

1. Acoustic modelling: Integrating new words/accents,
2. Semantic modelling: Integrating new ATC concepts/commands,
3. Verification: Evaluation of transcriptions and recognized ATC concepts/commands.

We demonstrate the feasibility of the approach using a case study in which a simulation environment with multiple remote tower control is enhanced by ASRU [4].

To achieve reasonable speech recognition and understanding rates, it is important to have a large set of audio recordings from the aviation domain, which can be used as training data. Earlier ASRU projects in ATC used around 30 hours of in-house training data [5], [6].

However, the available open-access corpora are still limited in size compared to other domains as they are specially protected by telecommunication laws. In our context the main issue is, however, that voice recordings for the new application do not exist at all as the simulation training is required before the operational introduction. Furthermore, the important ASRU step of extracting relevant semantic ATC concepts from the transcribed word sequences, predominantly covered the ATC approach and en route environment with operational and simulation audio recordings. The aerodrome environment including tower, multiple remote tower, and apron has only been tackled to a lesser extent in simulated environments.

To cope with these challenges, we further present two different tools with tested user interfaces to (i) record ATC speech data in a structured way and (ii) for automatic transcription of aviation operator utterances. The collected data fed the training data pool of an ASRU module for a close-to-reality ATC multiple remote tower human-in-the-loop (HITL) simulation.

This paper outlines related work in Section 1. Section 2 presents the user interfaces and configuration files to support data recording, transcription, and annotation as required in the three repetitive steps of our process model. A transcription contains the word-by-word utterance content, whereas we use the term annotation for speech understanding, i.e., performing semantic interpretation of word sequences from the transcriptions. Section 3 explains the human-in-the-loop simulation setup for a multiple remote tower simulation, the integration of ASRU, and results on the ASRU performance at different stages. This is followed by conclusions in Section 4.

1 Related Work

1.1 ATC Speech Recognition & Understanding

Communication between ATCos and pilots with mutual understanding is a cornerstone of safe and efficient air traffic [7]. Speech recognition delivers the spoken word sequences of ATC utterances [8]. Instruction understanding extracts the semantic meaning of such word sequences [9]. A combined ASRU module can enable downstream applications or help to assess communication quality parameters [10], e.g., in HITL-simulations or operational environments.

Such an ASRU module for ATC communication follows a series of steps:

- First, aircraft callsigns and ATC commands that will most likely appear in the next ATCo utterances are predicted based on contextual data such as surveillance data [11].
- Second, a speech-to-text engine delivers the recognized sequences of words from analyzed ATC utterances [12].
- Third, a text-to-ATC-concept component extracts the relevant callsigns and ATC commands from the recognized word sequences [13].

These extracted ATC concepts follow a European-wide agreed ontology for semantic annotation of ATC utterances [14]. The performance of the speech-to-text step is measured via word error rates, but more importantly the text-to-ATC-concept step is analyzed via recognition rates and error rates for the extracted ATC concepts [15]. The output of the ASRU module is then used to support ATCos within their workstation displays [4]. The possibility to quickly add new features from ASRU module outputs into displays used in HITL-simulations helps to get swift feedback from human operators on the features' feasibility [16].

An early assessment has been done for the feasibility of ATC automation within a simulation environment [17]. Virtual simulation pilots using an ASRU module can support automating ATC simulations [18].

1.2 Simulators in the Aviation Domain

Simulations are a central mean in ATC [19]. Simple experiments such as effects of modified airspace [20] or landing clearances [21] can be evaluated without human operators.

However, there are many other ATC operation experiments that require human involvement in situation assessment, decisions, and communication. The analysis of such relationships usually requires a HITL-real-time simulation (RTS) [22]. This methodology is as well common for simulations with pilots [23]. HITL-RTS can be scripted with the help of, for example, ATC scenarios to include relevant air traffic situations in the different domains tower, approach, and en route [24].

There exist some simulators at national air navigation service providers (ANSPs) or research institutes. There are even publicly available ATC simulators [25] and open source simulators based on open data [26]. ATC simulators represent very different fidelities regarding their realism and completeness [19]. They usually also focus on just one of the domains, i.e., approach/en route [27] or tower [28]. We focus on the tower environment also comprising remote and multiple remote towers [29]. While speech recognition is available for some simulators to automate or support simulation pilots, speech understanding to enable operational support (e.g., flight strip handling) exists to a much lesser automation degree.

1.3 Speech Recognition & Understanding and its Influence on Controller Workload

The ATCo workload in a HITL-RTS with and without ASRU was objectively measured with the time needed for a secondary task [12].

In the baseline condition, ATCos needed to maintain aircraft radar labels completely manual, i.e., enter the command content via mouse and drop-down menus.

In the solution condition, ATCos were automatically supported by an ASRU system [12].

After compensating sequence effects – depending on the first simulation run was baseline or solution – and eliminating outlier the average time to solve the secondary task was almost six minutes for the baseline condition (347 s), but less than five minutes for the solution condition (289 s).

Hence, the ATCos needed 20% more time to perform a secondary task if they were not supported by automatic radar label maintenance with the ASRU output in their primary ATC task. Thus, ATC simulators without ASRU can cause workload measurement errors of up to 20% in the future compared to operational environments with ASRU.

1.4 Creating Speech Recognition Models

Thanks to Siri or Alexa speech recognition has now reached the general public. However, these engines are not usable for ATC applications due to insufficient recognition performance and data privacy issues.

Recently, some general engines, so called open source end-to-end models like, Whisper [30] or wav2vec [31] gained more and more attention like with an application for ATC [32]. These end-to-end models often come with easier implementation and adaptation processes. This enables also non-speech recognition experts to reach suitable performances in different target areas. These engines have seen already ten thousand of hours of normal English conversation.

Nevertheless, some fine-tuning with, e.g., ten hours of airport dependent data is necessary for ATC applications. This fine-tuning has shown to reduce the word error rate (WER) from 90% to 5% for CoquiSTT [33]. The same was reported for DeepSpeech [34]. The STARFiSH [5] and HAAWAII project [6] used a basic engine, which has already seen a lot of ATC training data. This engine was fine-tuned with roughly 30 hours of domain dependent data. The HAAWAII project started with just one hour of domain dependent data from the Icelandic airspace. This one hour already reduced the WER from 50% to 33% while three further hours reduced it to 20%. This eases the effortful manual transcription task. The ATCO2 project utilizes unsupervised learning on 5000 hours of ATC voice recordings together with context information from radar data [35].

MITRE presents the FAA system DRAAS (DALR Remote Audio Access System), which, in principle, provides access to audio from 129 National Airspace facilities. More than 200,000 hours of silence-reduced audio are recorded each month, i.e., 2-3 billion ATC transmissions per year are recorded. This enables at least unsupervised learning [36].

Currently, end-to-end speech recognition models own the highest potential for the application in HITL-simulations. On one hand they are already trained on a large dataset of formal English language. On the other hand, they can easily be adapted without specific speech recognition expertise. Nevertheless, ATC applications require fine-tuning.

2 Simulation Setups and Tools

2.1 Online ATC Speech Recorder

For the adaptation of ASRU models within ATC applications such as acoustic model, language model, or command extraction model, training data is required to achieve an acceptable performance. Static or quasi-static data that frequently appear in ATC communication such as frequencies and airline names are quite easy to acquire as verbalized speech.

However, depending on the ASRU use case, the dynamic data comprises speech of ATCos and pilots, surveillance data, flight plans, meteorological data, ATC sector configurations, and many more.

The best recording environment for speech (and surveillance) data is the environment, in which the later ASRU-related ATC application is executed. Unfortunately, the targeted environment is not available in all cases as for instance conversion training and validation projects aim on setups which are not established in real operations. Moreover, ATCos' duty time is a rare resource [37] and should be used as little as possible for the preparation of the simulation setup.

Hence, the minimum setup should be easy to use and accessible from remote to provide the required training data for future working procedures without the need for ATCos to travel or train in advance. These requirements can be fulfilled with a website that ATCos can login to. A server as backend could offer a simplified traffic simulation and speech recording option.

One of the biggest challenges of a qualified data set given alternative data recording options is the feasibility for the later ASRU-related ATC application, i.e., the generated data content should be close to the data content expected in the final simulations. Hence, a good option would be a complete remote simulation, i.e., the ATCo manages interactive air traffic like at a normal controller working position with uttered ATC commands that are recorded. Again, a prioritization of requirements should be made in order to deliver a reasonable solution in a reasonable amount of time.

The simplest recording option consists of a sheet of paper with written ATC utterances that ATCos should read while being recorded on a headset. The recordings could help to learn acoustic models, e.g., the sound of ATC domain prosody. However, the language (sequence of words) would already be predefined and not realistically help to learn a language model, because ATCos more or less deviate from the International Civil Aviation Organization (ICAO) phraseology. This needs to be included into the ASRU models.

An enhanced setup version should give ATCos a greater level of freedom to formulate their own utterances just with some basic hints about the air traffic situation. The ATCos would see static figures with air traffic situations, the last utterances of the involved aircraft pilots, and some options on how to react in the current situation in a very basic style with ATC command type suggestions. This forces the ATCos to actively think about the situation, to use predefined aircraft callsigns, runways, airport names, and waypoint names as used in final simulations, and produces a more natural speech comparable to a minimum simulation. Such recorded speech data would support to learn a language model, i.e., the words and word sequences that ATCo utterances operationally contain and a command extraction model, i.e., what the ATCos mean by their utilized phraseology.

Figure **1** shows our implemented prototype of an online ATC speech recorder. As first step after login, the ATCos need to confirm their participation and data upload. One is then asked to walk through 20 different ATC scenarios. The online ATC speech recording environment offers static air traffic situations in a simple implementable map view as shown in Figure 1 and communication info that can be understood when going through the online tutorial with explanation boxes.

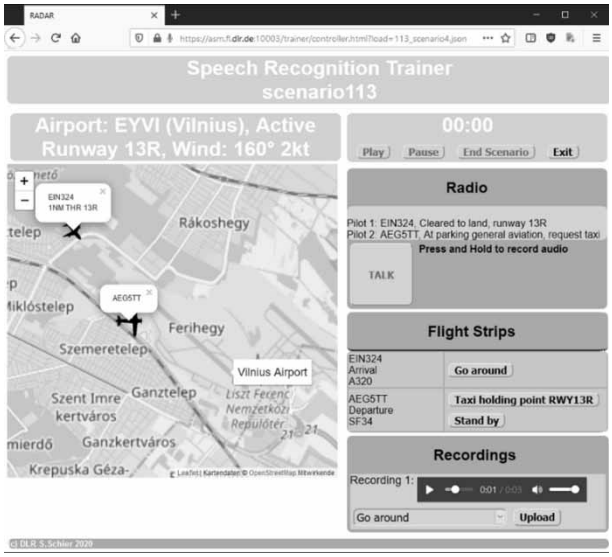

Figure 1: Screenshot of the Online-ATC-Speech-Recorder with a map view (left) and the communication info (right) for a given situation at Vilnius airport with a voice recording.

The map view presents one involved airport with some aircraft information on the left side. The right side shows scenario information (see Figure 1, light blue with white font), last radio calls (see Figure 1, light grey; from Pilot 1 and Pilot 2), flight strips of both involved aircraft with reasonable commands in the given situation (see Figure 1, yellow and blue), and options for recording, labelling, and upload of audio files (see Figure 1, red).

The corresponding ATCo voice utterance is recorded while pushing the "TALK"-button via the ATCo's headset. This speech recording setup forms the second lowest simulator fidelity category 'B' with the used electronic equipment attributed to category 'C' of table 1 in [19].

Nine ATCos from the Lithuanian ANSP and five ATCos from the Austrian ANSP contributed to an online-recording resulting in 667 audio files with one hour of net speech, i.e., each utterance lasts five seconds on average.

2.2 Online ATC Speech Recognition

The recording of audio files can as well be directly connected to an online speech-to-text engine to immediately receive the transcripts of ATCo or pilot utterances in the desired format. The developed browser-based application as shown in Figure 2 utilizes hypertext markup language (HTML), cascading style sheets (CSS), and JavaScript for the front-end as well as Python 3.8 and its Flask application programming interface (API) with the integration of DeepSpeech 0.9.3 for the back-end. The app has been tested within different browsers on Ubuntu 20 and 22 as well as Windows 10 and 11 leveraging ffmpeg and sox for audio recording and conversion from opus-files into 16 kHz wav-files.

After pressing "A" on the keyboard for ATCo mode or "P" for pilot mode, the audio recording and live transcription starts. After releasing the pressed key, the audio file is saved with the current timetick in its filename. This timetick is reused for the transcription file name. The speech-to-text engine DeepSpeech continuously delivered the transcription of recognized words even if the utterance has not been completed yet, e.g., "false" in Figure 2 indicates that the endpoint of the utterance has not been reached as the push-to-talk button is not released yet.

The console version of the application is also able to use defined speech pauses as the endpoint for utterance recordings and their transcriptions. The audio filename, recognized words, and endpoint information are stored in a JavaScript object notation (JSON) file, which eases the readability by machines. as shown in the black bottom part of Figure 2.

As DeepSpeech offers to integrate own speech recognition models, the speech-to-text quality can be improved through utilizing sophisticated acoustic models and language models trained on ATC data. The application setup forms the lowest simulator fidelity category 'A' with the used voice recognition capability attributed to category 'C' of table 1 in [19].

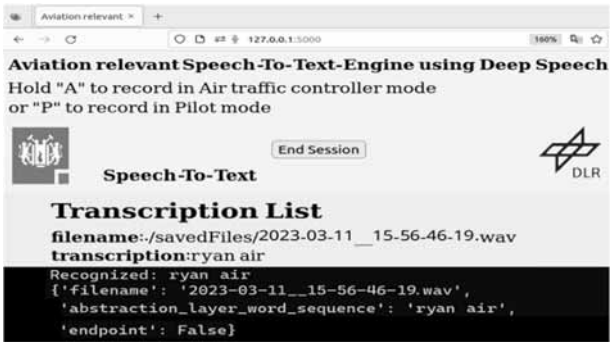

Figure 2: Screenshot-Collage of the Online-ATC-Speech-Recognition Interface with filename format, recognized word sequence, and endpoint information in graphical user interface (top) and JSON format (bottom).

2.3 ASRU Simulation Configuration

The ASRU module requires two important configuration inputs to be customized for a given ATC scenario. These inputs are set via configuration files in JSON format. The two files contain the ATC command types and the ATC concepts that shall or shall not be extracted for the given scenario.

Figure 3 shows three example entries of a JSON file for supported ATC command types and their qualifiers nested in the *Commands* array.

The *Type* key indicates the main part of the ATC command type while the *SndTypeName* indicates the sub part.

This allows to define command types such as *CLEARED LANDING*, *CLEARED ILS*, *TAXI TO*, or *VACATE VIA*. Some command types can have a *Qualifier* to specify the value such as *LEFT* or *EXPEDITE* for the command type *VACATE*.

The key *SupportedInThisAirspace* marks if the command type should be considered for the current simulation setup. This enables, e.g., to deactivate *PUSHBACK* commands for en route traffic scenarios (if *false*).

```
{ "Commands": [
 { "Type": "CLEARED",
   "SndTypeName": ["TAKEOFF","TOUCH_GO",
                   "LANDING","ILS",
                   "VISUAL","VIA","TO"] },
   "SupportedInThisAirspace": true,
 { "Type": "TAXI",
   "SndTypeName": ["TO","VIA"],
   "SupportedInThisAirspace": true },
 { "Type": "VACATE",
   "SndTypeName": ["VIA"],
   "Qualifier": ["LEFT","RIGHT","EXPEDITE"],
   "SupportedInThisAirspace": true }
]}
```

Figure 3: Configuration file excerpt example with supported ATC command types and qualifiers in JSON format.

Figure 4 shows two example entries of a JSON file for ATC concepts with further relevant values nested in the *AtcConcepts* array. The *Name* defines how an entity of an ATC concept shall be referred to. The *Locator*, e.g., contains a four-letter ICAO code for an airport such as *EYVI* for Vilnius where the concept is relevant.

```
{  "AtcConcepts": [
 { "Name": "JOZA",
   "Locator": "LHBP",
   "KeyWordSeq": ["joza","joza point",
                  "juliett oscar zulu alfa"],
   "CommandTypes": ["DIRECT_TO"],
   "ConceptType": "WAYPOINT",
   "AdditionalInfo":
      [{"LatLong": "47.592500 21.557222"}] },
 { "Name": "VILNIUS_TOWER",
   "Locator": "EYVI",
   "KeyWordSeq": ["tower vilnius","vilnius",
                  "vilnius airport",
                  "vilnius tower"],
   "CommandTypes": ["STATION"],
   "ConceptType": "FREQUENCY_POSITION",
   "AdditionalInfo":
   [{"FrequencyValues": "118.200"}] } ]}
```

Figure 4: Configuration file excerpt example with ATC concepts, their word sequences, and additional information in JSON format.

The *KeyWordSeq* array lists all word sequences that should be mapped to the concrete ATC concept if they are found in an utterance transcription. The *CommandTypes* array lists all types that could make use of the ATC concept, i.e., a runway could e.g., be used for *CLEARED TAKEOFF* or *HOLD_SHORT*. The *ConceptType* represents the nature of the ATC concept. *AdditionalInfo* might contain numeral data about latitude/longitude of a waypoint or a frequency depending on the *ConceptType*.

With these two configuration files, it is possible to list the expected ATC commands and concepts. Hence, commands with Mach numbers can be excluded for tower scenarios and only those waypoints are added to the configuration file that exist in the current ATC environment. This enables to manually customize the command recognition, i.e., adapting speech understanding to the application without ASRU expert knowledge.

3 Multiple Remote Tower ASRU Simulations and Results

3.1 Real-Time Simulation with Controllers

For the HITL-RTS described in this paper, ATCos were responsible for three airports at the same time (named Vilnius, Kaunas, Palanga). The ATCos had three rows of monitors presenting the camera image of the respective airports and a head-down ATC system unit to monitor and influence the given traffic (see Figure 5). The communication with simulation pilots was done via radio telephony (over IP) on three different frequencies.

Figure 5: Multiple Remote Tower Setup in the Remote Tower Lab of DLR Braunschweig: One ATCo is controlling three airports, using head-down electronic flight strips and a tower radar display.

The ASRU module (1) automatically transcribed all ATCo utterances word-by-word and (2) automatically annotated the word sequences with the semantic meanings using a command extraction algorithm and the defined ontology of European ATC stakeholders.

The relevant recognized ATC commands were (3) displayed in an abstracted form in an ATCo display to be confirmed/maintained. An example transcription following defined transcription rules was: "*airest cargo five five zero vilnius tower you are cleared to destination via erlos one delta departure squawk is two one seven four startup approved QNH one zero two one runway one three*".

The relevant ATC commands with values were extracted from the transcription. The annotation of the above example transcription in a human-readable format, ignoring the JSON tagging, is:

AEG550 STATION VILNIUS_TOWER
AEG550 CLEARED TO DESTINATION
AEG550 CLEARED VIA ERLOS_1D
AEG550 STARTUP
AEG550 INFORMATION QNH 1021
AEG550 INFORMATION ACTIVE_RWY RW13.

The relevant recognized callsign and ATC command values of each utterance were automatically shown to the ATCo on an electronic flight strip display. This means, the aircraft with the recognized callsign was highlighted and the content of the ATC commands was displayed as abbreviated information either in text form or as symbols (e.g., an aircraft engine icon for STARTUP). The ATCo only needed to check the highlighted commands and correct if needed in seldom cases. Hence, these automatically entered and displayed information from verbal ATC commands reduced the manual ATCo workload for electronic flight strip maintenance.

Our conducted HITL-RTS evaluates the benefit of an ASRU module to support tower ATCos with electronic flight strip maintenance in a multiple remote tower environment. The setups for the HITL-RTS and pre-trials form the highest simulator fidelity category 'E' of table 1 in [19]. Hence, they were very realistic, but costly in the RTS conduction.

The verbal ATCo utterances of 116 roughly 45-minutes long RTS runs have been analysed in order to compare the automatic ATC concept extraction results with the actually intended ATC concepts. Therefore, all RTS runs have been automatically transcribed word-by-word and annotated concept-by-concept. Afterwards all of them were manually checked and corrected if necessary.

The HITL-RTS campaigns have been conducted at six different points in time with slightly varying setups between 2017 and 2022 with tower ATCos from four European ANSPs as follows: 17 from the Lithuanian ANSP, 13 from the Hungarian ANSP, 7 from the Austrian ANSP, and 3 from the Finnish ANSP. The complete data set is called "116/40" as it contains 116 simulation runs of 40 ATCos. It comprises 177,847 transcribed words with 32,436 commands in 10,712 audio files.

The simulation setup for the Lithuanian (LIT, 52 simulation runs), Hungarian (HUN, 41 simulation runs), and Austrian (AUT, 16 simulation runs) ATCos differed only very slightly in airport names. The aircraft callsigns, airport layouts, configuration files, etc. remained the same. However, the simulation setup for the Finnish (FIN, 7 simulation runs) ATCos included different callsigns, airport layouts, and configurations despite being a multiple remote tower (MRT) simulation with three remote airports and comparable traffic amount and traffic mix, too.

A subset of the complete data set, i.e., the final HITL-RTS with Lithuanian and Austrian ATCos in winter 2022 are analysed specifically. This data set contains ten simulation runs from Austrian ATCos and eight simulation runs from Lithuanian ATCos, in the latter eliminating two simulation runs of one ATCo due to technical issues, i.e., the sub-data set is called "18/9" due to 18 simulation runs of 9 ATCos. It comprises 35,022 transcribed words with 6963 commands in 2437 audio files.

3.2 Iterative ASRU Simulation Results

First, we present the results achieved with the ASRU model during and after the final simulation runs on the 18/9 data set, respectively. Second, we detail the ASRU results given the same ASRU model for the 116/40 data set. Third, we show the improved results with our current ASRU model on the 116/40 data set.

The following tables show the recognition rates (Recog) and error rates (Err) on callsign level (Csgn) and command level (Cmd) as well as the WER if applicable. The recognition and error rate results do not sum up to 100% due to not shown *rejection rate*, i.e., correctly annotated commands were not recognized at all, e.g., a startup, pushback, and taxi clearance were given, but only startup and taxi were recognized. Then we have one rejection. If the pushback would be replaced by, e.g., a climb command or the pushback value would be recognized wrongly, it as an error.

Table 1 shows the results for ASRU performance of three different speech recognition modes for the 18/9 data set. The mode *Live* shows the ASRU results on the continuous audio stream during the simulation runs as 'perceived' by the ATCos with a Kaldi based speech-to-text engine that has seen the earlier available audio files – before 2022 – as training data. This training data encompassed 3.6h of LIT and 0.9h of AUT next to other ATC data sources that did not match the final MRT simulation setup with Lithuanian and Austrian ATCos. The mode *AllTrain1* shows the WER for speech recognition on recorded wav-files with a Coqui speech-to-text engine and a model that has been trained on all available audio files after the final simulation runs (roughly 17h). The command and callsign rates are then computed on the speech-to-text output. The mode *Perfect1* considers manual transcriptions, assuming no errors. As expected, the WER is highest with least training data in *Live* mode. The more data, the better for ASRU performance. Therefore, the iterative approach is helpful to collect, and steadily faster transcribe and annotate more data for the next phase.

Speech Recognition Mode	Word Error Rate	Cmd Recog Rate	Cmd Err Rate	Csgn Recog Rate	Csgn Err Rate
Live	10.7	80.7	7.0	94.1	2.2
AllTrain1	3.2	92.0	4.2	98.4	0.7
Perfect1	0.0	95.6	2.9	99.7	0.2

Table 1: First results in [%] for speech recognition and understanding of HITL-RTS runs (18/9 data set).

Table 2 presents the speech understanding metrics based on the *AllTrain1* mode while Table 3 shows the results for the *Perfect1* mode on the 116/40 data set.

Data Set	Cmd Recog Rate	Cmd Err Rate	Csgn Recog Rate	Csgn Err Rate
All	93.1	4.1	98.4	0.8
MRT_HUN	93.8	4.3	98.3	1.0
MRT_LIT	94.2	3.4	98.7	0.7
MRT_AUT	90.6	5.1	97.5	0.9
MRT_FIN	85.1	5.2	98.6	0.8

Table 2: First results in [%] for ATC concept recognition on 116/40 data set given a WER of 3.2%.

Data Set	Cmd Recog Rate	Cmd Err Rate	Csgn Recog Rate	Csgn Err Rate
All	96.0	2.7	99.4	0.4
MRT_HUN	95.8	3.2	99.0	0.7
MRT_LIT	97.3	1.7	99.8	0.1
MRT_AUT	94.7	3.6	99.4	0.3
MRT_FIN	90.0	3.7	99.4	0.2

Table 3: First results in [%] for ATC concept recognition on 116/40 data set given a WER of 0%.

As expected, the higher WER in Table 2 leads to worse recognition rates and error rates on semantic level than in Table 3. However, with the WER of 3.2%, the callsign recognition only decreases by roughly 1% absolute and the command recognition decreases by 3% absolute only. This demonstrates that the speech understanding process can compensate a lot of word errors through the use of contextual data, due to redundant information in the utterances, and due to word errors, that affect irrelevant portions of a sentence in some cases.

Now, we present the most recent results given the available complete data set for training in our latest iteration of the process model. We created a first ASRU model *PartTrain* – this encompasses acoustic model, language model, and command extraction model – with training based on speech data, correct transcriptions, and correct annotations of Vienna approach. This model was applied on a multiple remote tower data test set resulting in a WER of 77%, a command recognition rate of 1%, and a callsign recognition rate of 24% (see Table 4). These results are useless even if the ASRU model performs acceptable when applying to Vienna approach data on which it has been trained with a WER of 6.2%, a command recognition rate of 85.6% (error rate 5.3%), and a callsign recognition rate of 96.9% (error rate 1.2%).

Speech Recognition Mode	Word Error Rate	Cmd Recog Rate	Cmd Err Rate	Csgn Recog Rate	Csgn Err Rate
PartTrain	77.0	1.3	8.5	24.3	34.7
AllTrain2	2.7	94.8	3.0	99.1	0.4
AllTune	1.8	95.7	3.1	99.3	0.6
Perfect2	0.0	97.1	2.2	99.5	0.3

Table 4: Current results in [%] for ATC concept recognition on 116/40 data set.

We created a second ASRU model *AllTrain2* – this encompasses acoustic model, language model, and command extraction model – with training based on speech data, correct transcriptions, and correct annotations of many different available ATC environments including two en route environments, three approach environments, and an apron environment as well as some multiple remote tower data. This model was applied on the complete multiple remote tower data – that was already part of the training data – resulting in a WER of 2.7%, a command recognition rate of 95% (error rate 3%), and a callsign recognition rate of 99% (error rate 0.4%) as shown in Table 4.

When using the improved command extraction model with enhancements for seldom used commands or new commands such as *STATION*, based on transcriptions with a WER of 0% in mode *Perfect2*, we achieve a command recognition rate of 97% (error rate 2.2%) and a callsign recognition rate of 99.5% (error rate 0.3%).

We created a third ASRU model *AllTune* – this encompasses acoustic model, language model, and command extraction model – with fine-tuning the first model *PartTrain* with the same data as for the second ASRU model *AllTrain2*. The *AllTune* model was applied on the complete multiple remote tower data – that was already part of the fine-tuning data – resulting in a WER of 1.8%, a command recognition rate of 96% (error rate 3.1%), and a callsign recognition rate of 99.3% (error rate 0.6%) as shown in Table 4.

If four out of five ATC commands as given with a command recognition rate of around 80% in Table 1 *Live* mode are automatically recognized and entered correctly into digital flight strips, this already saves manual effort of the ATCo to enter command content into the controller working position. This result was already achieved based on a WER of 11%.

With an even lower WER, a positive effect on the speech understanding metrics as outlined in Table 1 and 4 can be expected. For example, the recognition of command types, values, qualifiers, and conditions as calculated with a command recognition rate of 92.5% requires a manual correction by the ATCo in less than every 13th recognized command if ASRU output was visualized, e.g., in digital flight strips. This again, could translate into less ATCo workload, i.e., faster execution times for a secondary task, which can be interpreted as a higher availability of mental capacity of ATCos if they get ASRU support.

If the callsign error rate is below 1%, this means that less than every 100th callsign is wrongly recognized and, in case of callsign highlighting in an ATCo display, might rarely drag ATCo attention to an unintended spot. However, ASRU can enable to very often drag ATCo attention to the desired display spots.

Independent of the concrete ASRU result values having the same order of magnitude for other multiple remote tower or ATC setups in general, the ATCo tool support with given ASRU performance showed to be a valuable support for HITL-RTS simulations in the ATC domain.

4 Conclusion

We presented an iterative process, which enables to adapt existing speech recognition and understanding (ASRU) models to new environments, for which in the beginning no recorded training voice utterances exist. This, however, is a prerequisite to use ASRU support already during first human-in-the-loop simulations for new environments. First speech data for rough adaptation of existing ASRU models can be gained by the presented web-based online tool. Efforts for traveling and training as required for HITL-RTS itself were not necessary. The word error rate (WER) from untrained models of approximately 77% decreased to 11% in the case study using the described process model's first iterations. Data from initial training and verification runs can be used to iteratively fine-tune existing ASRU models for final simulation runs, which in turn improve the ASRU performance further.

The integration of ASRU lead to feasible ATCo support for ATC HITL-RTS. It supports ATCos maintaining aircraft information in electronic flight strips even with a WER of 11%, because the resulting command recognition rates of above 80% are already sufficient to free mental capacity for ATC tasks as shown through a secondary task. The performance difference of the secondary task with and without ASRU support has demonstrated that. Without integration of ASRU support already during first simulations, the results with respect to ATCo workload measurements might be useless, because ASRU support can reduce ATCos' workload by 20%.

Using all recordings from the 12,500 utterances of 116 simulation runs with 40 different ATCos for fine-tuning an ASRU model enabled a WER of 3% resulting in command recognition rates of 92-95% and callsign recognition rates of 99%.

Similar results can be expected for other new ATC environments modelled in simulators when using the presented iterative approach starting with recordings supported by a web-based tool.

Funding

One of the HITL-RTS was part of SESAR2020's PJ.05-W2 (Sol97), "Digital Tower Technologies (DTT)" that received funding from the SESAR Joint Undertaking under the European Union's Horizon 2020 research and innovation program under grant agreement No 874470.

References

[1] Dow C, Histon J. Enroute ATC Industry Perceptions of Simulation Fidelity. *18th International Symposium on Aviation Psychology*. May 2015, Dayton, OH, 524-529.

[2] Nguyen VN, Holone H. Possibilities, Challenges and the State of the Art of Automatic Speech Recognition in Air Traffic Control. 2015.
DOI 10.5281/zenodo.1108428

[3] Helmke H, Kleinert M, Ohneiser O, Ahrenhold N, Klamert L, Motlicek P. Safety and Workload Benefits of Automatic Speech Understanding for Radar Label Updates. *Journal of Air Transportation* 32:4, 155-168, 2024.
DOI 10.2514/1.D0419

[4] Ohneiser O, Helmke H, Shetty S, Kleinert M, Ehr H, Murauskas Š, Pagirys T, Balogh G, Tønnesen A, Kis-Pál G, Horváth V, Kling F, Rinaldi W, Mansi S, Piazzolla G, Usanovic H. Understanding Tower Controller Communication for Support in Air Traffic Control Displays. *12th SESAR Innovation Days*. Dec 2022, Budapest, Hungary.

[5] 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.

[6] Helmke H, Ondřej K, Shetty S, Arilíusson H, Simiganoschi TS, Kleinert M, Ohneiser O, Ehr H, Zuluaga JP, Smrz P. Readback Error Detection by Automatic Speech Recognition and Understanding. *12th SESAR Innovation Days*. Dec 2022, Budapest, Hungary.

[7] ICAO. Doc 4444 – Procedures for Air Navigation Services – Air Traffic Management. Montréal, QC, Canada: International Civil Aviation Organization; 16 ed., 2016.

[8] Badrinath S, Balakrishnan H. Automatic Speech Recognition for Air Traffic Control Communications. *Transportation Research Record*. 2022; 2676(1): 798-810.

[9] Lin Y. Spoken Instruction Understanding in Air Traffic Control: Challenge, Technique, and Application. *Aerospace*. 2021; 8, 65.
DOI 10.3390/aerospace8030065

[10] Helmke H, Ohneiser O. Automatic Speech Recognition and Understanding in Air Traffic Management. *Aerospace*. 2024.
DOI 10.3390/books978-3-7258-0315-6

[11] Ohneiser O, Helmke H, Shetty S, Kleinert M, Ehr H, Murauskas Š, Pagirys T. Prediction and extraction of tower controller commands for speech recognition applications. *J. Air Transp. Manag*. 2021; 95, 102089.

[12] Ohneiser O, Helmke H, Shetty S, Kleinert M, Ehr H, Schier-Morgenthal S, Sarfjoo S, Motlicek P, Murauskas Š, Pagirys T, Usanovic H, Meštrović M, Černá A. Assistant Based Speech Recognition Support for Air Traffic Controllers in a Multiple Remote Tower Environment. *Aerospace*. 2023; 10(6):560.

[13] 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. *InterSpeech*; Aug/Sep 2021; Brno, Czechia.

[14] Helmke H, Slotty M, Poiger M, Herrer DF, Ohneiser O, Vink N, Cerna A, Hartikainen P, Josefsson B, Langr D, García Lasheras R, Marin G, Mevatne OG, Moos S, Nilsson MN, Boyero Pérez M. Ontology for transcription of ATC speech commands of SESAR 2020 solution PJ.16-04. *Proceedings of the IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*.
Sep 2018; London, UK.
DOI 10.1109/DASC.2018.8569238

[15] Helmke H, Shetty S, Kleinert M, Ohneiser O, Ehr H, Prasad A, Motlicek P, Cerna A, Windisch C. Measuring Speech Recognition and Understanding Performance in Air Traffic Control Domain Beyond Word Error Rates. *11th SESAR Innovation Days*. Dec 2021; Virtual.

[16] Ohneiser O, Helmke H, Ehr H, Gürlük H, Hoessl M, Mühlhausen T, Oualil Y, Schulder M, Schmidt A, Khan A, Klakow D. Air Traffic Controller Support by Speech Recognition. In: Ahram T, Karwowski W, Marek T, editors. *Advances in Human Aspects of Transportation: Part II*. Krakow, Poland: Applied Human Factors and Ergonomics (AHFE). 2014; 492-503.

[17] Taylor G, Miller J, Maddox J. Automating Simulation-Based Air Traffic Control. *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*. Nov/Dec 2005; 2913; Orlando, FL, USA.

[18] Zuluaga-Gómez JP, Prasad A, Nigmatulina I, Motlíček P, Kleinert M. A Virtual Simulation-Pilot Agent for Training of Air Traffic Controllers. *Aerospace*. 2023; 10(5).
DOI 10.3390/aerospace10050490

[19] Dow C, Histon J. An Air Traffic Control Simulation Fidelity Definition and Categorization System. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. 2014; 58(1), 92-96.

[20] Meyer D, Wypych T, Petrovic V, Strawson J, Kamat S, Kuester F. An air traffic control simulator for test and development of airspace management schemes. *IEEE Aerospace Conference;* Mar 2018; Big Sky, MT, 2018.

[21] Kulkarni VB. Intelligent air traffic controller simulation using artificial neural networks. *International Conference on Industrial Instrumentation and Control (ICIC)*. May 2015; Pune, India, 1027-1031.

[22] Alam S, Abbass HA, Barlow M. ATOMS: Air Traffic Operations and Management Simulator. *IEEE Transactions on Intelligent Transportation Systems*. 2008; 9, 2, 209-225. DOI 10.1109/TITS.2008.922877

[23] Williams KW, Christopher B, Drechsler G, Pruchnicki S, Rogers JA, Silverman E, Gildea KM, Burian BK, Cotton S. Aviation Human-in-the-Loop Simulation Studies: Experimental Planning, Design, and Data Management. 2014; Washington, DC, FAA, TR DOT/FAA/AM-14/1.

[24] Chhaya B, Jafer S, Coyne WB, Thigpen NC, Durak U. Enhancing Scenario-Centric Air Traffic Control Training. *AIAA Modeling and Simulation Technologies Conference*. Jan 2018, Kissimmee, FL, USA.

[25] Fothergill S, Loft S, Neal A. ATC-lab^{Advanced}: An air traffic control simulator with realism and control. *Behavior Research Methods.* 2009; 41, 1, 118-127.

[26] Hoekstra J, Ellerbroek J. BlueSky ATC Simulator Project: An Open Data and Open Source Approach. *7th International Conference on Research in Air Transportation (ICRAT)*. Jun 2016, Philadelphia, PA, USA.

[27] García Lasheras R, Albarrán J, Fabio A, Celorrio F, Pinto de Oliveira C, Bárcena C. Automatic Flight Callsign Identification on a Controller Working Position: Real-Time Simulation and Analysis of Operational Recordings. *Aerospace.* 2023; 10, 433.

[28] Fridyatama DAS, Suparji S, Sumbawati MS. Developing Air Traffic Control Simulator for Laboratory. *TEM Journal*. 2023; 12, 3, 1462-1474, UIKTEN.

[29] Fürstenau N, Jakobi J, Papenfuss A. Introduction: Basics, History, and Overview. In: Fürstenau N, editor. *Virtual and Remote Control Tower Research Topics in Aerospace*. Cham, Switzerland: Springer; 2022; 3-22.

[30] Radford A, Kim JW, Xu T, Brockman G, McLeavey C, Sutskever I. Robust Speech Recognition via Large-Scale Weak Supervision. 2022. [Online] http://arxiv.org/pdf/2212.04356v1.

[31] Baevski A, Schneider S, Auli M. vq-wav2vec: Self-Supervised Learning of Discrete Speech Representations. *Proceedings of the 8th International Conference on Learning Representations, ICLR 2020*, Addis Ababa, Ethiopia, Apr 2020.

[32] Zuluaga-Gomez JP, Prasad A, Nigmatulina I, Sarfjoo S, Motlicek P, Kleinert M, Helmke H, Ohneiser O, Zhan Q. How Does Pre-trained Wav2Vec2.0 Perform on Domain Shifted ASR? An Extensive Benchmark on Air Traffic Control Communications. *Proceedings of the IEEE Spoken Language Technology Workshop (SLT).* Doha, Qatar, Jan 2023.

[33] May M, Kleinert M, Helmke H. Automatic Transcription of Air Traffic Controller to Pilot Communication – Training Speech Recognition Models with the Open Source Toolkit CoquiSTT. *DLRK Congress*, Hamburg, Germany, Sep 2024.

[34] Kleinert M, Venkatarathinam N, Helmke H, Ohneiser O, Strake M, Fingscheidt T. Easy Adaptation of Speech Recognition to Different Air Traffic Control Environments using the DeepSpeech Engine. *11th SESAR Innovation Days*. Dec 2021; Virtual.

[35] Zuluaga-Gomez JP, et al. ATCO2 corpus: A Large-Scale Dataset for Research on Automatic Speech Recognition and Natural Language Understanding of Air Traffic Control Communications. DOI 10.48550/arXiv.2211.04054

[36] Chen S, Kopald H, Tarakan R, Anand G, Meyer K. Characterizing National Airspace System Operations Using Automated Voice Data Processing A Case Study Exploring Approach Procedure Utilization. *Research and Development Seminar*. Vienna, Austria, Jun 2019.

[37] EUROCONTROL. ATC Mobility and Capacity Shortfalls. Think Paper #19 – 19 December 2022. [Online] http://s.dlr.de/atcmobilityandcapacityshortfalls.