Journal of Open Aviation Science (2026), Vol.**4**
doi:10.59490/joas.2026.8462

JOAS

PROCEEDINGS | *The 13th OpenSky Symposium*

A Methodology for Quantifying Response Times for Deconfliction Actions Through ATC Communications

Timothé Krauth [*,1] Kim Gaume,[2] Xavier Olive [2] and Junzi Sun [3]

[1]Centre for Aviation, Zurich University of Applied Sciences, Winterthur, Switzerland
[2]ONERA DTIS, Université de Toulouse, France
[3]Faculty of Aerospace Engineering, Delft University of Technology, Delft, the Netherlands
*Corresponding author: timothe.krauth@zhaw.ch

(Received: 31 Oct 2025; Revised: 11 Feb 2026; Accepted: 12 Feb 2026; Published: 17 Feb 2026)
(Editor: Tatiana Polishchuk; Reviewers: Gabriel Jarry, Allan Tart, and Lucie Smetanová)

Abstract
The reaction time to a deconfliction situation refers to the interval between detecting a potential loss of separation and taking corrective action. For air traffic controllers, it represents the lead time between identifying a conflict and issuing a deconfliction instruction to a pilot. For pilots, it corresponds to the delay between receiving an ATC clearance and initiating the associated maneuver. Because both processes are influenced by human factors, these response times constitute a significant source of uncertainty in Air Traffic Management. While the controller's reaction time is particularly difficult to estimate since the exact moment at which a conflict is cognitively detected cannot be directly inferred from operational data, this paper focuses on quantifying pilot response times. To this end, we propose a methodology that combines Natural Language Processing techniques with surveillance and flight plan data. ATC–pilot voice communications are transcribed using a fine-tuned Automatic Speech Recognition model, and aircraft callsigns are identified through Named Entity Recognition. The transcriptions are then matched with corresponding flights in ADS-B surveillance data. Using flight plans, we identify lateral deconfliction maneuvers and align them temporally with the preceding ATC clearances to estimate the elapsed time between instruction and execution. Because the approach depends on a sequence of emerging algorithms whose robustness is still evolving, the study focuses on identifying the conditions under which the methodology performs reliably, highlighting its current limitations and associated data challenges, and proposing ways to overcome them.

Keywords: response time; air traffic controller audio; natural language processing; deconfliction actions extraction; ads-b data

Abbreviations: ASR: Automatic Speech Recognition; ATC: Air Traffic Control; MUAC: Maastricht Upper Area Control Centre; NER: Named Entity Recognition; NLP: Natural Language Processing; VAD: Voice Activity Detection

1. Introduction

The overall reaction time in Air Traffic Management is generally defined as the elapsed time between the detection of a potential conflict and the completion of the corresponding action to mitigate it. It can be decomposed into multiple components: (i) the air traffic control's (ATC) reaction time, defined as the interval between conflict detection and issuance of an instruction; (ii) the pilot's response time, defined as the interval between receipt of the controller's clearance and initiation of the maneuver; and (iii) the maneuver completion time, corresponding to the time between the ATC instruction and

full execution of the maneuver. Pilot response time is particularly critical: even small delays of a few seconds can lead to significant lateral or vertical deviations from the expected aircraft position after a maneuver [1]. Such delays also determine how much separation margin remains available to ATC before a conflict arises. Response times thus constitute an inherent source of uncertainty that directly impacts trajectory prediction accuracy, conflict detection, and ultimately, separation assurance.

Previous research has highlighted the operational consequences of such variability. For instance, [2] showed that increased variability in pilot response times degraded the accuracy of separation management. In current practice, controllers are trained to account both their own and the pilot's response times when issuing clearances, implicitly incorporating an expected delay margin. However, as Air Traffic Management evolves toward greater automation and AI-driven tools, particularly for trajectory prediction and strategic conflict resolution, a precise characterization of the distribution of global response times becomes indispensable.

Despite its operational importance, empirical quantification of response times remains limited. The challenge lies both in measurement, synchronizing communications with trajectory data, and in assembling sufficiently large datasets from real-world operations associated with ground truth. As a result, simplified assumptions are often applied in safety assessments. For example, ICAO's Safety Assessment Panel historically adopted conservative fixed values in their collision risk models, such as 30 seconds [3] for the pilot reaction time to a critical ATC instruction. With recent advances in Natural Language Processing (NLP) and growing access to synchronized positional and audio data, more precise measurements have become possible. For example, [4] combined ATCO–pilot audio with surveillance data to estimate maneuver initiation delays. This study demonstrates that while early work established basic timing distributions, new methodologies now enable a richer, more operationally relevant quantification of response times.

With a similar approach as [4], our study builds upon the outcomes of the ATCO2 project [5] and investigates the use of Automatic Speech Recognition (ASR) and Named Entity Recognition (NER) to quantify pilot response times to lateral deconfliction clearances. Our contribution lies in the integration of ATC communication broadcasting, NLP-based speech processing, surveillance data fusion, and trajectory action extraction to estimate the time interval between the instruction delivery and the initiation of the corresponding aircraft maneuver.

The proposed data pipeline proceeds as follows:

- **ATC–pilot audio acquisition**: we collect radio communications in the DELTA Sector of the Maastricht Upper Area Control Centre (MUAC).
- **Speech-to-text processing**: we transcribe audio streams into text with a fine-tuned version of the Whisper [6, 7] ASR model adapted for ATC communications.
- **Surveillance data alignment**: we collect ADS-B surveillance data from the OpenSky Network [8] for the same temporal and spatial ranges.
- **Callsign association**: we apply the ATCO2 NER model [9] to the transcripts to extract aircraft callsigns. Each detected callsign is cross-matched against active callsigns from the ADS-B feed at the corresponding time. This step yields a set of ATC communications linked to a flight in the surveillance dataset.
- **Trajectory action extraction**: we apply the lateral deconfliction extraction methods described in [10, 11] for flights with available flight plans. This enables the detection of trajectory modifications and their execution time due to lateral conflict-resolution instructions.
- **Response time estimation**: we estimate the execution delay required by the pilot to carry out

the instruction, given the time at which a clearance is issued and the moment the pilot initiates the corresponding deconfliction maneuver.

In this paper, we demonstrate the current capabilities of the proposed framework, identify its main limitations, and analyze the challenges related to both the robustness of individual tasks and the availability of supporting data. By examining representative scenarios, we expose the key technological barriers that currently prevent a large-scale analysis of response times and put forward concrete recommendations to address these issues.

In the following, Section 2 presents the current state-of-the-art in response time estimation. Then, Section 3 describes the data pipeline we propose to calculate the response time. Section 4 provides insights about the ability of our pipeline to identify pilot-ATC conversations, and describes specific situations highlighting the complexity of the problem at hand. Finally, Sections 5 and 6 respectively discuss the results and conclude the work.

2. Literature Review

The human-induced delays play a critical role in the conflict resolution timeline. Because these latencies consume precious seconds when resolving conflicts, underestimating them can decrease safety margins or force overly conservative buffer assumptions. Thus, any realistic conflict-detection or resolution tool must model those delays accurately.

In practice, separation minima and conflict-alerting thresholds are often derived assuming a worst-case controller and pilot latency. This assumption ensures that even slower-than-average responses leave enough buffer to avoid loss of separation. For instance, the International Federation of Air Traffic Controllers' Associations explicitly notes that the controller's intervention capability, including communication and pilot reaction latency, is a key determiner of how tight separation minima can be set [12]. Similarly, collision risk models such as those underlying ICAO standards, often include human reaction delay within the safety margins [13]. Because these human delays effectively reduce the time available for safe maneuvering, the more accurately we can characterize their distribution, the more precisely we can calibrate alert thresholds, separation buffers, and automation aids.

The first investigations into controller reaction latency date back to the late 1980s and early 1990s, primarily through human-in-the-loop simulations. With the introduction of automated conflict-alert systems in en-route ATC, researchers conducted experiments to measure the interval between an alert's onset and the controller's issuance of a verbal clearance. In one FAA simulation test of a prototype Conflict Resolution Advisory tool [14, 15], response latencies typically ranged from 12 to 18 s, with 95th-percentile values approaching 30 s. These studies showed that human decision-making and communication formulation can introduce non-negligible delays, particularly under high workload or traffic complexity.

Once a controller issues a clearance, the pilot's response can be divided into two phases: (1) acknowledgment through readback, and (2) initiation of the corresponding maneuver. One of the earliest empirical studies in this area is [16], which examined 64 hours of "time-critical" ATC messages recorded at U.S. Air Route Traffic Control Centers. The authors decomposed the interval from the start of the controller's instruction to the end of the pilot's correct readback, isolating the durations of the controller's speech, the pause before the pilot's reply, and the readback itself. Their findings showed that even under ideal conditions, the radio exchange alone accounts for a delay of several seconds.

However, pilot acknowledgment represents only part of the latency picture, as several seconds may elapse between the readback and the actual initiation of the maneuver. Recent research has leveraged both audio and surveillance data to pinpoint the moment when the aircraft trajectory first begins

to change following an ATC instruction. [4] developed an automated pipeline that analyzes spoken clearances and aligns them with tracking data. Their results indicate a mean pilot readback latency of approximately 0.6 s, but, more importantly, an average maneuver initiation delay ranging from 17 s to 25 s, depending on the clearance type (e.g., 17 s for heading and altitude changes, and 25 s for speed adjustments). They also reported corresponding maneuver completion times of about 69 s for heading changes, 176 s for altitude, and 182 s for speed. Building on this line of work, [1] used the large-scale SCAT dataset [17], comprising roughly 830,000 clearances, to predict pilot response delays. Their analysis revealed initiation delays of around 20–21 s, with a pronounced right tail. Moreover, while maneuver completion times were found to be relatively predictable based on the magnitude of change, initiation delays proved much harder to model accurately, suggesting that human factors introduce an element of intrinsic unpredictability. Overall, these empirical findings outline a consistent pattern: readbacks occur almost instantaneously (within about one second), maneuver initiation typically takes on the order of tens of seconds, and full completion may require several minutes, particularly for large altitude or speed changes.

With the development of NLP in the recent years, the methodological paradigm has shifted from hand-annotated voice recordings and small-scale simulations to automatic pipelines and large operational datasets to scale and refine response-time estimation. The application of NLP techniques to the analysis of ATC–pilot communications can be traced back to 2018 with the *Airbus Air Traffic Control Speech Recognition Challenge* [18]. In this context, participating teams demonstrated that, on a small-scale dataset, ASR and callsign detection could achieve strong performance, with word error rates below 8% for ASR and callsign-detection F1 scores exceeding 80%. These results should nonetheless be interpreted with caution, as the models were trained on approximately 40 hours of high-quality audio collected in a single, well-defined airspace. Even so, they indicate that automatic transcription of ATC communications can reach a level of accuracy sufficient to support downstream tasks such as response-time estimation.

The pipeline from [4] is an example: using ASR to transcribe ATC communications, they extract structured clearance data (callsigns, commands, values) through NER and synchronize them with surveillance tracks. From those results, they automatically compute readback latencies, initiation delays, and maneuver durations from real-world operations. However, the detection of pilot actions from surveillance data remains simple, based on a thresholding over the changes in altitude, heading and speed.

Another example in the domain of ATC communications analysis is [19]. The authors apply NLP techniques such as sentiment analysis, topic recognition and part-of-speech tagging, to categorize and distinguish miscommunications against regular messages in a corpus of transcribed communications. While their work is less focused on timing, it underscores how structured NLP can operate over communication corpora to reveal latent patterns in ATC speech behavior. [20] proposes a pipeline using ASR, and NER which extract relevant information from the communications to generate a spoken pilot response. Such systems not only assist in simulation and training of controllers, but also illustrate how voice-to-action mapping can be automated, which is directly relevant for measuring pilot response times in a real-time loop. Finally, [21] presents a real-time intelligent ATC system using ASR, natural language understanding and natural language response generation to transcribe, interpret and generate response to reduce pilot workload and improve communication efficiency.

The literature indicates that quantifying controller–pilot response times is critical for aviation safety. Whereas this was formerly a resource-intensive endeavor, often requiring human simulations and manual annotation, recent advances in NLP now enable the direct analysis of operational ATC recordings, substantially lowering the cost and effort of large-scale measurement.

3. Data and Methodology

This section presents our proposed methodology to evaluate the response time for lateral deconfliction instructions from ATC-pilot audio communications, ADS-B data and flight plans. The high-level data processing pipeline is described in Figure 1.

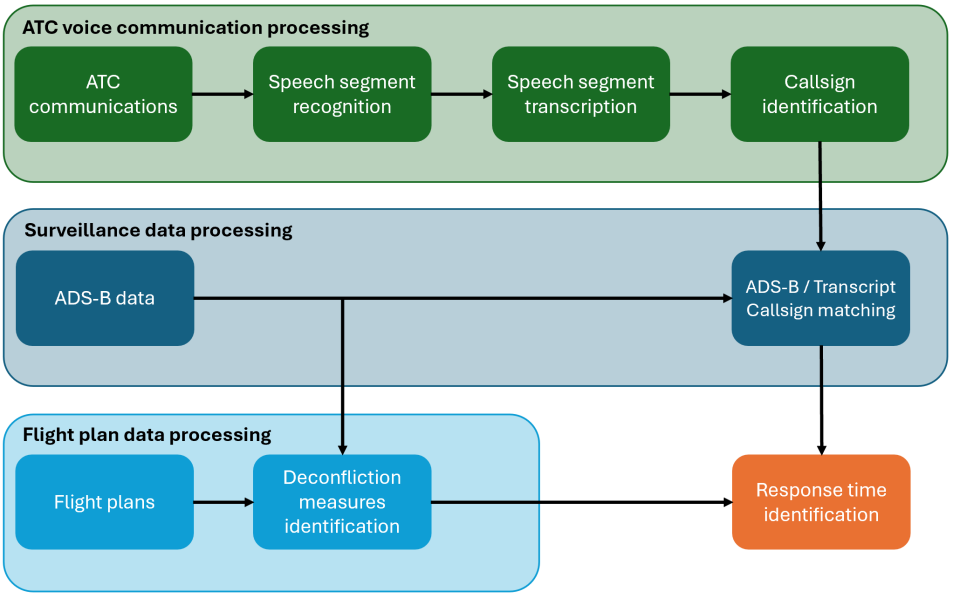

Figure 1. Data processing pipeline of ATC voice communications, surveillance data and flight plans

The pipeline is composed of three processing modules: (i) an NLP-based module that transcribes ATC voice communications and identifies spoken aircraft callsigns, (ii) an ADS-B trajectory analysis module that matches ATC communications with active flights, and (iii) a deconfliction maneuvers detection module that links those events with the corresponding ATC communication.

The proposed methodology shares conceptual similarities with the approach introduced in [4], as both aim to leverage an automated voice communication processing pipeline to analyze large volumes of ATC audio data with minimal human intervention. Accordingly, the overall structure of the data-processing pipeline relies on comparable core components: an ASR module, a callsign identification mechanism, and a track-based analysis module. Despite these similarities, the proposed methodology differs from [4] in several key aspects:

- Our speech-to-text module is based on an openly available model, enabling further performance improvements through domain-specific fine-tuning. This is particularly relevant for ANSPs that possess large volumes of high-quality ATC recordings, which can be leveraged to adapt the model to local acoustic and operational characteristics
- In [4], commands and callsigns are identified using rule-based detection of triggering keywords (e.g., "altitude" for altitude-change instructions, or airline telephony and the aeronautical alphabet for callsign identification). In contrast, our approach relies on a machine learning–based NER module trained within the ATCO2 project [5]. This design improves robustness to transcription errors and provides greater flexibility when handling non-standard or unconstrained phraseology used by controllers and pilots. In addition, we enhance the callsign–flight association step by replacing simple string-based matching (e.g., Levenshtein distance) with a semantic similarity–based

matching algorithm, which better accounts for phonetic and transcription discrepancies.

- While [4] identifies maneuver initiation and completion using rule-based thresholds applied directly to surveillance data (e.g., detecting altitude changes exceeding ±100 ft), our maneuver detection relies on a more sophisticated algorithm that compares observed trajectories against filed flight plans [22]. This approach enables a more precise identification of maneuver initiation times and reduces false positives due to weather conditions or inaccuracies in track data. The main trade-off is that the current implementation is limited to deconfliction maneuvers.

The remainder of this section provides a detailed description of each module composing the proposed methodology.

3.1 Speech-to-text processing

The first component is the speech-to-text stage. Using raw hourly ATC voice recordings, we apply the following steps:

- **Speech segmentation:** detect speech segments using Voice Activity Detection (VAD).
- **Timestamping:** anchor each detected segment to an absolute UTC time using the file start time and the segment's relative offset.
- **Transcription:** transcribe each speech segment with an ATC-tuned Whisper model, producing time-stamped text sequences.

3.1.1 Speech segment detection

To identify regions of speech within the ATC audio recordings, we employ the WebRTC VAD[1]. This lightweight, frame-based algorithm, originally developed by Google for real-time telecommunication systems, classifies short audio frames as either speech or non-speech using signal-derived features and an internal decision tree model optimized for low-latency voice detection.

The WebRTC VAD analyzes the input in fixed-length frames of 10, 20, or 30 milliseconds, processing each frame independently. For each segment, it computes spectral and energy-based features from the narrow-band audio and estimates the likelihood of speech presence. The aggressiveness parameter (ranging from 0 to 3) governs the balance between sensitivity and precision: lower values are more permissive, labeling ambiguous or noisy frames as speech, whereas higher settings reduce false positives at the risk of missing weak or low-intensity utterances. Because the model operates at the frame level, its raw output must be temporally smoothed to yield coherent speech segments. To achieve this, our implementation applies an additional aggregation layer that enhances temporal continuity and suppresses spurious detections.

Before processing, each audio file is standardized to meet the WebRTC VAD's expected conditions: it is converted to mono, resampled to 16 kHz, band-limited to the voice frequency range (150–3,800 Hz), and loudness-normalized to −20 dBFS. The waveform is then divided into 20 ms frames and passed through the VAD, which produces a binary sequence of speech and non-speech decisions. This sequence is smoothed using a hysteresis mechanism: speech onset is declared after approximately 60 ms of consecutive voiced frames (three frames), and speech offset after 120 ms of continuous silence (six frames). This strategy mitigates fragmentation caused by short pauses or noise bursts. Following smoothing, brief inter-speech gaps (< 200 ms) and short speech segments (< 300 ms) are discarded to eliminate residual artifacts. Each retained segment is then padded by 120 ms before and 150 ms after the detected boundaries, providing acoustic context for transcription.

Finally, each segment's onset time is mapped to absolute UTC timestamps by adding its local offset to

[1] https://github.com/wiseman/py-webrtcvad

the base timestamp of the source file. These timestamped speech intervals constitute the foundation for the subsequent transcription stage using the Whisper model.

3.1.2 Automatic speech recognition

To transcribe the detected speech segments into text, we employ a fine-tuned version of OpenAI's Whisper model [6]. Whisper is a transformer-based ASR system trained on 680,000 of hours of multilingual, multitask labeled audio data. Whisper performs end-to-end transcription: it converts raw audio waveforms directly into text without requiring an explicit phoneme model or language-specific decoder. Though it can be used on large chunks of audio, It is usually good practice to use Whisper after VAD for better performance.

Whisper uses an encoder–decoder transformer architecture. The encoder converts 16 kHz audio into a sequence of log-Mel spectrogram embeddings, while the decoder autoregressively predicts text tokens conditioned on these embeddings and its previous predictions. It is trained using a cross-entropy loss on transcribed speech segments, allowing it to model linguistic and acoustic dependencies jointly. The model operates on 80-channel log-Mel spectrograms computed from 25 ms windows with 10 ms stride, providing robust features even under noisy or clipped conditions. Because the model is fully sequence-to-sequence, it performs well on noisy, accented, or partially overlapping speech, making it suitable for ATC communications.

While the original Whisper model performs well on general speech, its performance on aviation phraseology and radio transmissions is limited. To address this, we use the ATC-Whisper model [7]. This model was fine-tuned specifically for air traffic control speech, achieving an 84% relative improvement in transcription accuracy compared to the base Whisper model. The fine-tuning process combined two curated ATCO corpora: the free version of the ATCO2 [5] dataset with 871 samples, and the UWB-ATCC dataset [23] comprising about 11,000 training annotated samples, and 3,000 test samples. During fine-tuning, [7] augmented the training dataset with Gaussian noise, pitch shifts, time stretching, and clipping distortion to simulate realistic radio conditions. According to [7], the fine-tuned model achieved a word error rate of 15.08% on the test ATC audio set, compared to the 94.59% for the original Whisper baseline.

The resulting output of the speech-to-text module is then a tuple that associates an absolute UTC timestamp with a transcribed sentence for each detected speech segment. These time-stamped transcriptions form the basis for the subsequent callsign recognition and ADS-B trajectory matching stages of the pipeline.

3.2 Callsign association

After the speech-to-text module, we associate the transcribed messages with aircraft that are active in the airspace at the same time through the use of surveillance data. Association proceeds in three stages:

- **NER detection in transcripts:** identify the tokens corresponding to callsigns in each transcribed message using a domain-adapted NER model.
- **Normalization of ADS–B callsigns:** convert each alphanumeric ADS–B callsign into its spoken radiotelephony form (airline telephony, NATO letters, digit words).
- **Time-gated semantic matching:** for each message, the detected spoken callsigns are compared with candidate ADS–B callsigns active within the same temporal window. The best-scoring match is retained if its similarity score exceeds a predefined threshold; otherwise, the utterance remains unmatched.

3.2.1 Named-entity recognition for callsigns

Spoken callsigns directly detected in the transcribed text with a domain-adapted NER model developed in the ATCO2 context [9]. The model is a BERT-base token classifier fine-tuned for ATC communications. Given a sentence, it assigns BIO-style labels over subword tokens for three semantic classes: CALLSIGN (often airline telephony plus alphanumerics), COMMAND (e.g., "cleared to land"), and VALUE (e.g., "runway three four left"), together with confidence scores. Because callsigns are frequently fragmented across adjacent sub-entities (e.g., "seven nine | whiskey | uniform"), we apply a post-processing step that merges consecutive CALLSIGN entities into a single span, assuming that each speech contains only one callsign. The output per utterance is a (possibly empty) list of detected callsign strings expressed in spoken form.

3.2.2 Time-gated candidate selection in ADS–B

ADS–B callsigns are alphanumeric (e.g., `EXS79WU`), whereas NER-detected callsigns contain spoken radiotelephony (e.g., "*channex seven nine whiskey uniform*"). To compare those, we map each ADS–B callsign to a *spoken* form:

- airline ICAO prefix → radiotelephony (e.g., `EXS` → *channex*, `DLH` → *lufthansa*);
- letters → NATO alphabet (e.g., `W` → *whiskey*, `U` → *uniform*);
- digits → number words (e.g., 7 → *seven*, 9 → *nine*).

To narrow the search space, we precompute for each ADS–B callsign c its active interval, denoted as $[t_{\min}(c), t_{\max}(c)]$, corresponding to the earliest and latest observation times in the surveillance data. Then, for a communication occurring at time t_{comm}, we retain as candidates only those callsigns whose active intervals overlap with a predefined tolerance window centered on the utterance time:

$$\big[t_{\min}(c),\ t_{\max}(c)\big] \cap \big[t_{\text{comm}} - \Delta,\ t_{\text{comm}} + \Delta\big] \neq \emptyset,$$

where Δ is a small buffer (typically 60 s). This buffer is introduced because communications might happen before entering of after exiting the airspace.

3.2.3 Semantic matching in the spoken domain

Once a callsign has been detected in the ATC transcript and a time-gated set of ADS–B candidates has been assembled, we must associate the most plausible ADS–B callsign to the utterance. Exact string matches are often unreliable due to (i) imperfect speech segmentation, (ii) ASR errors, (iii) NER mislabeling or span fragmentation, (iv) controllers using partial or shortened callsigns and variable phraseology, and (v) incomplete airline telephony mappings. To address this, we use a *soft-matching* procedure in the spoken domain based on sentence embeddings and cosine similarity.

For each utterance with one or more NER callsign strings $\{q_j\}$ and its time-gated ADS–B candidate set $\{c_i\}$ (each converted to spoken radiotelephony), we compute a semantic similarity as follows:

- Encode all candidate spoken forms $\{c_i\}$ with *MiniLM-L6-v2* to obtain vectors $\phi(c_i) \in \mathbb{R}^d$.
- For each detected NER callsign q_j, encode $\phi(q_j)$ and compute cosine similarities $s_{ij} = \cos\big(\phi(q_j),\ \phi(c_i)\big)$.
- Select the best candidate $i^* = \arg\max_i s_{ij}$ and accept the association if $s_{i^*j} \geq \tau$ (default $\tau = 0.7$); otherwise the utterance remains unmatched.

We report the winning ADS–B callsign, the detected spoken string, the similarity score, and the original transcript timestamp. This spoken-space, semantic approach is robust to common ASR variations (e.g., "x-ray" vs. "xray"), partial insertions/omissions, and minor lexical drift, while the time gate suppresses false positives from unrelated traffic.

3.3 Extracting aircraft conflict-resolution actions

To identify instances where controllers likely issued deconfliction instructions, we applied the extraction algorithm introduced in [11]. The method detects lateral deviations in aircraft trajectories and evaluates whether these deviations were performed to avoid potential conflicts with surrounding traffic.

The algorithm first isolates trajectory segments that deviate from the original flight plan. Each detected deviation is then analysed within its local traffic context: surrounding aircraft at compatible flight levels and within a relevant time window are retrieved, and for each potential encounter, the observed lateral distance at the closest point of approach (CPA) with surrounding aircraft d_{min} is compared with the predicted distance $\widehat{d}_{min}$ obtained from a reconstructed trajectory assuming the deviation had not occurred.

A deviation is classified as a deconfliction when it satisfies the following:

- the deviation increases the minimum separation: $d_{min} - \widehat{d}_{min} > \varepsilon$,
- the time between the start of the maneuver and the predicted CPA t_{CPA} is above a threshold T_{CPA},
- two or more aircraft were at risk of loss of separation, such that $\widehat{d}_{min} < D$.

The thresholds ε, T_{CPA} and D are determined through statistical analysis of historical deviations.

Compared with [4], which treats the execution of an ATC instruction as a simple deviation from the nominal aircraft trajectory, the proposed methodology enables a more detailed characterization of the deconfliction maneuver actually performed by the aircraft. In particular, it can discriminate such maneuvers from other sources of trajectory deviation, including routine turns between successive route segments or persistent offsets caused by meteorological conditions.

4. Results

This section evaluates the performance of our data pipeline in accurately identifying ATC–pilot conversations for each flight, thereby enabling reliable computation of reaction times. Section 4.1 first presents the dataset used for evaluation, selected as a trade-off between acquisition time and the number of exploitable operational days. Section 4.2 then illustrates one minute of transcribed communication, exemplifying both the strengths of the approach and its potential limitations. Next, Section 4.3 introduces quantitative metrics that assess how effectively the pipeline reconstructs complete conversation threads between ATC and individual flight crews. Finally, Section 4.4 examines a series of situations in which deconfliction measures were identified, highlighting the improvements required to achieve fully automated and reliable response-time computation.

4.1 Data

The proposed methodology was tested on ATC communications from the MUAC DELTA Sector, shown in Figure 2, covering the period from 25 August to 17 September (UTC).

The MUAC airspace offers a particularly suitable testbed: it is legally permissible to collect and process ATC audio in this region; recording quality is relatively high; and, as an en-route sector with available flight plans, it places the deconfliction-event detection algorithm [11] in its intended operating regime.

The radio communication between air traffic controllers and pilots operates make use of three different VHF frequencies:

- 135.958 MHz for DELTA Low (FL245 to FL335)

- 135.508 MHz for DELTA Middle (FL335 to FL365)
- 132.083 MHz for DELTA High (FL365 and above)

The VHF receiver is set up on the roof of the Faculty of Aerospace Engineering at TU Delft. The VHF signal is collected using a `RSPdx-R2` software-defined radio. Each hour of audio for a single frequency is saved in a single `MP3` format audio file.

At the start of the data collection, the collected audio contains three frequencies at different times. However, towards the later stage of the experiment, we focus primarily on DELTA High, aiming to capture as much communication as possible on 132.083 MHz, as it contains only cruise flights.

For the surveillance context, ADS–B tracks were retrieved from the OpenSky Network [8] via the `traffic` Python library [24], extracting trajectories over the MUAC sector with an additional ~50 km spatial buffer for the same time span as the audio recordings. Days with missing audio or unavailable ADS–B data were excluded from analysis (5–8 September and 13 September).

In addition to the surveillance data, flight plan information (field 15) was incorporated to describe the planned trajectories of the identified flights. Each plan includes the sequence of navigation aids (navaids) defining the intended route through the MUAC sector.

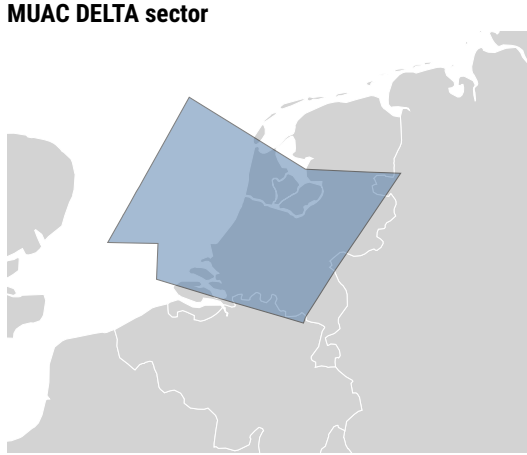

Figure 2. MUAC DELTA sector

4.2 Example of transcribed conversation

Table 1 presents an excerpt of ATC communication transcriptions from 31 August 2025, covering the interval between `23:58:55` and `23:59:29`. *Callsign (spoken)* denotes the callsign detected by the NER algorithm in the transcriptions, while *Callsign (ADS–B)* refers to the callsign present in the ADS–B data that was associated with the same utterance. The *Matched* flag indicates whether a transmission was successfully linked to an ADS–B callsign, and the *Score* column reports the cosine similarity between the embeddings of the NER-extracted and ADS–B callsigns.

This conversation thread is representative of the typical challenges encountered in the matching process. The overall audio quality in this excerpt is relatively high, resulting in mostly accurate transcriptions. However, the transmission at `23:58:58` illustrates a limitation of the VAD: a slight hesitation at the start of the recording led the VAD to trim the initial segment, causing a mis-transcription ("navigator") and a subsequent misidentification of the callsign by the NER model. Fortunately, the soft-matching algorithm we developed is resilient to such localized errors and was still able to con-

fidently associate the transcription with the correct ADS–B callsign. A similar trimming artifact appears to affect the transmission at `23:59:27`.

The communication at `23:59:19` poses another difficulty. Because it is short and partially unintelligible, the transcription likely contains an incorrect word (“austrian”), which in turn disrupted callsign recognition by the NER. Nevertheless, the soft-matching mechanism mitigates the impact of such misclassifications as long as the detected sequence remains reasonably close to the true callsign.

Finally, the communication at `23:59:29` illustrate another common situation: neither the pilot nor the controller explicitly states the callsign when the intended recipient is contextually obvious. In such cases, the system cannot automatically associate the message with a specific flight, underscoring a fundamental limitation of automated conversation reconstruction.

Table 1. Callsign matching results for sample transcripts (UTC).

Time (UTC)	Callsign (ADS-B)	Callsign (spoken)	Score	Matched
`2025-08-31 23:58:58Z`	EXS6AH	navigator six alfa hotel	0.75	True
	Transcript: navigator six alfa hotel we re just getting light continuous traffic three six zero here have you got any report for further on route			
`2025-08-31 23:59:07Z`	EXS6AH	channex six alfa hotel	1.00	True
	Transcript: channex six alfa hotel earlier we had report of light navigational moderate bump on your routing between level three one and three eight zero but standby second			
`2025-08-31 23:59:19Z`	EXS79WU	channex seven nine whiskey uniform austrian	0.90	True
	Transcript: channex seven nine whiskey uniform austrian			
`2025-08-31 23:59:23Z`	EXS79WU	channex seven nine whiskey uniform	1.00	True
	Transcript: channex seven nine whiskey uniform go ahead			
`2025-08-31 23:59:27Z`	EXS79WU	yes seven nine whiskey uniform	0.83	True
	Transcript: yes seven nine whiskey uniform any turbulence			
`2025-08-31 23:59:29Z`	—	—	—	False
	Transcript: yeah we have just experienced occasional moderate currently smooth occasional light			

4.3 Performance evaluation of conversation identification

The response time is calculated as the interval between an ATC instruction and the actual movement of the aircraft. In order to reliably identify those, the pipeline must be capable of retrieving all utterances belonging to a conversation between a controller and a specific pilot. In practice, pilot–controller response times can only be computed if every transmission associated with the same aircraft is consistently recovered; missing associations may render reaction time estimation impossible. Two components are therefore critical: the accurate detection of CALLSIGN spans in transcripts via NER, and the robust association of these spans with ADS–B callsigns active within the same temporal window. This section presents descriptive statistics summarizing daily communication transcriptions and evaluating the quality and stability of the callsign-matching process.

Table 2 summarizes daily activity, including the number of transcribed communications, the number of retrieved ADS–B trajectories, and the proportion of communications successfully associated with

an ADS–B callsign. The daily communication load is generally high but varies substantially (median ≈ 6,035 communications; IQR: 4,777–6,795), with a comparable magnitude of observed flights (median ≈ 5,258). The notable difference between the mean and median number of flights suggests that a few days exhibit unusually low traffic counts. Given that MUAC is a centrally located en-route sector where overflight volumes are typically stable, such deviations reflect acquisition gaps in either the audio recordings or the ADS–B surveillance data. Moreover, the similar magnitudes of communication and flight counts indicate that while a substantial portion of transmissions are captured, many are still missed. Under routine operational conditions, at least four transmissions per flight would typically be expected: an instruction and readback for both entry into and exit from the sector. It implies that only a fraction of total communications are currently recovered.

The callsign detection and matching results confirm these observations. The NER model successfully identifies a callsign in approximately 80% of all utterances; however, only a subset of these can be linked to a concrete ADS–B callsign. The median *match rate* at a similarity threshold of 0.7 is ≈ 0.441 (IQR: 0.390–0.475), meaning that roughly two in five utterances are confidently associated with an ADS–B aircraft callsign. The *perfect match* rate is lower (median ≈ 0.105), as expected given clipped audio segments from the VAD, imperfections in ASR and NER outputs, the frequent use of partial or shortened callsigns in radio communications, and occasional inconsistencies in airline telephony mappings. These limitations justify the use of a *scored* association approach rather than strict string matching.

Conditional on acceptance, similarity scores are tightly concentrated (median ≈ 0.853; IQR: 0.840–0.862), indicating that once a candidate passes the time-gating and thresholding stages, the semantic evidence supporting the match is both strong and consistent across days.

Table 2. Daily communications, flights, and callsign–ADS-B matching summary. Means/medians are per day; Q1/Q3 are the 25th/75th percentiles. Rates are proportions in $[0, 1]$.

Metric	Mean	Median	Q1 (25%)	Q3 (75%)
Communications per day				
Count	5,624.6	6,035.0	4,776.5	6,795.0
Flights per day				
Count	4,661.53	5,258.0	5,040.0	5,447.0
NER callsign detection				
Proportion	0.787	0.804	0.756	0.830
Match rate (score ≥ 0.7)				
Proportion	0.398	0.441	0.390	0.475
Perfect match rate				
Proportion	0.090	0.105	0.059	0.126
Match score (conditioned on score ≥ 0.7)				
Score	0.851	0.853	0.840	0.862

Day-to-day reliability is assessed through the fraction of utterances whose detected CALLSIGN could be associated with an ADS–B callsign using the matching algorithm, with a score threshold of $\tau = 0.7$, as shown in Figure 3. Sharp drops in matching performance on 10 September 2025 and 15 September 2025 correspond to the data acquisition issues identified earlier. On these days, only a limited number of aircraft were captured in the ADS–B dataset, which led to a marked decrease in the overall match rate.

These results demonstrate that the pipeline is naturally sensitive to input data quality; however, when this condition is satisfied, matching performance remains consistent across days. Typical coverage ranges between 35% and 55%, indicating that roughly two in five transcriptions can be confidently linked to a specific flight under the current selection method. While lowering the score threshold could increase the match rate, it would also raise the risk of incorrect associations. In its present configuration, a threshold of $\tau = 0.7$ appears to provide a reliable balance between precision and coverage.

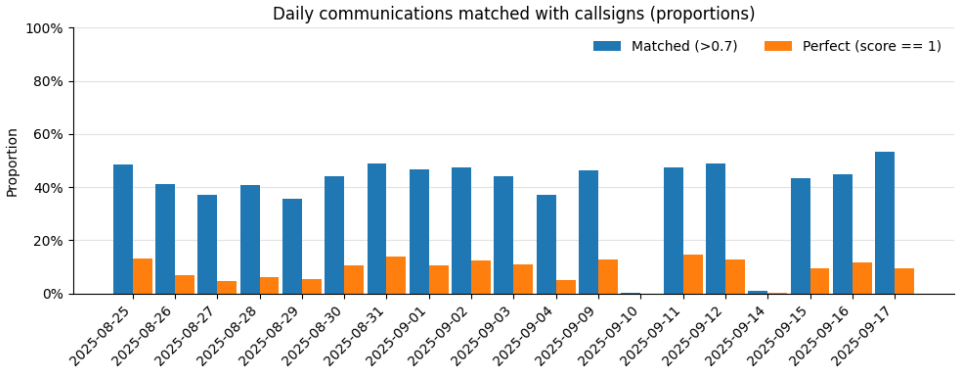

Figure 3. Proportion of ATC transcription that were matched with callsigns from surveillance data. A match is considered valid only if the matching score between the callsign detected in the transcription and the candidate callsign from surveillance data is above 0.7.

Figures 4 and 5 characterize how effectively the pipeline reconstructs complete pilot–controller conversations for individual flights by illustrating the distribution of the number of communications assigned to each callsign. As shown in Figure 4, most conversations comprise between one and seven communications, and only a few exceed eleven. This distribution is consistent with operational expectations, as most exchanges in en-route sectors are typically short. However, the large number of callsigns associated with only a single communication indicates that, while callsign–communication matches are generally accurate when they occur (high similarity scores), a substantial proportion of transmissions remain unlinked.

Figure 5 provides a more detailed view by reporting the daily average number of communications per callsign, along with the interquartile range and extreme values. A few callsigns are linked to more than 20 communications—reaching up to 50 in rare cases—which likely reflects erroneous associations, often arising when two aircraft with similar telephony callsigns operate concurrently within the same airspace. On average, each callsign is associated with approximately 2.5 communications (third quartile around 3), which remains below operational expectations: for a typical overflight, at least four transmissions are anticipated—two during sector entry and two during exit. This further underscores the current pipeline's difficulty in reconstructing complete conversation threads.

4.4 Pilot-ATCO conversation analysis

This section analyzes three situations that were identified as conflict-resolution events. The objective of this analysis is to compute pilot response times for these maneuvers and to highlight the current methodological limitations that hinder large-scale response time estimation.

Figure 6 illustrates a lateral deconfliction maneuver between flights `RYR23LB` and `IBE07FE` that occurred on 31 August 2025. Focusing on flight `RYR23LB`, two distinct deviations from its planned route were detected: the first at 19:38:54, corresponding to a pilot-initiated heading change, and the second at 18:44:53, when the aircraft resumed navigation toward the `BODSO` waypoint. The transcribed

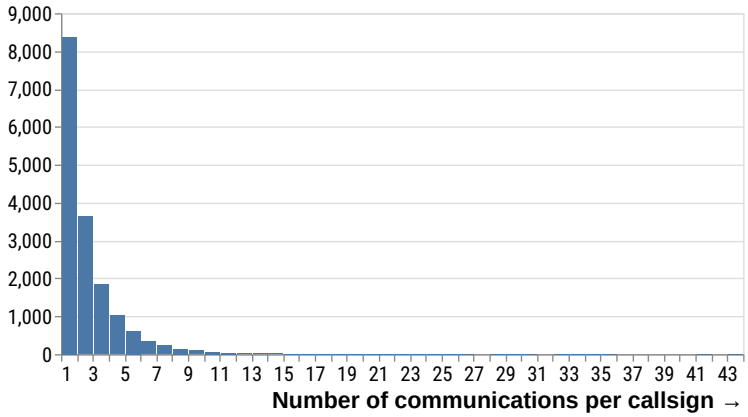

Figure 4. Distribution of the number of communications assigned to each callsign. For instance, more than 8,000 callsigns are linked to only one voice audio, and rougly 2,000 callsigns are associated to a thread of 3 voice audios. Uneven numbers of communications indicate under-threading, meaning that though a decent proportion of communications are matched, the algorithm struggled to capture both sides of a short exchange (instruction + readback).

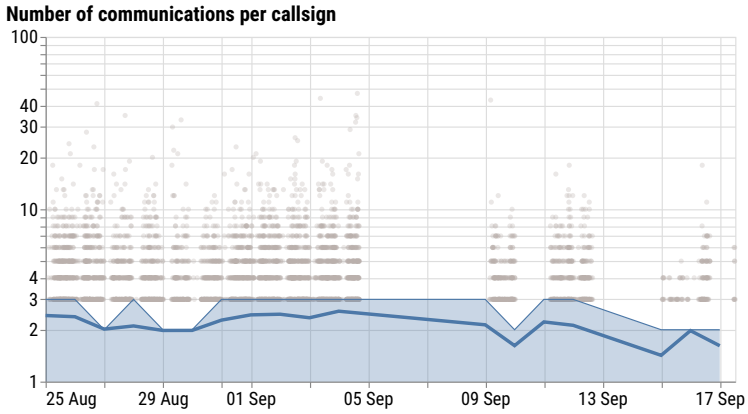

Figure 5. Scatter plot of the number of communications per callsign. The dark blue line shows the average number of communications per day, and the shaded area the inter-quartile range. Days without data points (5–8 September and 13 September) indicate periods of VHF signal loss or missing ADS–B data.

ATC–pilot communications generated by the proposed methodology are shown alongside their corresponding timestamps. Based on these, the pilot's reaction times to the ATC instructions were estimated as approximately **20 s** for initiating the first diversion and **39 s** for resuming the planned route.

Several shortcomings were, however, identified. First, the pilot's readback of the initial instruction was mistranscribed: "two seven five" was incorrectly rendered as "csa five". In addition, the matching algorithm failed to associate the pilot's readback of the second instruction with flight RYR23LB, resulting in a missing segment in the reconstructed conversation. Both issues likely stem from poor audio quality in the cockpit recordings, which is noisy. Finally, the ATC's second instruction was also mistranscribed: "tango taxi" was likely not part of the clearance, and the fix name was misidentified as ODNOK instead of BODSO. This error is less related to signal quality and more to the clarity of the controller's speech, which was difficult to interpret in this instance.

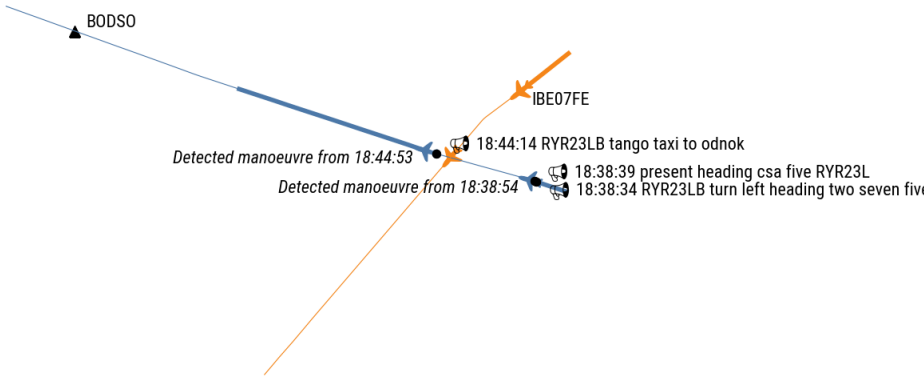

Figure 6. Two detected maneuvers for aircraft RYR23LB.

Figure 7 illustrates a lateral deconfliction maneuver between flights RYR22UZ and DLH7K that occurred on 4 September 2025. At 12:11:12, ATC instructed flight RYR22UZ to initiate a right turn, and the pilot acknowledged the clearance three seconds later, at 12:11:15. The maneuver began at 12:11:34. The resulting readback time is therefore **3 s**, and the pilot's response time to initiate the maneuver is **22 s**, which aligns well with findings from previous studies.

In this example, the conversation thread appears to have been correctly reconstructed by the algorithm, and the pilot's readback accurately transcribed. However, the ATC instruction itself was not fully decoded: the transcription indicates a heading toward VEMUT, whereas this waypoint does not correspond to the intended direction.

Figure 8 illustrates a lateral deconfliction maneuver between flights AFR53XE and EZY69HP that occurred on 31 August 2025. At 13:52:34, ATC instructed flight AFR53XE to maintain its current heading, despite a turn being scheduled in the flight plan. The corresponding maneuver was detected at 13:54:04. In this case, estimating the pilot's response time is challenging, since no immediate action was required. Instead, we can infer that the controller issued the "continue heading" instruction approximately **90 s** before the expected turn, reflecting a measure of the controller's anticipation or lead time rather than the pilot's reaction time.

This example also reveals several limitations of the current pipeline. The pilot's readback, though clearly audible in the recording, was not captured by the transcription module. The segment was correctly detected by the VAD, but the speech-to-text model failed to recognize the portion contain-

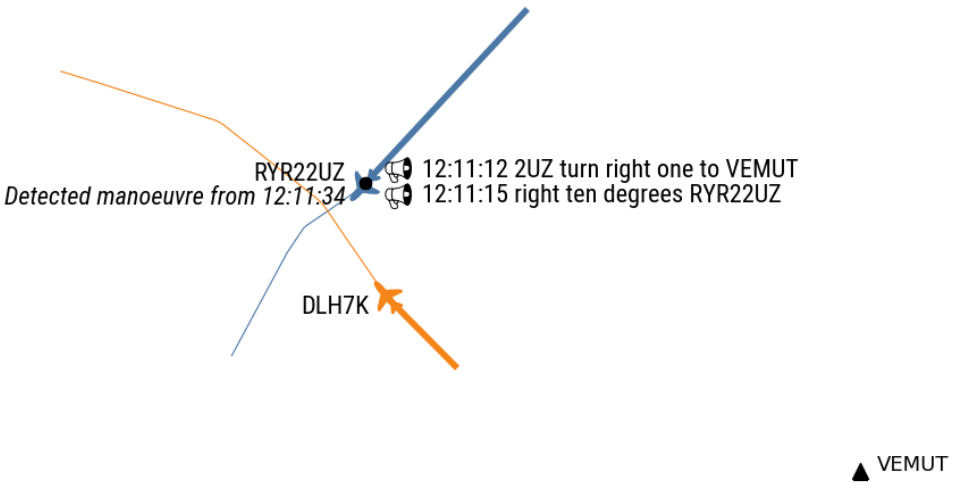

Figure 7. One detected maneuver for aircraft RYR22UZ

ing the callsign due to strong echo in the audio, preventing successful callsign attribution. Similarly, the callsign in the ATC instruction itself was imperfectly transcribed. Nevertheless, the matching algorithm still associated the communication with AFR53XE, thanks to sufficient semantic similarity to the ground truth.

Finally, the last exchange in the sequence was also incorrectly processed. In the audio, the pilot requested confirmation of the "continue heading" instruction at 13:56:24, but this transmission was missed because of echo distortion. The controller's subsequent reply was captured, but the audio was poor and largely unintelligible. The pilot then responded without repeating the callsign, leading to a missed association.

5. Discussion

The proposed methodology shows strong potential for enabling large-scale processing and analysis of ATC audio communications. While its immediate application lies in estimating pilot response times, it can be extended to a broader range of human-factors analyses in air traffic management: for example, assessing controller lead times between conflict detection and instruction issuance, investigating loss of attention due to excessive communication frequency, or analyzing instances of ambiguous or non-standard phraseology. Nevertheless, as highlighted in Section 4, several limitations currently prevent the methodology from being deployed in a fully automated, large-scale setting.

A first challenge concerns data quality, since the pipeline relies on raw, openly available sources. During our acquisition campaign, we encountered frequent disruptions either in the audio recording setup or in the retrieval of ADS–B trajectories, resulting in missing or incomplete data. Such issues directly affect the ability to assign accurate callsigns to transcriptions, as illustrated in Figure 3. Moreover, the audio quality itself is often suboptimal, particularly for pilot transmissions, which are typically affected by strong cockpit background noise. Poor signal quality degrades speech-to-text transcription performance and, consequently, the accuracy of callsign matching. Another recurring difficulty is the variability of radio phraseology: callsigns are sometimes omitted or only partially pronounced. In partial cases, the algorithm may still recover the correct callsign, though occasionally

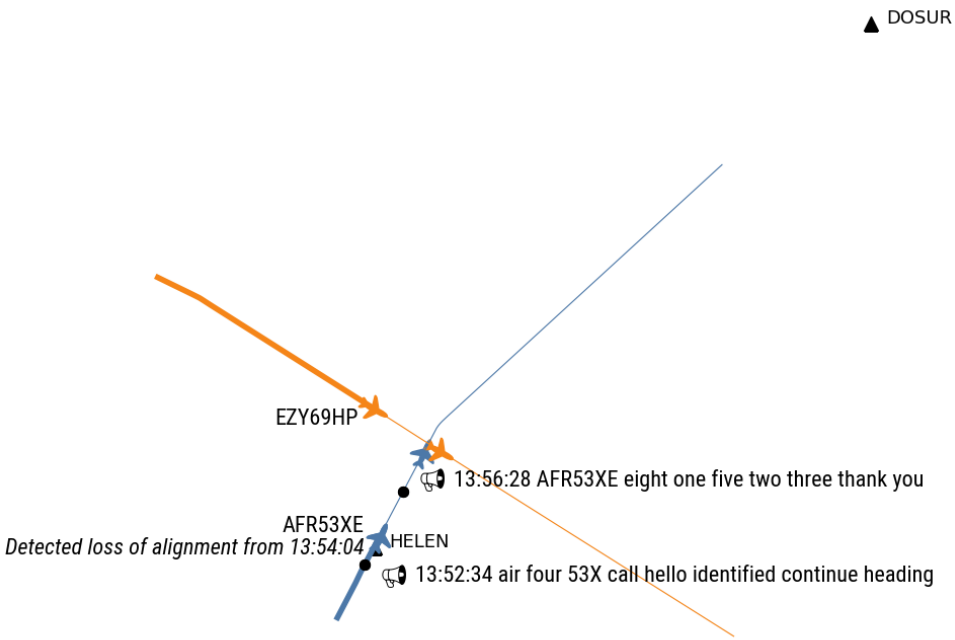

Figure 8. One detected "heading continuation" for aircraft AFR53XE

it misattributes it; in the absence of any callsign mention, however, conversations cannot be reconstructed, as each utterance is treated independently without contextual linkage between controller and pilot turns.

A second limitation arises from error propagation across the pipeline's modular structure. Although safeguards and post-processing steps are implemented at each stage (e.g., VAD smoothing, merging of consecutive NER callsigns, soft-matching of candidate flights), errors in early modules can cascade through subsequent components. For example, an overly restrictive VAD segmentation can truncate voice segments, removing critical acoustic context and thus degrading transcription quality. This behavior was observed in Table 1, where a clipped audio segment led to incorrect callsign recognition. Similar propagation effects occur when the speech-to-text or NER modules fail, ultimately resulting in mis-transcribed communications or incorrect ADS–B associations.

Finally, identifying deconfliction maneuvers remains a particularly difficult task. Access to recent and detailed flight plans is limited, and aircraft often deviate from their planned routes for operational reasons unrelated to conflict resolution. Furthermore, the current implementation focuses solely on lateral deconfliction maneuvers, which are less frequent than altitude or speed adjustments. Consequently, fewer than 1% of flights are flagged as potential deconflictions. Combined with the fact that only about 40% of communications are matched, the probability of obtaining a usable ATC–pilot exchange for a deconflicted flight remains low. Even among these, missing speech segments within an ATC-pilot conversation can prevent accurate response time estimation. As a result, manual inspection is still required to verify the completeness and usability of each identified case, which currently prevents fully automated computation of response times at scale, and therefore the computation of the distribution of the response times for a large number of communications. It is also important to note that the current maneuver identification algorithm is not applicable to free-route airspaces, where no predefined flight routes exist. Such airspace structures are becoming

increasingly common in Europe.

6. Conclusion and outlook

This study introduced a methodology for the automatic processing of ATC voice communications by integrating speech processing, natural language understanding, and surveillance data. The proposed pipeline was primarily applied to estimate pilot–controller response times, but it can readily be extended to other tasks that currently require manual annotation of audio recordings. The MUAC airspace was selected as a test environment because it legally permits the recording and processing of ATC communications and provides access to flight plans—conditions that enable the application of the deconfliction detection algorithm for quantifying pilot response times. More broadly, the speech-to-flight association pipeline can be deployed in any airspace where both ATC audio and surveillance data are available. Minor adaptations to the callsign-matching algorithm may, however, be necessary to account for regional variations in radio phraseology across different continents.

The results demonstrate that the proposed pipeline can reliably identify and align ATC–pilot exchanges under realistic operational conditions. When data quality is sufficient, the system achieves consistent matching performance and produces response time estimates in line with empirical findings from earlier simulation-based studies.

However, several challenges remain before the framework can be deployed for large-scale, fully automated analyses. Chief among them is the dependence on high-quality input data: reliable audio recordings and complete surveillance coverage are prerequisites for accurate conversation reconstruction. Furthermore, the current callsign-matching algorithm treats each utterance independently, overlooking the temporal continuity that naturally exists in ATC dialogues. Incorporating conversational context such as question–answer or instruction–readback sequences, would reduce the number of missing associations within a thread, and improve readback detection.

Future work will also focus on improving the transcription stage by refining or fine-tuning the Whisper ATC model. Training on a larger and more diverse set of annotated ATC communications, or performing domain-specific fine-tuning for individual centers or frequency bands, could substantially enhance recognition accuracy. Finally, extending the deconfliction detection algorithm beyond lateral maneuvers to include speed and altitude adjustments will broaden the range of detectable events and enable a more comprehensive analysis of response behavior. Adapting the algorithm to free-route airspaces also represents a natural next step toward broader applicability across different operational environments.

Overall, this work represents a step further toward a scalable framework for the data-driven assessment of human response times in air traffic control operations. With continued improvements in data quality, model robustness, and event-detection capabilities, the proposed methodology has the potential to support large-scale, real-world evaluations of controller and pilot performance.

While constraints on ATC audio data quality and availability currently limit large-scale reproducibility in airspaces governed by stricter privacy regulations (e.g., France), this work presents opportunities for operational stakeholders. In particular, the pipeline could be highly generalizable if deployed directly by ANSPs, which legally possess access to high-quality ATC voice communications. Such stakeholders could implement the methodology internally to monitor systemic safety performance (for example, through sector complexity or workload analyses) without encountering the regulatory barriers faced by open-source researchers. This would, however, require the establishment of appropriate governance frameworks, such as Just Culture principles and data anonymization procedures, to address legitimate concerns related to individual performance monitoring.

Author contributions

Conceptualization (T.K), Methodology (*all*), Software (T.K, K.G), Validation (T.K), Formal analysis (*all*), Investigation (*all*), Data Curation (*all*), Writing – Original Draft (*all*), Writing – Review & Editing (*all*), Visualization (T.K, X.O), Project administration (T.K), Funding acquisition (T.K)

Open data statement

The data is available at https://doi.org/10.5281/zenodo.18611465

Reproducibility statement

The code is available at https://github.com/kruuZHAW/atc_clearances

References

[1] Niclas Wüstenbecker, Justus Renkhoff, Dagh Zeppenfeld, Mohsan Jameel, and Sebastian Schier-Morgenthal. "Analysis and prediction of pilot response time to air traffic control clearances". In: *CEAS Aeronautical Journal* (2025). DOI: 10.1007/s13272-025-00848-9.

[2] Esa M Rantanen, Jason S McCarley, and Xidong Xu. "Time delays in air traffic control communication loop: effect on controller performance and workload". In: *The International Journal of Aviation Psychology* (2004). DOI: 10.1207/s15327108ijap1404_3.

[3] International Civil Aviation Organization (ICAO). *Global Operational Data Link Document (GOLD), Second Edition*. Tech. Rep. Second Edition. Available online: https://www.caa.co.uk/media/2cdpufa4/gold_2edition.pdf. ICAO / GOLD ad hoc Working Group, Apr. 2013. URL: https://www.caa.co.uk/media/2cdpufa4/gold_2edition.pdf.

[4] Michael Lutz, Gano Broto Chatterji, and Husni R Idris. "Characterization of response times based on voice communication and traffic surveillance data". In: *Proceedings of the AIAA AVIATION 2022 Forum*. 2022. DOI: 10.2514/6.2022-3762.

[5] Juan Zuluaga-Gomez, Karel Veselỳ, Igor Szöke, Petr Motlicek, et al. *ATCO2 corpus: A Large-Scale Dataset for Research on Automatic Speech Recognition and Natural Language Understanding of Air Traffic Control Communications*. 2022. DOI: 10.48550/arXiv.2211.04054.

[6] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. "Robust speech recognition via large-scale weak supervision". In: *Proceedings of the 40th International Conference on Machine Learning*. 2023. DOI: 10.48550/arXiv.2212.04356.

[7] Jack Tol. *Fine-Tuning Whisper for Air Traffic Control: 84% Improvement in Transcription Accuracy*. Accessed: February 17, 2026. Oct. 9, 2024. URL: https://jacktol.net/posts/fine-tuning_whisper_for_atc/.

[8] Matthias Schäfer, Martin Strohmeier, Vincent Lenders, Ivan Martinovic, and Matthias Wilhelm. "Bringing up OpenSky: A Large-Scale ADS-B Sensor Network for Research". In: *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. 2014. DOI: 10.1109/ipsn.2014.6846743.

[9] Juan Zuluaga-Gomez, Seyyed Saeed Sarfjoo, Amrutha Prasad, et al. "BERTraffic: BERT-based Joint Speaker Role and Speaker Change Detection for Air Traffic Control Communications". In: *IEEE Spoken Language Technology Workshop (SLT), Doha, Qatar* (2022). DOI: 10.1109/SLT54892.2023.10022718.

[10] Kim Gaume, Xavier Olive, David Gianazza, Richard Alligier, and Nicolas Durand. "A Catalogue of Deconfliction Actions Extracted from Historical ADS-B Data". In: *Proceedings of the 11th OpenSky Symposium*. 2023. DOI: 10.59490/joas.2023.7222.

[11] Kim Gaume, Xavier Olive, David Gianazza, Richard Alligier, and Nicolas Durand. "Extracting aircraft conflict-resolution situations from historical ADS-B data". In: *Transportation Research Interdisciplinary Perspectives* (2025). DOI: 10.1016/j.trip.2025.101669.

[12] Jean-François Lepage. "Safety Components in a Separation Minima". In: *38th Annual Conference of IFATCA, Working Paper No. 83*. Santiago, Chile, 1999. URL: https://ifatca.wiki/kb/working-paper/1998/wp-1999-83/.

[13] Steven D. Thompson. *Terminal Area Separation Standards: Historical Development, Current Standards, and Processes for Change*. ATC-258. Lexington, Massachusetts: MIT Lincoln Laboratory, 1997. URL: https://archive.ll.mit.edu/mission/aviation/publications/publication-files/atc-reports/Thompson_1997_ATC-258_WW-15318.pdf.

[14] Federal Aviation Administration. *Report No. DOT/FAA/NA-92/1 — Controller Response to Conflict Resolution Advisory Prototype*. Technical Report DOT/FAA/NA-92/1. Washington, D.C.: U.S. Department of Transportation / Federal Aviation Administration, 1992. URL: https://rosap.ntl.bts.gov/view/dot/8634/dot_8634_DS1.pdf.

[15] Federal Aviation Administration. *Report No. DOT/FAA/NA-92/2 — Controller Response to Conflict Resolution Advisory Prototype*. Technical Report DOT/FAA/NA-92/2. Washington, D.C.: U.S. Department of Transportation / Federal Aviation Administration, 1992. URL: https://rosap.ntl.bts.gov/view/dot/8635/dot_8635_DS1.pdf.

[16] Kim M Cardosi and Pamela W Boole. *Analysis of pilot response time to time-critical air traffic control calls*. Tech. rep. Federal Aviation Administration, 1991. URL: https://apps.dtic.mil/sti/tr/pdf/ADA242527.pdf.

[17] Jens Nilsson and Jonas Unger. "Swedish civil air traffic control dataset". In: *Data in brief* (2023). DOI: https://doi.org/10.1016/j.dib.2023.109240.

[18] Thomas Pellegrini, Jérôme Farinas, Estelle Delpech, and François Lancelot. "The airbus air traffic control speech recognition 2018 challenge: Towards ATC automatic transcription and call sign detection". In: *arXiv preprint arXiv:1810.12614* (2018).

[19] Jasenka Rakas, Matthew Alvarado, Kezhi He, Drew Kim, and Della Qu. "Analysis of Controller-Pilot Communication Messages with Natural Language Processing". In: *AIAA Aviation 2022 Forum*. 2022. DOI: 10.2514/6.2022-3832.

[20] Amrutha Prasad, Juan Zuluaga-Gomez, Petr Motlicek, Saeed Sarfjoo, Iuliia Nigmatulina, and Karel Vesely. "Speech and natural language processing technologies for pseudo-pilot simulator". In: *12th SESAR Innovation Days* (2022). URL: https://publications.idiap.ch/attachments/papers/2022/Prasad_SID-2_2022.pdf.

[21] Amany M Sarhan, Rawda Fathy, and Hesham A Ali. "Intelligent air traffic control using NLP-enhanced speech recognition and natural language generation". In: *Journal of Electrical Systems and Information Technology* (2025). DOI: 10.1186/s43067-025-00234-9.

[22] Kim Gaume, Richard Alligier, Nicolas Durand, David Gianazza, and Xavier Olive. "Extracting Lateral Deconfliction Actions from Historical ADS-B data with Median Regression". In: *International Conference on Research in Air Transportation*. 2024. URL: https://drive.google.com/file/d/1F6FOorwkBUkjuKdGE4i2uiYnBODL4Y0d/view?usp=sharing.

[23] Luboš Šmídl, Jan Švec, Daniel Tihelka, Jindřich Matoušek, Jan Romportl, and Pavel Ircing. "Air traffic control communication (ATCC) speech corpora and their use for ASR and TTS development". In: *Language Resources and Evaluation* (2019). DOI: 10.1007/s10579-019-09449-5.

[24] Xavier Olive. "traffic, a toolbox for processing and analysing air traffic data". In: *Journal of Open Source Software* 4 (2019). DOI: 10.21105/joss.01518.