Characterization of Response Times based on Voice Communication and Traffic Surveillance Data

Michael Lutz* and Gano Chatterji†
Crown Consulting Inc. at NASA Ames Research Center, Moffett Field, CA, 94035-1000

Husni Idris‡
NASA Ames Research Center, Moffett Field, CA, 94035-1000

A barrier to the integration of remotely piloted aircraft operations in the U.S. National Airspace System is the latency of voice communications between the air traffic controller and the remote pilot, and the latency of communication between the aircraft and the remote pilot. The latency can be substantial especially when satellite-based beyond-radio-line-of-sight communication and relay through the aircraft are employed. This study uses voice recordings of controller-pilot communications and aircraft track data to establish a baseline of pilot readback latencies and maneuver detection delays in the current piloted operations. A machine learning pipeline was developed to parse the contents of the air traffic control clearances including the callsigns using natural language processing. After manually validating the results obtained using the pipeline, the average pilot readback latency was found to be about 0.6 seconds. The average latency between the end of maneuver (inferred from track data), initiated by the pilot in response to the clearance, and the end of clearance was found to be about 176 seconds for altitude change commands, 69 seconds for heading change commands, and 182 seconds for speed change commands. The average latency between the beginning of maneuver and the end of clearance was found to be about 17 seconds for altitude change commands, 17 seconds for heading change commands, and 25 seconds for speed change commands.

I. Introduction

The National Aeronautics and Space Administration (NASA) is studying integration of cargo operations conducted with Remotely Piloted Aircraft (RPA) in the National Airspace System (NAS) as an important use case for investigating scalability of operations with increasingly automated aircraft. A potential barrier to integration of RPA in the NAS is the latency of voice communication between the air traffic controller and the remote pilot when satellite-based Beyond-Radio-Line-of-Sight (BRLOS) communication is used with relay via the aircraft. Furthermore, data downlink from the RPA to Remote Pilot (RP) and uplink of control commands from RP to RPA can suffer latencies because of BRLOS satellite link. These latencies can delay a conflict resolution maneuver and cause other pilots to step-in before the RP gets a chance to readback the controller's clearance. In addition, lack of visual and haptic feedback might also cause delayed RP response. Characterization of voice communication and maneuver time latencies in the current piloted operations is therefore needed for creating a baseline. This study aims to use voice recordings of controller-pilot communications and aircraft track data to characterize these latencies.

There is evidence in the literature that command and control latencies affect pilot acceptability, and voice communication latencies affect air traffic controller acceptability. Reference [1] found control link latency of 100 milliseconds leads to measurable degradation of human performance; latencies of about 250-300 milliseconds leads to unacceptable airplane handling qualities. Reference [2] found it is difficult to control remotely piloted aircraft on approach and landing tasks if the time-delay exceeds 400 milliseconds. The study described in Ref. [3] conducted Human-in-the-Loop (HITL) simulations for determining the effect of communications and command execution delays on air traffic controller interactions with remote pilots and pilots flying conventional aircraft. The delay was varied between 1.5 seconds and 5 seconds, constant or variable within the scenarios. The study found, as expected, smaller latencies for verbal communications with the remote pilots was acceptable by the air traffic controllers compared to longer latencies. The air traffic controllers were unaffected by command execution delays, where the command

* Student Intern, Crown Consulting, Inc., NASA Ames Research Center.
† Senior Scientist and Lead, Crown Consulting, Inc., NASA Ames Research Center, Associate Fellow.
‡ Aerospace Research Engineer, NASA Ames Research Center, Associate Fellow.

execution delay is the difference between the time the pilot begins command execution and the time the air traffic controller completes delivery of the clearance.

The study described in this paper employs a machine learning-based pipeline to measure pilot readback and maneuver detection delays. Maneuver detection delay refers to the amount of time it takes for the aircraft maneuver, initiated by the pilot in response to the commands, to appear in the surveillance data with respect to the time of the end of the clearance issued by the controller. With recent improvements in computational power as well as new neural network architectures such as Long Short-Term Memory (LSTM) networks and transformers, Natural Language Processing (NLP) has become increasingly viable for Air Traffic Control (ATC) applications [3-5]. Thus, many recent papers have explored the integration of machine learning and NLP into air traffic control communications. One of the primary objectives of this field is Automatic Speech Recognition (ASR), which transcribes voice communication to text. A recent Airbus report demonstrated the viability of ASR in the ATC domain, with their best model achieving a 7.62% Word Error Rate (WER) [6]. ASR methods with low WER provide a computer-based process, instead of the tedious manual one, for good quality transcription of large volumes of voice data, especially archived data, for further analysis with computer-based algorithms.

Being able to determine the callsign from the ASR transcribed text is important because the callsign relates the controller clearance to the pilot readback of the clearance. A recent analysis described in Ref. [7] achieved a 95% Callsign Detection Rate (CDR) based on matching callsigns extracted from the transcribed text with the ones in the list of possible callsigns from track data in the region of flight. Furthermore, another paper utilized a named entity recognition transformer, combining track data and command data to achieve a 95.3% F1-score (harmonic mean of precision and recall computed using true positive, true negative, false positive and false negative values) for callsign detection based on the callsign matching approach from the previously described paper [8].

In addition to callsign detection, the ability to recognize the commands within the ATC clearance is important. This task consists of parsing the clearance to identify individual commands (e.g., “turn right heading 030”). The study in Ref. [9] created a pipeline with ASR, Callsign Detection (CD), and Command Recognition (CR) models, demonstrating that it was possible to achieve an 85% Command Recognition Rate (CRR) with respect to truth dataset created by listening to the audio [9]. Moreover, this paper achieved a 96.3% CDR after adding a semi-supervised aspect to their model.

The study described in this paper combines the most efficient and successful aspects of the methods mentioned above: a LSTM-based speech recognition model trained by Appareo Systems, Inc., a rule-based command recognition approach, and a novel callsign extraction and matching approach that utilizes efficiency-optimized Levenshtein distance. The Levenshtein distance metric measures the minimum number of insertions, deletions, and substitutions needed to make the two strings match [10]. To avoid confusion with the terminology, a transmission (clearance or readback) is defined as a complete, uninterrupted sequence of words spoken by an air traffic controller or pilot. For instance, air traffic controller clearance, “American 113 turn right heading 030 and descend maintain 3000” is considered a transmission. A command, however, refers to a specific instruction included within the transmission. For example, “turn right heading 030” and “descend maintain 3000” are referred to as commands in this paper. Readback delay refers to the amount of time it takes the pilot to start readback after the time of end of clearance from the controller.

The rest of the paper is organized as follows. Section II outlines the voice communication processing pipeline enabled by machine learning. Section III discusses the methodology for using the pipeline to compute maneuver detection and pilot readback delays. The results are discussed in Section IV and the paper is concluded in Section V.

II. Voice Communication Processing Pipeline

The machine learning-enabled voice communication processing pipeline developed to automate the semantic understanding of spoken air traffic control commands and for callsign detection is described in this section. This capability is also employed to parse pilot response to ATC transmissions. The inputs to the pipeline are verbal communications audio and track data. Track data are required for CD. The pipeline shown in Fig. 1 performs three tasks: (1) ASR, (2) CR, and (3) CD. ASR is employed in the first step to transcribe the audio transmission from ATC to text. The transcribed text is then processed to obtain the semantic intent for recognizing the commands within the transmission in the second step. Finally, the callsign in the transcribed text is obtained in the third step. If the callsign is shortened or incomplete in the transcribed text, it is inferred from the track data in the third step. Track data are also used for verifying the callsigns inferred from the transcribed text.

A. Automatic Speech Recognition

The function of ASR is transcribing the input audio data into parsed sentences. While automatic speech recognition technology has been available for decades, the viability of ASR systems has only recently been established due to improved computational power and algorithmic developments [11-13]. ASR models typically consist of two sub-models: an acoustic model and a statistical word-stringing model. The acoustic model typically processes small audio segments (e.g., a 25-millisecond segment is often used) and predicts the phoneme being spoken within that given segment. Phonemes represent the phonetic sounds used to build words; the English language has 44 phonemes [14]. For instance, the "a" sound from the word "radio" would be represented by the phoneme "æ." Once the acoustic model identifies the phonemes, a statistical model strings together related phonemes to predict words. The segmentation model assigns the words to sentences, thereby separating the speech of one speaker from that of another speaker.

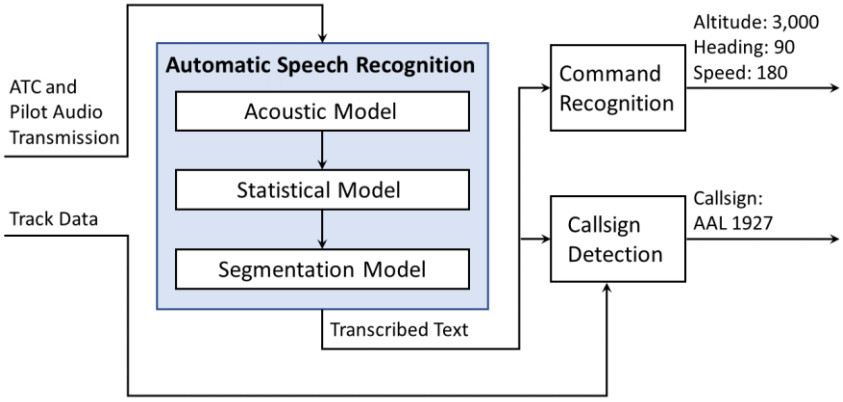

Figure 1. The voice communication processing pipeline.

To measure the success of ASR systems, researchers typically use the WER metric, which is based on the Levenshtein distance. WER compares a transcribed body of text with the truth body of text and quantifies the similarity between the two bodies. A WER of 0.0 is perfect, and a WER near 1.0 is considered severely inaccurate. The WER metric, defined in Eq. (1), factors in the number of word substitutions *S*, insertions *I*, and deletions *D* within a body of text of a given size *n*.

$$WER = \frac{S + I + D}{n} \tag{1}$$

However, there are significant challenges to implementing ASR for air traffic control communication. To begin with, pilots and air traffic controllers use highly specialized vocabulary that is uncommon in speech corpuses commonly used for training ASR systems. Furthermore, significant amount of static noise is often present in ATC voice communication. Both challenges mean that an ASR model must be trained on ATC-specific data for the model to learn air traffic control vocabulary and learn to place less weight on frequencies that typically belong to noise. Signal processing prior to ASR is often employed to reduce or eliminate audio noise.

The pipeline proposed in this paper makes use of an ATC-contextualized ASR model trained by Appareo Systems, which uses bi-directional LSTM layers for phoneme identification. The model was trained on 500 hours of ATC communication and a larger corpus of audio roughly 6,000 hours in length. Because the model was trained on verbal communications between pilots and air traffic controllers, it can identify words and phrases unique to the context of aviation, including the names of waypoints and airports. Furthermore, it provides additional weight to words that are commonly used by pilots and air traffic controllers but are uncommon in everyday English (e.g., "squawk," "foxtrot," and "zulu"). Appareo's ASR system has thus far achieved a 6% WER on live cockpit ATC radio channels. For reference, prior to reaching out to Appareo Systems, attempts with the standard ASR model from Amazon Web Services (AWS) only achieved about 62% WER on our expert-transcribed historical FAA audio data. It might be possible to improve the performance of the AWS ASR by training it on ATC voice communication data.

Yet another challenge faced by ASR systems is understanding when to segment, or separate, utterances into individual sentences. The ASR system must avoid joining two separate transmissions into a single sentence because

of the goals of (1) recognizing commands within the transmissions and (2) separating air traffic controller clearances from pilot readbacks. To this end, the Appareo ASR system segments utterances via a tuned timeout, effectively searching for long pauses in significant audio activity to determine where transmissions begin and end. Furthermore, internal cockpit chatter that is unrelated to ATC communication is intelligently filtered out.

B. Command Recognition

After voice data are transcribed into words, CR is performed by identifying the command category (e.g., "altitude command") via the associated keywords (e.g., "climb") and the attributes (e.g., "3000 feet") of the keywords. Three common command categories are: altitude change, heading change, and speed change. Because ATC communication needs to adhere to the phraseology established by the FAA in the Air Traffic Control Handbook (JO 7110.65), a rule-based command recognition approach can be employed. The command recognition approach used in this paper is designed for multiple sets of commands included within a single clearance. The rules defined for determining the command category and for parsing the content of the speed, heading, and altitude commands are discussed below.

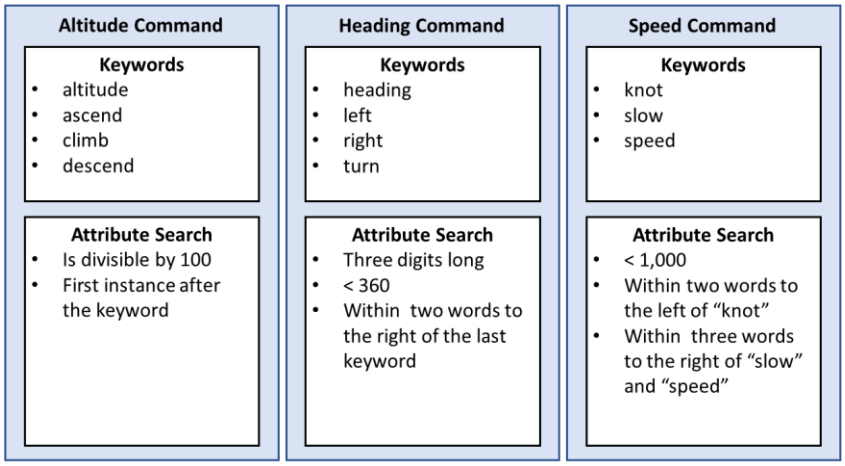

Figure 2. Command recognition keywords and queries.

The parsing of speed change commands follows a two-step rule-based approach shown in Fig. 2. The first step consists of identifying keywords such as "knot," "slow," and "speed" to determine whether the transcribed clearance contains the speed change command. The second step consists of identifying the attributes associated with keywords in the speed change command such as "320 knots." If the keyword "knot" is detected, the text is searched for numbers smaller than 1,000 placed within two words directly to the left of the keyword. Alternatively, if the keywords "speed" or "slow" are detected, the text is searched for numbers smaller than 1,000 placed within three words to the right of the keyword. Thus, a command that includes "speed up to 350" will be processed correctly even if the air traffic controller omits the word "knots." If any of the keywords "knot," "slow," or "speed" are present in a command, but the associated attributes cannot be found, the value "Missing SI" is returned. This serves as a marker for trained humans to listen to the audio of the command to identify the attribute if possible.

For heading change command, the keywords are: "heading," "left," "right," and "turn." The associated attribute is found by searching for a three-digit number with magnitude less than 360 within two words from the last instance of the keyword. Searching two words instead of one helps with the edge cases. For instance, both "…heading um 340" and "…heading 340" will be processed correctly.

Finally, the keywords associated with the altitude command are: "altitude," "ascend," "climb," and "descend." The attribute of these keywords is detected by searching for the first number following the last altitude keyword (e.g., "climb") that is divisible by 100. Results of three real-world examples of ATC clearances parsed by the procedure in Fig. 1, with the keywords and logic summarized in Fig. 2, are shown in Table 1.

Table 1. Results of parsing of ATC clearances.

ATC Clearance	Altitude (feet)	Heading (degrees)	Speed (knots)	Callsign
Alaska 370 turn right heading 150 descend and maintain 4000	4000	150	N/A	ASA370
Delta 779 reduce speed 190	N/A	N/A	190	DAL779
Skywest 5383 descend and maintain 2000	2000	N/A	N/A	SKW5383

C. Callsign Detection

Another principal task for the pipeline is identifying the callsign of the aircraft specified in an ATC clearance such as the ones in Table 1. To correlate the commands with the resulting maneuvers observed in the track data, one must be able to link these two data sources by callsign. This is not a simple task because air traffic controllers often abbreviate callsigns; thus, the spoken callsign often does not match the formal callsign. This challenge will be addressed later in this section.

The first step of the callsign detection process consists of identifying the spoken callsign in the transcribed text. This is a named entity recognition task, which can be accomplished via a rule-based approach. Specifically, the parser searches for relevant keywords, usually the names of airlines or the keyword "November." The keyword "November" is used for the N-Numbers (tail numbers) of general aviation aircraft. It is important to note that "AAL2066" found in track data would be transcribed as "American Airlines 2066" by Appareo's ASR system after data cleaning from voice data. The International Civil Aviation Organization (ICAO) code associated with the airline is assigned as the initial part of the callsign. For American Airlines this part is AAL. Next, the parser searches for alphanumeric characters that immediately follow the airline name or the keyword "November." The alphanumeric characters found are added to the initial part of the callsign to construct the complete callsign. The final step consists of matching this callsign with one of the ones seen in the track data within the line-of-sight range of the radio frequency being used for voice communication (for example, 150 nautical miles around an airport).

The extent of match between the callsign derived from the transcribed text and the callsigns in the track data can be established using the Levenshtein distance metric. Examples of the Levenshtein distance are presented in Table 2.

Table 2. Example Callsign Matches

Detected Callsign	Actual Callsign	Levenshtein Distance	Explanation
AAL779	AAL1779	1	one addition
SKW16802	SKW1680	1	one deletion
JAL1780	JBU1780	2	two substitutions
UAL047	UAL1472	2	one addition, one substitution

It should be noted that if one applies the Levenshtein distance to a large track dataset, there is increased likelihood of multiple matches. For example, "AAL2766" and "AAL2866" are equidistant to "AAL266" because they can be obtained by inserting one number (7 or 8) to "AAL266" after the number 2. To prevent multiple matches, callsign search is restricted to track data within a 150 nautical mile radius from a suitable reference location such as the airport of arrival—Los Angeles International Airport (KLAX) in this instance—for arrival traffic and within 20 minutes of the voice clearance in this study.

D. Measuring Success

The subsections above have explained the three primary tasks: ASR, CR and CD of the language processing pipeline. To test the viability of the pipeline, however, a metric of its success must also be established. The Command Recognition Rate (CRR) is the metric that has been used in this paper. It is defined as the ratio of the number of correctly identified commands—keywords and their associated attributes—to the total number of commands. CRR is computed using Eq. (2).

$$CRR = \frac{1}{n}\sum_{i}^{n} T(c_{p_i}, c_{t_i}) \tag{2}$$

$$T(c_p, c_t) = \begin{cases} 1 & if\ c_p = c_t \\ 0 & if\ c_p \neq c_t \end{cases}$$

where n represents the total number of commands, c_p is the parsed command identified by processing transcribed text data and c_t is the same command verified by the pilot based on listening to the audio. CRR is 1 if all the commands identified by processing are the same as that transcribed by the pilot based on listening to the audio. It is 0 if they are different.

The truth data corresponding to the transcribed to text data were created with the assistance of an Instrument Flight Rules (IFR)-rated pilot with 1,500+ in-flight hours, who manually processed the command data and the callsign data. A data table with 600 rows, with each row containing a transmission's audio file and ASR transcription, was prepared for the pilot. The pilot listened to the audio clips to identify the altitude, heading, and speed commands and noted the associated values. Next, the callsigns were identified by the pilot. Afterwards, these two steps—identifying the commands and identifying the callsigns—were done on the transcribed text using the procedure illustrated in Fig. 1 and compared with truth data for computing the CRR. Note that comparison with the truth data is important because it is possible for the computational procedure to achieve perfect CRR with respect to incorrect transcription despite achieving low accuracy with respect to the actual spoken command. Both CRR with respect to transcribed text and truth data are important to consider, as they isolate specific insight into the quality of ASR, CR and CD.

Later in Section IV, CRR and CDR results are presented. CDR is also defined by Eq. (3) with n as the total number of callsigns, c_p as the callsign identified by processing transcribed text data and track data and c_t as the same callsign verified by the pilot based on listening to the audio.

III. Response Time Characterization Methodology

In this section, procedures for maneuver detection delay and pilot readback delay are discussed. Figure 3 shows the end-to-end response times. The Maneuver Completion Detection Delay (MCDD) is the difference between the maneuver completion seen on ATC Display (inferred from track data) and the start time (end of ATC clearance delivery) shown in Fig. 3. The Maneuver Initiation Detection Delay (MIDD) is the temporal difference between when the aircraft maneuver is first recognized in the track data and the end of clearance delivery. Note end of clearance heard on frequency and beginning of pilot readback heard on frequency means that a radio listening to ATC communication on the said frequency would hear them after broadcasting equipment, wireless transmission, and reception equipment delays.

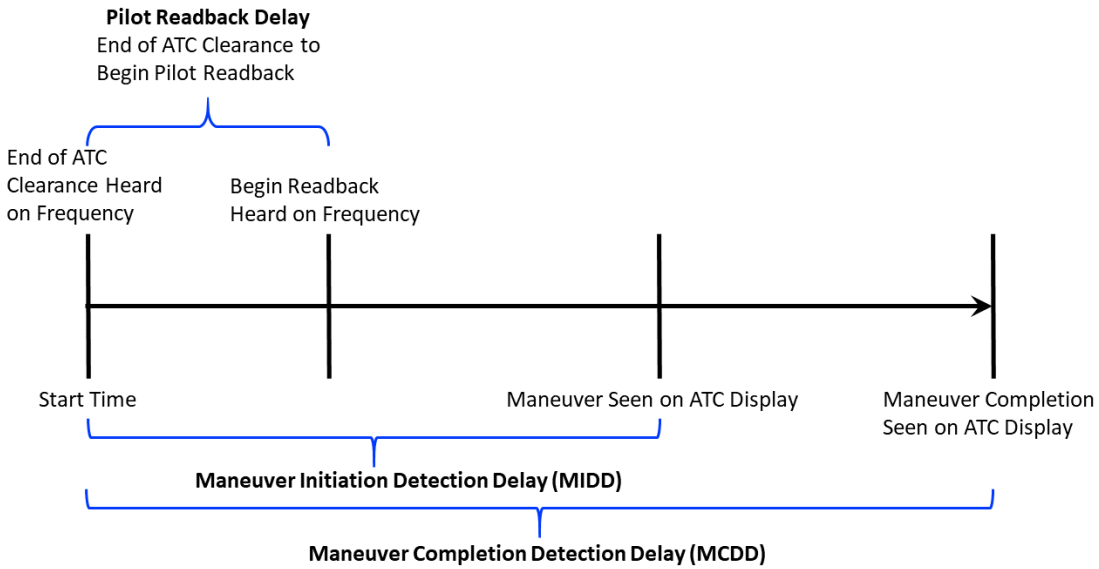

Figure 3. End-to-end response times.

A. Data Sources: ADS-B Exchange and LiveATC.net

The flight track data were obtained from ADS-B Exchange, a worldwide network of ADS-B, Mode S and Multilateration (MLAT) feeders. ADS-B Exchange can provide flight position updates faster than one update per

second. In this study, the ADS-B Exchange's *readsb-hist* dataset, which stores the position and callsign information of all airborne aircraft as a function of time, has been used for CD. The relevant values in this dataset include the following: altitude, groundspeed, track-heading, and timestamp. The ATC voice data were obtained from LiveATC.net, which provides live and historical radio data collected via receivers positioned across the globe.

B. Measuring Maneuver Completion Detection Delay

To determine MCDD and MIDD, 24 hours of spoken ATC data from the KLAX Approach Zuma/NW Arrival (frequency: 124.500 Hz), and ADS-B track data of aircraft within the 150-nautical-miles range from the KLAX beginning on March 1, 2022 (00:00:00) GMT and ending on March 1, 2022 (23:59:59) GMT were collected. KLAX was chosen because of the large volume of daily traffic.

Following data collection, the process outlined in Fig. 1 was applied to the recorded speech corpus to output the command dataset with the following columns: transcription, altitude, heading, speed, callsign, and timestamp. However, before inputting the transcribed commands into this process, the following data cleaning techniques were applied leveraging the Pandas Python library and regular expression. First, radio noise was removed by filtering out utterances less than 15 characters in length. Next, numbers that were spelled out such as One-Zero-Zero were transformed into alphanumeric characters such as 100. This proved to be especially necessary for the callsign detection task discussed in Section II. Finally, the parsed results obtained from the process in Fig. 1 were manually verified against the pilot verified truth dataset to assess the performance of the procedure in terms of CRR. Finally, the maneuver completion detection delays were determined using the logic illustrated in Fig. 4.

The process in Fig. 4 is initialized for each command detected in the transmission by setting the time, t, to the timestamp, t_0, in the transcribed text of the transmission containing the command. The next step consists of checking the values of the attributes of the keywords in the detected commands against the values in the track data. For example, if the value of the attribute of the keyword "descend" is 4,000 in the altitude command "descend and maintain 4000," the maneuver would be deemed completed when the altitude in the track data is found to be close to 4,000 feet. Time t is incremented by five seconds iteratively to keep searching the track data for a possible match within the specified tolerance. The tolerance values are set to ± 200 feet for altitude change commands, $\pm$ 5 degrees for heading change commands, and $\pm$ 10 knots for speed change commands. MCDD is given as the difference between the timestamp in the matched track data and the timestamp of the command in the transcribed text, t_0.

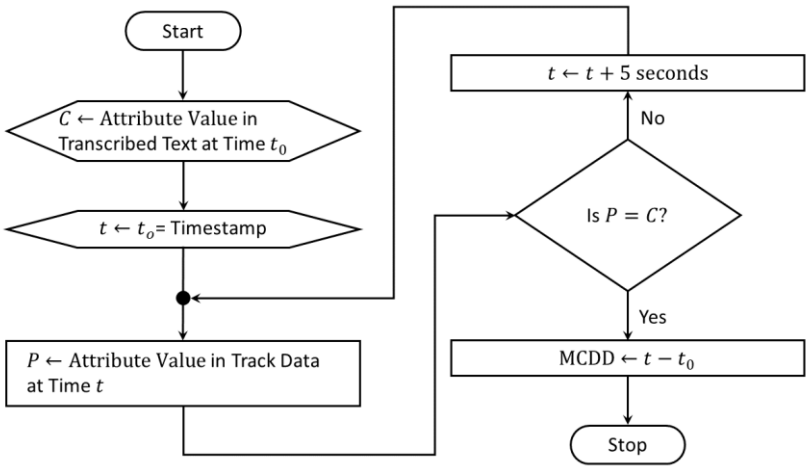

Figure 4. Flowchart describing MCDD computation.

Finally, if a pilot has not finished executing a command within 500 seconds, the execution is considered to have not been completed in a timely fashion. Figure 5 shows the visualization of the trajectory of Air Canada 558 (ACA558) destined to KLAX on March 1, 2022 resulting from maneuvers in response to the ATC clearances listed in Table 3. This table also lists the MCDD computed following the procedure outlined in Fig. 4. Solid lines in Fig. 5 begin when a command is issued and end when a command is completed. Position from track data is displayed at the endpoints. Each circle represents the aircraft position at a given time, drawn at five-second intervals.

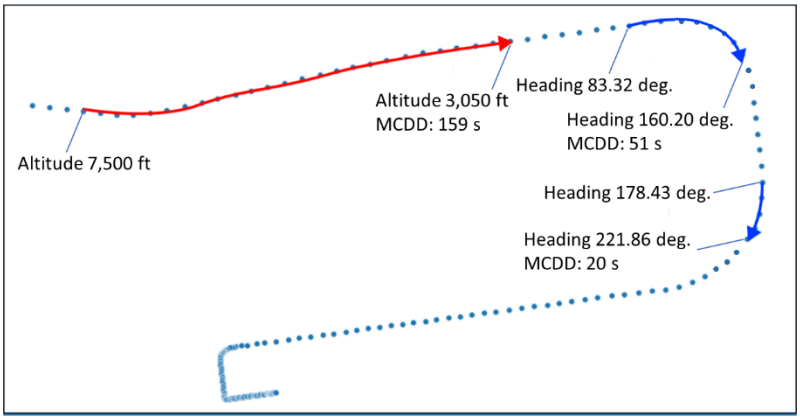

Figure 5. Visualization of ACA558 trajectory resulting from pilot actions in response to ATC clearances.

Table 3. Results of parsing of ATC clearances for ACA558

ATC Clearance	t_0	MCDD
Air Canada 558 descend and maintain 3000	22:19:10 (PST)	159 s
Air Canada 558 turn right heading 160	22:22:18 (PST)	51 s
Air Canada 558 fly heading 220 to join final	22:23:44 (PST)	20 s

C. Measuring Maneuver Initiation Detection Delay

The maneuver is deemed to be initiated when the change expected in the aircraft state due to pilot response to the command in the clearance is first seen in the track data. This expected change is based on specified thresholds, which are ± 100 feet for altitude change, ± 2.5 degrees for heading change, and ± 2.5 knots for speed change. For instance, an aircraft commanded to descend to 3,000 feet from 7,000 feet is considered to have initiated the maneuver when the altitude is seen to be 6,900 feet or less in the surveillance data.

The following logic was followed for determining the MIDD: (1) note the timestamp indicating the end of ATC clearance containing the command from the transcribed text, (2) loop through track data in time until maneuver initiation is detected, (3) subtract the noted timestamp from the time of maneuver initiation found in the track data. This process is the same as the process for determining MCDD shown in Fig. 4 except for the decision block checking for the beginning of the maneuver instead of the end of the maneuver, and the time difference, $t - t_0$, being MIDD instead of MCDD. Based on the MCDD results, MIDD is considered incorrect if greater than 100 seconds.

D. Measuring Pilot Readback Delay

Compared to maneuver detection delay, pilot readback delay is relatively simple to measure; it only requires ASR transcribed text. To analyze pilot readback in different ATC domains, voice communication on KLAX Tower South (frequency: 120.950 Hz), ZLA Center Sector 25 (frequency: 126.525 Hz), and KLAX Approach Final North (frequency: 133.375 Hz) were recorded. These data from 6:00 AM to 7:30 AM PST (1400Z—1530Z) are representative of the busy period of KLAX traffic.

After collecting data, the audio data were processed using Appareo's ASR system, which output the transcribed text containing the list of separated ATC and pilot transmissions. The pilot readback delay is calculated by subtracting the end timestamp of the ATC transmission heard on frequency from the start timestamp of the pilot's readback heard on frequency. The key to this capability is successful CD, because it is the callsign that relates the ATC clearance to the pilot readback. The pilot readback accuracy was determined by comparing the transcribed readback responses to the truth dataset provided by the pilot.

IV. Results and Discussion

A. Assessing Machine Learning Pipeline Quality

Following the methodology outlined in Section II, the CRR is computed with respect to both the truth dataset as well as the transcriptions. The average CRR with respect to the transcriptions of 600 commands recorded from voice communication on KLAX Zulu/NW Approach frequency was found to be 90.0%, where this average is based on the altitude, heading and speed command categories taken together. The average CRR dropped to 85.3% with respect to the truth dataset. This means that command parsing is more error prone compared to transcription. Given that Appareo System's ASR has a 6% WER (although benchmarked on live cockpit ATC radio channels), it performs reasonably well to make command recognition viable.

For a more complete characterization of the viability of the CR procedure, the CRR values for altitude, heading, and speed change commands were examined individually; the results are summarized in Table 4. Altitude change commands were correctly recognized with a CRR of 94.2% with respect to the truth dataset. Heading change commands were recognized with a CRR of 91.4%. The performance on the speed commands was not as good; the CRR for speed commands was found to be 70.2%. Speed change commands are difficult to detect because the current rule-based procedure misses certain combinations of words in the speed commands such as "return to normal speed" or "maintain present speed." The performance could probably be improved by examining larger datasets for finding additional keywords and the associated attributes for addition to the rule-based procedure for improving the recognition of speed commands.

Table 4. Command and callsign recognition rate results

	Altitude	Heading	Speed	All Commands	Callsign
CRR/CDR-Transcriptions	95.1%	92.6%	82.3%	90.0%	99.8%
CRR/CDR-Truth	94.2%	91.4%	70.2%	85.3%	83.4%

The CDR with respect to the transcriptions was found to be nearly perfect with a value of 99.8%. The CDR was 83.4% with respect to the truth dataset. CDR could be improved by additional training of the ASR to recognize more airline and aircraft operator names. However, recognizing airline names is a bit challenging because of mergers and acquisitions, and airlines going out of business. CDR is also lower because air traffic controllers often speak the alphanumeric characters of a callsign relatively quickly, making it inherently difficult to transcribe a callsign perfectly.

B. Maneuver Detection Delay

The procedures described for computing the MCDD and MIDD were used on the 600 manually verified transmissions from the KLAX Zulu/NW Approach frequency on March 1, 2022 to generate the MCDD and MIDD statistics summarized in Table 5. The average MCDD, considering the altitude, speed and heading command categories together, was 151.3 seconds for this dataset. On average it took 176.1 seconds to complete an altitude change command, 69.3 seconds to complete a heading change command, and 182.3 seconds to complete a speed change command.

The average MIDD was 20.70 seconds considering altitude, heading, and speed commands together. The average MIDD values for altitude, heading, and speed commands in isolation were 16.69 seconds, 16.70 seconds, and 25.47 seconds, respectively. These results show that within the LAX approach, speed change MIDD is larger compared to altitude and heading change MIDDs. This is probably due to faster pitch and roll dynamics compared to acceleration/deceleration dynamics. A smaller threshold value for detecting speed change might help in detecting speed change sooner, but it might be susceptible to falsely indicating a maneuver prior to the actual maneuver.

To get some insight into MCDD results, the results were examined for common values of attributes associated with the keywords in the altitude, heading, and speed change commands. The larger MCDD for altitude commands is consistent with the fact that most altitude commands during approach in this dataset ask aircraft to reduce altitude from 7,500 feet to 3,000 feet. MCDD value of 176.1 seconds for descent of 4,500 feet translates to a descent rate of about 1,533 feet/minute. For reference, the descent rate of Boeing 737-900 is 1,289 feet/minute and that of Airbus A320-100 is 1,527 feet/minute at 8,000 feet altitude according to BADA version 3.9. Aircraft are required to fly at a Calibrated Airspeed (CAS) of 250 knots or less below 10,000 feet. This translates to True Airspeed (TAS) of about 280 knots at 8,000 feet altitude and 229 knots at 3,000 feet altitude assuming standard atmosphere. This is a speed

reduction of 51 knots. Considering average descent rate over 5,000 feet altitude (8,000 feet to 3,000 feet descent), the time for speed reduction of 51 knots using the BADA Boeing 737-900 model is 4.48 minutes (about 269 seconds) and with the Airbus A320-100 model is 3.73 minutes (about 224 seconds). These two values are within one standard deviation bounds listed in Table 5. Unlike altitude and speed change commands, heading change commands were completed in less than half the time. To maintain the standard rate of turn of 3 degrees/second at the average TAS of about 254 knots during descent from 8,000 feet to 3,000 feet altitude, a bank angle of 35 degrees is needed. The aircraft can complete a 360 degree turn in 120 seconds at the standard rate of turn. In 69.3 seconds, the aircraft can change heading by about 208 degrees at the standard rate of turn.

Table 5. Command maneuver delay summary statistics

	Altitude	Heading	Speed	All Commands
Maneuver Completion Detection Delay (seconds)				
Average MCDD (seconds)	176.1	69.3	182.3	151.3
MCDD Standard Deviation (seconds)	62.3	43.5	107.8	98.3
MCDD 25th Percentile (seconds)	143.0	45.0	101.5	69.0
MCDD 75th Percentile (seconds)	205.5	81.0	244.5	211.0
Maneuver Initiation Detection Delay (seconds)				
Average MIDD (seconds)	16.69	16.70	25.47	20.70
MIDD Standard Deviation (seconds)	11.23	6.47	16.35	13.60
MIDD 25th Percentile (seconds)	11.00	10.00	17.00	12.00
MIDD 75th Percentile (seconds)	15.00	21.00	30.00	23.00

C. Pilot Readback Delay

Following the procedure described in Section II and the computation of pilot readback delay described in Section III Subsection D, pilot readback delays were computed using ATC voice data from the approach, center, and tower domains. The dataset consisted of 257 commands issued between 6:00 AM and 7:30 AM. Specifically, there were 101 clearances on KLAX Tower South (frequency: 120.950 Hz), 100 clearances on ZLA Center Sector 25 (frequency: 126.525 Hz), and 56 clearances on KLAX Approach Final North (frequency: 133.375 Hz). The average readback delay was found to be 0.639 seconds considering the three domains. Figure 6 shows the average readback delay of 0.443 seconds for tower communications, 0.935 seconds for center communications, and 0.669 seconds for approach communications. The standard deviation values were determined to be 0.326 seconds, 0.949 seconds, and 0.716 seconds, respectively. While these differences were seen in the different phases of flight, there isn't a definite reason to expect them, especially with two pilots onboard the aircraft. The figure also shows the 95% confidence interval bars drawn over each bar graph, where the confidence level is the probability of the mean value being within the indicated bounds.

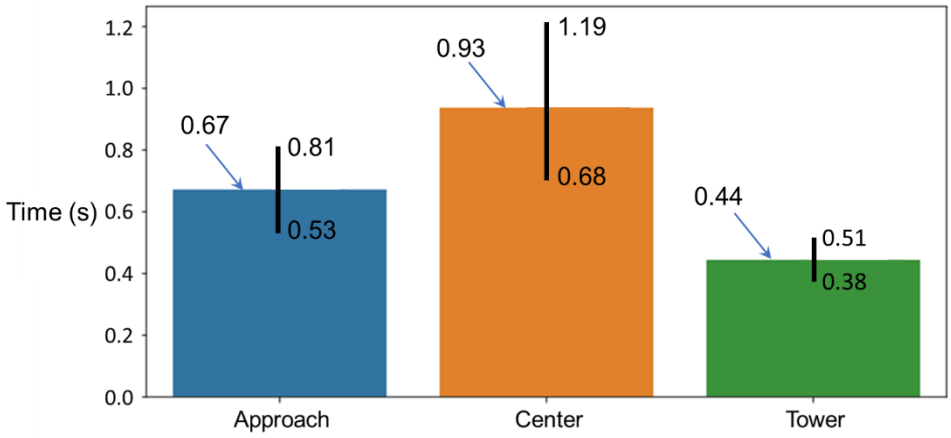

Figure 6. Pilot readback delays on approach, center, and tower frequencies.

V. Conclusions

In this paper, a natural language processing pipeline was developed that receives audio data and flight track data as inputs and returns transcribed text for the commands, detected command category based on keywords, values associated with keywords, and the callsigns. Results obtained by this procedure compared to the truth dataset, created with input from an instrument-rated pilot who listened to the air traffic control audio data, show that about 85% of the commands in the air traffic control clearances and 83% of the callsigns in the dataset were correctly recognized. With respect to transcriptions, 90% of the commands and 99.8% of the callsigns were correctly recognized. The maneuver completion detection delay—the difference between the time when maneuver completion is recognized in the track data and the time of end of clearance delivery heard on frequency—for heading change command was less than half compared to the altitude and speed change commands during approach to the Los Angeles airport. The maneuver initiation detection delay—the temporal difference between the initiation of maneuver seen in the track data and the time of the end of clearance—was found to be the largest for speed change commands. The pilot readback delay—the difference between the time pilot starts readback heard on frequency and the end of the time of clearance heard on frequency—on tower, approach and center frequencies was found to be less than one second. The pilot readback delay, maneuver initiation detection delay and maneuver completion detection delay results presented for the manned aircraft establish a baseline for comparison with those obtained for remotely piloted aircraft. Larger pilot readback delays might require alternative means of communication for remotely piloted vehicles to be operationally acceptable. Large maneuver initiation delays could impact separation assurance.

Acknowledgments

The authors thank Brenda Wyland, Josh Gelinske, Jesse Trana, and Jaden Young from Appareo Systems for providing their speech-to-text transcription system, James Stanford and Dan Streufert from ADSB Exchange for providing aircraft track data, and Andy Lutz, an IFR-rated pilot, for listening to the audio recordings and manually transcribing the air traffic control clearances and pilot readbacks to text for validating the results in this paper.

References

[1] De Vries, S. "UAVs and Control Delays," *Defense Technical Information Center*, 2005.

[2] Wang, F., Qia, S., and Jing, L., "An Analysis of Time-delay for Remote Piloted Vehicle," *MATEC Web of Conferences 114*, 04012, 2017.

[3] Vu, K.-P. L., Chiappe, D., Morales, G., Strybel, T. Z., Battiste, V., Shively, J., and Buker, T. J., "Impact of UAS Pilot Communication and Execution Latencies on Air Traffic Controllers' Acceptance of UAS Operations," A*ir Traffic Control Quarterly*, vol. 22, 2014, pp. 49–80.

[4] Hochreiter, S., and Schmidhuber, J., "Long Short-Term Memory," *Neural Computation*, vol. 9, 1997, pp 1735-1780.

[5] Sak, H., Senior, A., Beaufays, F., "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," *arXiv preprint arXiv:1402.1128*, 2014.

[6] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I., "Attention is All You Need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
[7] Pellegrini, T., Farinas, J., Delpech, E., and Lancelot, F., "The Airbus Air Traffic Control Speech Recognition 2018 Challenge: Towards ATC Automatic Transcription and Call Sign Detection," *arXiv preprint arXiv:1810.12614*, 2018.
[8] Shore, T. "Knowledge-Based Word Lattice Re-Scoring in a Dynamic Context," Master's Thesis, Saarland University, Saarbruken, Germany, 2011.
[9] Zuluaga-Gomez, J., Veselý, K., Blatt, A., Motlicek, P., Klakow, D., Tart, A., Szöke, I., Prasad, A., Sarfjoo, S., Kolčárek, P., Kocour, M., Černocký, H., Cevenini, C., Choukri, K., Rigault, M., and Landis, F., "Automatic Call Sign Detection: Matching Air Surveillance Data with Air Traffic Spoken Communications," *Proceedings of 8th OpenSky Symposium*, vol. 59, 2020, p. 14.
[10] Konstantinidis, S., "Computing the Levenshtein Distance of a Regular Language," *IEEE Information Theory Workshop on Coding and Complexity (ed. Dinneen, M. J.)*, 2005.
[11] Ohneiser, O., Sarfjoo, S., Helmke, H., Shetty, S., Motlicek, P., Kleinert, M., Ehr, H. and Murauskas, Š., "Robust Command Recognition for Lithuanian Air Traffic Control Tower Utterances," *Proceedings of the InterSpeech*, 2021.
[12] Connolly, D. W., "Voice Data Entry in Air Traffic Control," *Proceedings of Voice Technology for Interactive Real-Time Command/Control Systems Application*, NASA Ames Research Center, Moffett Field, CA, December 6-8, 1977.
[13] Hamel, C. J., Kotick, D., and Layton, M., "Microcomputer System Integration for Air Control Training," Special Report SR89-01, Naval Training Systems Center, Orlando, FL, 1989.
[14] Kessler, B., and Treiman, R., "Syllable Structure and the Distribution of Phonemes in English Syllables." *Journal of Memory and Language*, vol. 37, No. 3, 1997, pp.295-311.