Int J Speech Technol (2017) 20:1049–1061
DOI 10.1007/s10772-017-9467-4

A voice command detection system for aerospace applications

Shima Tabibian[1]

Received: 27 May 2017 / Accepted: 3 October 2017 / Published online: 26 October 2017

Abstract Nowadays, according to ever-increasing volumes of audio content, audio processing is a vital need. In the aerospace field, voice commands could be used instead of data commands in order to speed up the command transmission, help crewmembers to complete their tasks by allowing hands-free control of supplemental equipment and as a redundant system for increasing the reliability of command transmission. In this paper, a voice command detection (VCD) framework is proposed for aerospace applications, which decodes the voice commands to comprehensible and executable commands, in an acceptable speed with a low false alarm rate. The framework is mainly based on a keyword spotting method, which extracts some pre-defined target keywords from the input voice commands. The mentioned keywords are input arguments to the proposed rule-based language model (LM). The rule-based LM decodes the voice commands based on the input keywords and their locations. Two keyword spotters are trained and used in the VCD system. The phone-based keyword spotter is trained on TIMIT database. Then, speaker adaptation methods are exploited to modify the parameters of the trained models using non-native speaker utterances. The word-based keyword spotter is trained on a database prepared and specialized for aerospace applications. The experimental results show that the word-based VCD system decodes the voice commands with true detection rate equal to 88% and false alarm rate equal to 12%, in average. Additionally, using speaker adaptation methods in the phone-based VCD system improves the true detection and false alarm rates about 21% and 21%, respectively.

Keywords Keyword spotting · Hidden Markov model · Voice command · Adaptation · Language model · Rule-based

1 Introduction

1.1 Definition

Speech recognition is used in a wide range of applications such as receiving and understanding a set of simple commands and even extracting all information from speech signal. In some applications, the goal is to detect only specific keywords or phrases uttered by a speaker. In such cases, if the speaker utters other words or phrases rather than the special keywords, speech recognition problem is converted to keyword spotting problem.

Four major applications of keyword spotting are keyword monitoring, audio document indexing, command-controlled devices and dialogue systems. The focus of this paper is mainly on the command control devices, which monitor the input spoken utterances and reacts after detecting specific voice command.

1.2 Applications

Voice command Detection (VCD) has many social and industrial applications. Different applications of VCD can be divided into four groups. The first group relates to the autonomous robots specialized for helping elder people or people with disabilities (Fezari and Bousbia-Salah 2006; Wang et al. 2015). The second group includes the smart

✉ Shima Tabibian
tabibian@ari.ac.ir

[1] Aerospace Research Institute, Ministry of Science, Research and Technology, Aerospace Research Center Lane, Mahestan Street, Iran Zamin Street, Tehran 14657-74111, Iran

homes and controlling the home appliances via spoken commands (Ahmed et al. 2012; Butt et al. 2011; Cornu et al. 2002; Manikandan et al. 2015; Principi et al. 2015). The third group relates to the vehicle controlling via voice (Firdaus et al. 2015; Özkartal 2015) and the last group contains software controlling via spoken commands (Watile et al. 2015). In the aerospace field, voice commands could be used instead of data commands in order to speed up the command transmission (command agility) compared to the usual data command transmission methods. In addition, voice commands can help crew members in completing their tasks by allowing hands-free control of supplemental equipment. As the third point, voice commands could be used besides (not instead of) the data commands as a redundant system for increasing the reliability of command transmission.

1.3 Literature review

Regardless of the reason for using voice commands in the aerospace applications, it is necessary to have an approach or a system to recognize them, accurately. One example of VCD systems is a voice guiding system for a robot arm (Fezari et al. 2012). The reported VCD methods differ in various speech features and classification methods used for decoding the voice commands. All the experimental results are obtained using a corpus prepared for the project. There are five voice commands and each of them contains just one word. The performance of the system is evaluated for each command, separately. The best accuracy is obtained for the fifth command and is about 92%. The average accuracy is about 88.2 in the best case. Another example is a method that enables a computer system to perform tasks via voice commands (Gupta et al. 2014). Accuracy of the proposed method is 90%, which is applicable for the female voices as well as male. Again, in this work the voice commands contain just one word such as “mute” and “open”. The third example is a voice command based ground truth collection system which its dictionary is consisted of 5 digit word (“one”, “two”, “three”, “four”, “five”) and 5 command word (“up”, “down”, “start”, “stop”, “back”) (Hoque et al. 2014). The original dataset is a recording of 30 native speakers. The system produces an average accuracy rate of 93.89% in the environment without noise and 58.1% in environments with noise. Voice commands are also used in the field of aerospace applications (Morris et al. 1993; Weinstein 1995) and result in acceptable performance of the whole system.

1.4 The main idea

In almost all different mentioned works, the voice commands are limited to 5–10 single-word commands. In such condition, the VCD converts to a simple word recognizer. It is expected that for such limit systems the detection rate is very high (about 90% or even more). In this paper, a VCD system is proposed to decode about 63 different voice commands, which contains more than two words, in an online manner with a low false alarm rate. Thus, the proposed system has two main advantages compared to the prevailing systems. First, the number of voice commands are not limited to just 5 or 10. Second, the voice commands contain more than one word (seven words in some commands).

1.5 Paper contributions

The system is mainly based on a keyword spotting method (in contrast to the existent VCD systems which are based on speech recognition methods), which detects only target keywords predetermined in the dictionary. The detected keywords are input to a rule-based language model that decodes the voice commands based on them and their locations. In this paper, both phone-based and word-based keyword spotting methods are mentioned. The phone-based keyword spotter is trained on TIMIT database. In order to compensate the differences between the test and train platforms, speaker adaptation methods are used to estimate new model parameters for new speakers. For training word-based keyword spotter, a complete database (about 4 h) has been prepared and labeled based on word unit. In this case, the train and test platforms are the same and it is expected that the performance of the word-based system would be higher than that of the phone-based one (even with adaptation phase). The main contributions of this paper are:

1. A framework for VCD based on *keyword spotting*.
2. Complete *VCD software* for aerospace application with a guide for users.
3. A rule-based *language model* for decoding the voice commands based on discriminative keywords (since in the existent VCD system, the voice commands are usually composed of just one word, these is no need to use a language model).
4. A 4-h non-native (the mother language of speakers are Persian but they utter voice commands in English) *database* for aerospace applications.

1.6 Paper structure

This paper is organized as follows. The proposed framework for VCD is proposed in Sect. 2. Section 3 introduces the non-native database, which is prepared for aerospace applications. The experimental conditions and results evaluations are presented in Sect. 4. Finally, the paper is concluded in Sect. 5.

2 The proposed framework for voice command detection

Figure 1 shows the proposed framework for keyword spotting-based VCD.

As Fig. 1 shows, the whole VCD system is composed of two main parts: keyword spotting and voice command decoding parts. The output of the first part is some discriminative keywords, which are injected to the second part as input arguments. In the second part, a rule-based language model is proposed as a voice command decoder, which decodes input voice commands based on the mentioned discriminative keywords and several rules. The output of the second part is the decoded voice commands.

As another viewpoint, the VCD system includes two main phases: Train and test phases. The train phase contains three sub-sections; Feature extraction, post processing and model training. The test phase includes pre-processing, feature extraction, post-processing, keyword spotting and rule-based language model. The adaptation phase is optional and is applied to the trained models in order to compensate the differences between test and train platforms. These subsections have been discussed completely, in the following.

2.1 Feature extraction

Feature extraction from speech signals converts the speech waveform into some useful parametric representation. It plays an important role in separating speech patterns from one another. However, extracted features should meet some criteria such as easy to measure extracted speech features, discriminating different classes accurately, perfect in showing environment variation and stability over time.

There are different methods for feature extraction from speech signals. The more important features are Linear Predictive Coding (LPC), Linear Predictive Cepstral Coefficients (LPCC), Perceptual Linear Predictive (PLP)

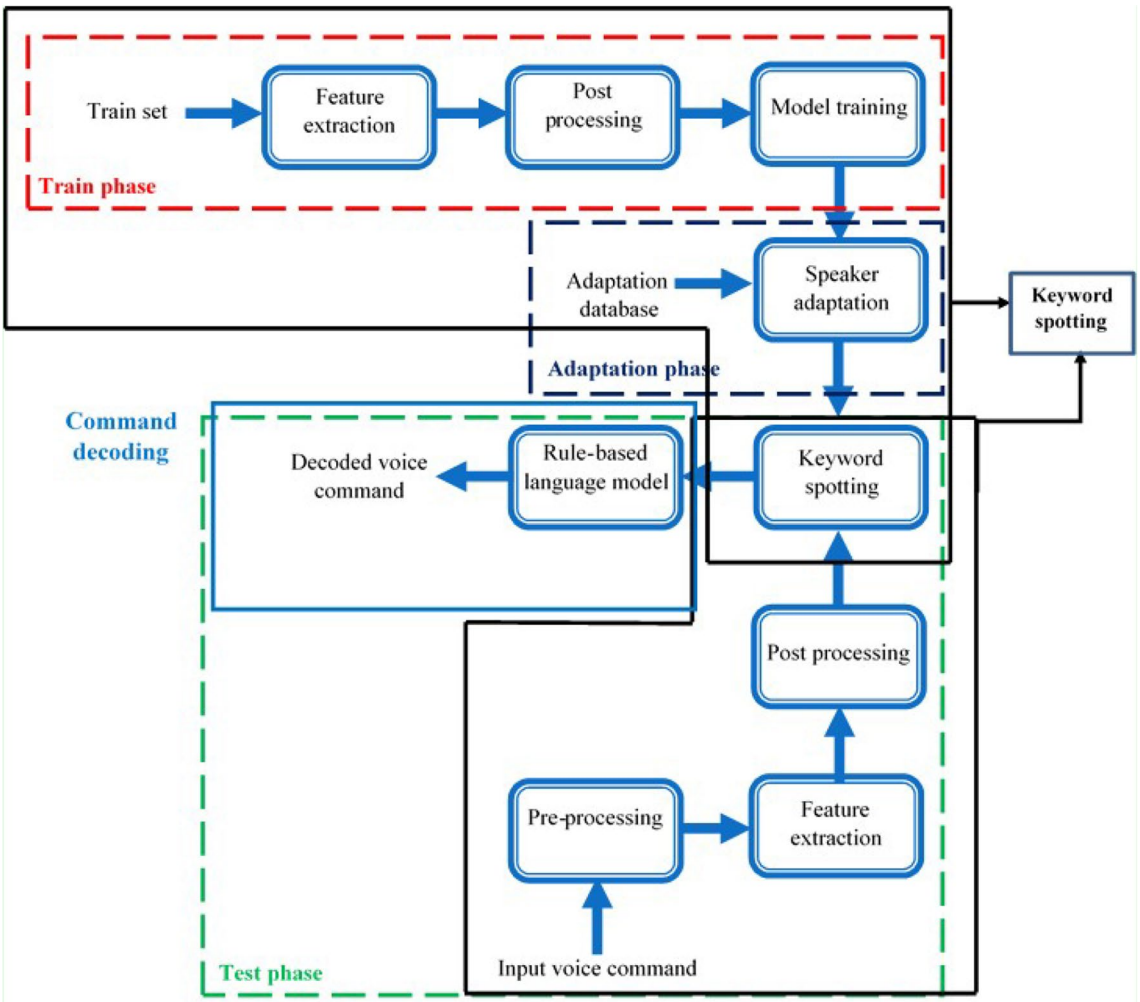

Fig. 1 The proposed framework for VCD system

Coefficients, Mel-Frequency Cepstral Coefficients (MFCC) and Wavelet features (Mporas et al. 2007).

According to acceptable performance of MFCC features, they are extracted from speech signals in both train and test phases of the baseline framework. Thus, 39 MFCC features (12 MFCC features plus energy plus their first and second derivatives [Delta and Delta Delta coefficients)] are extracted from speech signals.

2.2 Pre-processing

The pre-processing of speech utterances in this paper is applied in order to cancel the noise effects and enhance the input speech. Several methods can be used for improving the input voice commands, which are grouped to speech enhancement methods, feature compensation methods, model adaptation methods and methods based on the ear properties (Li et al. 2014). In this paper, the speech enhancement methods have been considered for noise cancelation. There are various speech enhancement methods (Ngo et al. 2012; Tabibian et al. 2015; Vaseghi 2008). In this paper, the input command is segmented into speech and silence parts using a voice activity detection (VAD) method (Tranter et al. 2004). Then, the noise signal is estimated in the silence parts. After that, the noisy speech is enhanced using the estimation of noise. This method is perfect when the signal to noise ratio (SNR) is greater than 10 db. In the lower SNRs, the method leads to perceptual quality degradation and signal distortion. Unidirectional microphones can be used in this method to improve the quality of the input command as more as possible.

2.3 Post processing

The performance of speech recognition systems degrades dramatically when speech is corrupted by background noise and channel distortion. To overcome this problem, several normalization techniques have been proposed such as cepstral mean normalization (CMN), cepstral variance normalization (CVN) (Viikki et al. 1998), cepstral mean and variance normalization (CMVN) and cepstral gain normalization (CGN) (Yoshizawa et al. 2004) techniques. In the baseline framework, the post-processing is applied to normalize MFCC features using CMVN method. In CMVN method, cepstral coefficients are normalized using Eq. (1) (Chen et al. 2002):

$$\bar{C}_{td} = (C_{td} - \mu_d)/\sigma_d \tag{1}$$

where μ_d and σ_d are mean and variance of d-th feature dimension along frames and so time. Additionally, C_{td} is the cepstral coefficients of d-th feature dimension and t-th frame.

When speech is corrupted by noise, both its statistical distribution and temporal structure are distorted. Hence, it is desirable to normalize the temporal structure of the features as well. The methods used in this paper to normalize the temporal structure of the features are discussed in the pre-processing section.

2.4 Model training

Speech recognition problem and in its special form, keyword spotting, can be considered as classification problems. Different classification approaches can be divided into two general categories; Generative and discriminative approaches. Generative approaches learn a model of joint probability p(x,y) of input signal x and class label y and make their predictions by Bayes rules to calculate p(y|x) and then picking the most likely class. In discriminative approaches, p(y|x) is directly computed, without considering any statistical conditions or limitation on observation space. Hidden Markov Model (HMM) is a sample of generative models commonly used in the field of speech recognition and keyword spotting. HMM based keyword spotting approaches are divided into three main groups; whole-word modeling (Rohlicek et al. 1989), phonetic-based approaches (Manos and Zue 1997) and Large Vocabulary Continuous Speech Recognition (LVCSR) based approaches (Szöke et al. 2005).

Despite their popularity, HMM-based approaches have several known drawbacks such as convergence of the training algorithm to local maxima and lack of accurate estimates in some cases due to insufficient observations. Another important weakness of HMM for keyword spotting is that it does not aim at maximizing the detection rate of the keywords, directly. In recent years, various approaches are presented for resolving some of these drawbacks. Maximum Mutual Information Estimation (MMIE) (Bahl et al. 1986), Minimum Classification Error (MCE) (Juang and Katagiri 1992) and Minimum Word Error (Povey and Woodland 2002) are some examples of these approaches using discriminative algorithms for training Hidden Markov Models. The following research investigated various discriminative training techniques and models such as neural networks (Chen et al. 2014; Fernández et al. 2007) and large-margin-based approaches (Keshet et al. 2009; Tabibian et al. 2013, 2014, 2016; Vapnik and Vapnik 1998). There is both theoretical and empirical evidence that discriminative approaches outperform generative approaches for the same task (Vapnik and Vapnik 1998). However, the feature extraction part in discriminative approaches is very important and has noticeable effects on the computational complexity and performance of the whole keyword spotting system.

In aerospace applications it is necessary to have a VCD system with high accuracy, quick response and low computation complexity. Although large-margin based and deep neural network approaches have higher accuracy compared with the HMM-based ones, in this paper, the HMM-based

approaches are used for training the phone/word models due to their quick response to the user, less computation complexity and simpler feature extraction phase. Moreover, due to the small size of the dictionary and suitable recorded dataset (discussed in the following sub-sections), HMM-based approaches could achieve high accuracy on this limit dictionary.

In the proposed phone-based VCD systems (Ph-VCD), each phone is represented by a simple left-to-right three-state HMM with 16 Gaussian mixtures per state. The HMM Toolkits (HTK) (Young et al. 1993) is used for implementing the HMM-based approaches. The phone-based HMMs are trained on the whole TIMIT database (5040 Sentences) (Lamel et al. 1989). The speakers, which communicate with the system, are non-native persons. However, the TIMIT database is a complete database contains utterances of speakers with eight different English accents. Since the train and test platform are different, the true detection rate of the VCD system will be degraded in real conditions. In order to compensate the differences between the test and train platforms, speaker adaptation methods are used to estimate new model parameters for new speakers. Maximum a Posteriori (MAP) (Vergyri et al. 2010) and Maximum Likelihood Linear Regression (MLLR) (Liu and Fung 2000) are two main approaches in speaker and accent adaptation. These two approaches have been used due to their efficiency and simple implementation (HMM toolkits provide a suitable platform for implementing these adaptation techniques).

If the number of target keywords is limited and a complete database exists, using word-based models instead of phone-based models will improve the performance of the VCD system. In the proposed word-based VCD system (Wrd-VCD), each word is represented by a simple left-to-right six-state HMM with 16 Gaussian mixtures per state. 47 target keyword models, one model for silence and one model for other words have been trained. The non-keyword model has been trained with all parts of utterances in the train set except the keywords and silence parts. The HMM-based keyword spotting approach is discussed in the next section. In the Wrd-VCD the adaptation phase has not been applied to word-based models.

2.5 Keyword spotting

Phone-based or word-based keyword spotting (KWS) is performed using phone-based filler model, as in (Shokri et al. 2011). The rule-based language model will be discussed in the next section.

2.6 Rule-based language model

The main contribution of this paper is the rule-based language model. The rule-based language model converts the VCD problem from whole speech recognition to spoken keyword spotting. As presented in the next section, each speaker utters 63 commands. These 63 commands contain 78 keywords. From these 78 keywords, because of the rule-based language model, the VCD system has to detect just 47 discriminative keywords (the keywords that could be assigned to a special voice command and discriminates accurately a voice command from other commands). Other 31 keywords will be labeled as non-keyword parts of speech utterances (it is not important what they are, exactly). This will decrease the computational complexity of the search algorithm and the number of word-based HMM models. Additionally, it speeds up both test and train phases. The discriminative keywords are the main features of the voice command decoding part. Table 1 shows the discriminative keyword(s) of each voice command. The rule-based language model works based on the content of this table.

As it is clear from Table 1, there are some voice commands that have the same discriminative keywords. For such voice commands, another keyword(s) have to be considered, which is (are) not shared between those commands. For example, keyword “launcher” is shared between commands 1 and 61. Thus, keyword “separation” and “parameters” are considered to discriminate these two commands. Since, commands 1 and 2 and commands 57, 58 and 61 differ in their first keywords (“launcher” and “nose” for commands 1 and 2—“biological”, “environmental” and “launcher” for commands 57, 58, 61), “launcher” should be considered as a discriminative keyword.

Sometimes, two commands have the same discriminative keywords with different locations. In such cases, the location of keywords discriminate two commands. For example, as Table 1 shows, both commands 46 and 54 have discriminative keyword “temperature”. However, in command 46, this keyword is located at the beginning of the command while it is located at the end of command 54. Therefore, in this case, the location of the keyword discriminates the two commands.

According to Table 1, the voice commands can be divided into three groups based on the number of their discriminative keywords. Commands type I are decoded using only one discriminative keyword. In the commands type II, two discriminative keywords are used for decoding them. Commands type III are decoded using three discriminative keywords. The rule-based language model is proposed based on these three types of commands. In addition, there are some exceptions for some voice commands. The general rules have been modified for these commands to cover those exceptions. The different exceptions types and the main general rules for decoding three types of commands and different types of exceptions are proposed in the following.

Table 1 Discriminative keyword(s) of input voice commands

Number	Voice command	Discriminative keyword(s)
1	Launcher separation	Launcher & separation
2	Nose separation	Nose
3	Open main parachute	Main
4	Open drogue parachute	Drogue
5	Open drogue chute	Drogue
6	Flight computer on	Flight & computer & on
7	Flight computer off	Flight & computer & off
8	Analogue video record on	Analogue & on
9	Analogue video record off	Analogue & off
10	Digital video record on	Digital & video & on
11	Digital video record off	Digital & video & off
12	Issuing commands on	Issuing & on
13	Issuing commands off	Issuing & off
14	Data acquisition on	Acquisition & on
15	Data acquisition off	Acquisition & off
16	Power supply on	Power & on
17	Power supply off	Power & off
18	Instrument on	Instrument & on
19	Instrument off	Instrument & off
20	Toolbox on	Toolbox & on
21	Toolbox off	Toolbox & off
22	Data telemetry on	Data & on
23	Data telemetry off	Data & off
24	Video telemetry on	Video & on
25	Video telemetry off	Video & off
26	Digital telemetry on	Digital & telemetry & on
27	Digital telemetry off	Digital & telemetry & off
28	Telecommand on	Telecommand & on
29	Telecommand off	Telecommand & off
30	GPS consolidated on	GPS or consolidated & on
31	GPS consolidated off	GPS or consolidated & off
32	Radio tracking on	Tracking & on
33	Radio tracking off	Tracking & off
34	Navigation on	Navigation & on
35	Navigation off	Navigation & off
36	Control computer flight on	Control & on
37	Control computer flight off	Control & off
38	Supervisor on	Supervisor & on
39	Supervisor off	Supervisor & off
40	Supervision and monitoring on	Monitoring & on
41	Supervision and monitoring off	Monitoring & off
42	Acoustic register on	Acoustic & on
43	Acoustic register off	Acoustic & off
44	Atmosphere control on	Atmosphere & on
45	Atmosphere control off	Atmosphere & off
46	Temperature management on	Temperature or management & on
47	Temperature management off	Temperature or management & off
48	Canister on	Canister & on
49	Canister off	Canister & off
50	Bio-Lab on	Bio-lab & on
51	Bio-Lab off	Bio-lab & off

Table 1 (continued)

Number	Voice command	Discriminative keyword(s)
52	Oxygen	Oxygen
53	Carbon Dioxide	carbon or dioxide
54	Internal temperature	Temperature
55	Moisture	Moisture
56	Pressure	Pressure
57	Biological parameters	Biological
58	Environmental parameters	Environmental
59	Flight path, navigation, control and guidance parameters	(Flight & navigation & parameters) or (flight & navigation & guidance) or (flight & control & parameters)
60	Flight profile parameters	Flight & parameters
61	Launcher parameters	Launcher & parameters
62	Display camera videos	Display or videos
63	Monitoring parameters	Monitoring & parameters

2.6.1 Different exceptions types

2.6.1.1 Exception 1 The command is determined due to just one discriminative keyword, but it contains more than one keyword. Some of them are discriminative keywords of other commands. In such cases, the rules are modified to cover this exception and decode the command, correctly.

2.6.1.2 Exception 2 Some discriminative keywords have very similar pronunciations. Thus, any recognition error (a deletion/ addition/ substitution) can convert these keywords to each other. For example, "drogue" and "analogue" have similar pronunciation. In addition, it is possible to detect "main" instead of "management". In these cases, the rules are modified to consider the probability of pronouncing the other discriminative keyword and thus, the other voice command. Therefore, it is possible to compensate the recognition errors in the detecting keywords. It means that if the detected keyword is not the same as the pronounced keyword, the input voice command is decoded, correctly.

2.6.2 Rules for commands type I

The main general rule for commands type I is as follows:

```
If (CurrentKeyword==discriminative keyword) && (there is no previous and
future detected keywords))
        DecodedCommand=table1.column2(row.column3(content ==
        discriminative keyword));
End
```

As it is clear from the above rule, when there is only one detected keyword (a discriminative keyword), the decoded command is the second column content of that row of Table 1, which its third column content is the same as the detected keyword.

2.6.3 Rules for commands type II

Commands of type II divided into two groups. Commands that contains on/off as the second discriminative keyword and commands that does not. For example, the command "launcher separation" is of the second group. The main general rule for commands type II is as follows:

```
If (CurrentKeyword==discriminative keyword1)
        If (there is no future detected keyword)
            DecodedCommand="Command is not detected";
                                    or
                                       table1.column2(row.column3(content==
                                       discriminative keyword1));
        else If (FutureKeyword==discriminative keyword2)
                DecodedCommand=table1.column2(row.column3(content==
                discriminative keyword1&discriminative keyword2));
              Else
                Store discriminative keyword 1 as previous keyword;
                Look for the suitable rule according to the second
                discriminative keyword;
        End
End
```

In the above rule, which is very general (in order to cover the exceptions, our rule-based language model is more special than this form), at the first condition the possibility of existing other detected keywords after the current one is checked. If there is no keyword, two cases can be considered. The second discriminative keyword is very effective for decoding the command or not. In the first case, the command is not detected. In the second case, the commands, which their discriminative keyword is the first keyword, will be put ahead. For example if the first discriminative keyword is "launcher" and there is no detected keyword after it, the input command will be decoded to "launcher separation"; because, it is expected the user to utter this command

more probably in comparison with command "launcher parameters". However, one can decode the input command to "launcher parameters" due to less risk of executing this command (if it is decoded wrongly).

If there is another discriminative keyword after the first one, the rule will check it. If it is discriminative keyword 2, as is expected, the input speech will be decoded to the command with discriminative keywords 1 and 2. Otherwise, discriminative keyword 1 will be stored and the rule-based language model searches the suitable rule according to the second discriminative keyword.

2.6.4 Rules for commands type III

Only seven commands of 63 commands are of type III. One of the rules for decoding these seven commands is mentioned here as an example for special rules which cover exceptions. For example, rule for commands 6 and 7 is as follows:

```
If (CurrentKeyword=='flight')
If(there is no future detected keywords)
    If (‘flight’ is not at the beginning of the command)
      If((PreviousKeywords=='computer')OR(PreviousKeywords== 'control')
      OR (PreviousKeywords == 'canister'))
            DecodedCommand='Control computer flight off';
        Else  If (there is no previous detected keywords)
            DecodedCommand='Control computer flight off';
        End
    Else
      DecodedCommand='Flight computer off';
    End
Else If (‘flight’ is not at the beginning of the command)
      If((PreviousKeywords=='computer')OR(PreviousKeywords== 'control')
      OR (PreviousKeywords == 'canister'))
          Store ‘flight’ as the previous keyword.
          Extract next detected keyword.
          If(CurrentKeyword== ‘on’))
            DecodedCommand='Control computer flight on';
          Else If((PreviousKeywords=='flight')AND(CurrentKeyword ==
          'parameters')
                DecodedCommand='flight profile parameters';
              Else
                DecodedCommand='Control computer flight off';
              End
          End
        Else
          Store ‘flight’ as the previous keyword.
        End
        Else
          Extract next detected keyword.
          If(CurrentKeyword=='on')
              DecodedCommand='Flight computer on';
          Else If(CurrentKeyword=='telemetry')
              Store ‘data’ as the previous keyword.
            Else
              DecodedCommand='Flight computer off';
                Store ‘flight’ as the previous keyword.
            End
            Go to the beginning of the rule-based language model.
          End
      End
    End
End
```

The main user interface of the VCD system is depicted in Fig. 2.

As Fig. 2 shows, the user interface of the VCD system is composed of four tabs. The first tab "system" gives user some information about the system. In addition, user can exit from the system via this tab. The second tab provides five possibilities for users: Adding all files from a selected archive, adding just one file, deleting some files from the loaded archive, resetting the list of files and getting a voice command from the microphone, in online mode. The third tab "Dictionary" provides the possibility of loading a predefined dictionary of keywords or adding some special keywords for the user. In addition, the user can reset the selected keywords list or delete some keywords from the list. The last tab "processing" is an option for running the system, plotting the results and listening to them (for evaluating the correctness of the decoded commands).

3 Non-native database for aerospace applications

Voice commands in a vehicular system are divided into two main groups. The sent commands from ground station to the vehicular system and inter vehicular system commands. In this paper, the most well-known commands are considered as depicted in Table 1.

The mentioned commands contain five flight commands (commands 1–5), 46 sub-system commands (commands 6–51), five capsule commands (commands 52–56) and seven panel commands (command 57–63). As said in the previous section, just 47 keywords from all 78 keywords have been selected as the discriminative keywords. These 47 discriminative keywords are "Computer, Video, Videos, Launcher, Separation, On, Off, Nose, Main, Drogue, Analogue, Digital, Issuing, Commands, Acquisition, Data, Power, Supply, Instrument, Toolbox, Telemetry, Telecommand, GPS, Consolidated, Radio, Tracking, Navigation, Control, Supervisor, Monitoring, Acoustic, Atmosphere, Temperature, Management, Canister, Bio-Lab, Oxygen, Carbon, Dioxide, Moisture, Pressure, Parameters, Biological, Environmental, Guidance, Display, Flight. Since the VCD system is evaluated in a non-native platform (Persian speakers who speak in English), and there is not any available non-native database contains the mentioned commands, a non-native database is recorded for training and testing the proposed system.

In our application, speakers utter the voice commands discretely and formally. Thus, the type of speech in our work is formal speech. It is one of the simplest forms of speech in comparison with the continuous and spontaneous speech. Different speakers have different properties. Their ages, education level and gender are the most important properties in preparing a spoken database. There are 20 speakers (10 male and 10 female speakers). Their age has a range from 16 to 54. The education level of these 20 speakers are various from student to M. S. (Master of Science). In Table 2, (A)

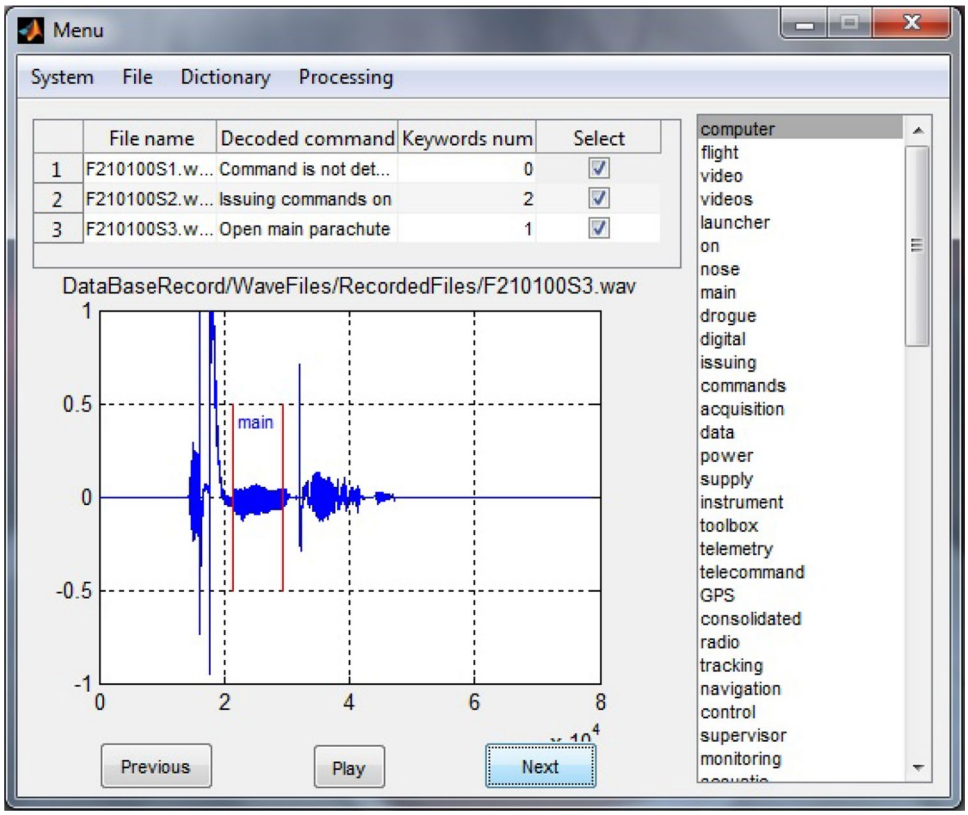

Fig. 2 The main user interface for the VCD system

Table 2 Experimental conditions

Train set for phone-based HMMs	The whole TIMIT database (train set and test set except SA1 and SA2 in each directory-about 5040 spoken utterances)
Train set for word-based HMMs	The spoken utterances of 16 speakers of the non-native database (2192 spoken utterances equal to 3 h speech)
Test set	The spoken commands of the other 4 speakers of the non-native database (252 voice commands)
Minimum keyword length	3 phonemes
Maximum keyword length	14 phonemes
Number of discriminative keywords	47
Number of voice commands for each speaker	63
Feature vector extracted from each speech frame	39 MFCC normalized through CMVN method
Number of states of each phone-based HMM	3 states + 2 emitting states
Number of states of each word-based HMM	6 states + 2 emitting states
Number of Gaussian Mixture Models (GMMs) in each state	16
Garbage model	Left-to-right 3 state HMM with 16 Gaussian mixtures per state
Method for calculating the score of the garbage model for a special keyword	Normal harmonic [3]

S. and (B) S. are abbreviating forms of Associate and Bachelor of Science, respectively. Each speaker utters 137 commands and keywords. Thus, 2740 spoken utterances have been recorded. Since each utterance has duration equal to 5 s, the duration of the whole database is 3 h and 48 min.

One of the main parts of preparing a spoken database is segmentation. The main goal of speech segmentation is determining the boundary between different segments of utterances. These segments could be phone, diphone, triphone, syllables, word and other meaningful units. Word unit is considered as the main unit of segmentation in this project. Labeling the prepared non-native database based on phone unit, is a very time consuming and costly task. Additionally, it is uneconomical to do this time consuming

task for a dataset customized for a very special application. Thus, a very small fraction of the database (about 10 min) is labeled based on phone unit and is used for adapting the phone-based HMMs (trained on TIMIT) according to the non-native database properties. 100 sentences of the non-native database [Five sentences of each speaker (from 20 speakers presented in Table 2)] make up the 10-min adaptation set. Those five sentences are selected for adaptation, which contains all possible phones that there exist phone models correspond to them. In this paper, the manual segmentation is used for segmenting and labeling the spoken sentences. The manual segmentation refers to a process in which, an expert segments and labels spoken utterances. He/she uses only the spectrogram form of the utterances and the voice content of them.

The whole database has been labeled in word and a fraction of it (about 10 min) is labeled in phone. The phone-labeled fraction will be used for adaptation. In order to record the database, a user interface has been implemented in MATLAB. The main window of this user interface is presented in Fig. 3.

The user has to enter his/her ID, which is a unique code. Then, he/she determines his/her gender and clicks "Record next command". A dialogue box informs the user to utter the voice command for about 5 s and another dialogue box informs him/her to stop. If the user needs to know the list of sentences and how pronounce them, he/she can click "help". Then, two windows will be opened. One contains the whole sentences list and the other makes it possible for user to select a special sentence and listen to its pronunciation. After completing the recording for a user, an expert listens to them to delete the extra and false spoken utterances and wants the user to utter again some sentences, if is necessary.

Fig. 3 The user interface for dataset recording

4 Experimental results

In this section, the results of evaluating the proposed framework for VCD are discussed. All systems have been tested on the prepared non-native database. The phone-based and word-based HMMs have been trained on TIMIT (Lamel et al. 1989) and the prepared non-native database, respectively. The experiments conditions are presented in Table 2 and the implemented methods are discussed in Table 3.

For evaluating the implemented methods, four evaluation measures have been used: True Detection Rate (TDR), False Alarm Rate (FAR), Miss Rate (MR) and Real Time Factor (RTF).

True detection rate in the field of VCD refers to decode a voice command, truly. It calculates the ratio of the number of true decoded commands to the whole number of voice commands. The following equation shows this calculation:

$$\text{TDR} = \text{Total True decoded commands} / \text{Total number of voice commands} \quad (2)$$

False alarm rate in the field of VCD refers to decode a voice command, falsely. It calculates the ratio of the number of false detection of voice commands to the whole number of voice commands. The following equation shows this calculation:

$$\text{FAR} = \text{Total False decoded commands} / \text{Total number of voice commands} \quad (3)$$

Miss rate refers to the number of non-detected voice commands to the whole number of voice commands. The following equation shows this calculation:

$$\text{MR} = \text{Total number of non-decoded commands} / \text{Total number of voice commands} \quad (4)$$

RTF is a common metric of measuring the speed of an automatic speech recognition system. If it takes time P to process an input of duration I, RTF is computed as:

$$RTF = P/I \quad (5)$$

The proposed VCD system is evaluated in two ways. First, the results of using adaptation methods are compared

Table 3 Implemented methods

Method name	Description
Ph-VCD	The proposed phone-based VCD system
Ph-VCD-MAP	The proposed Ph-VCD adapted based on MAP
Ph-VCD-MLLR	The proposed Ph-VCD adapted based on MLLR
Wrd-VCD	The proposed word-based VCD system

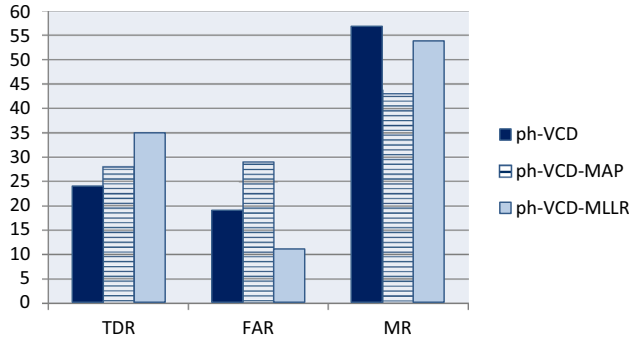

Fig. 4 The evaluation results for Ph-VCD based on TDR, FAR and MR

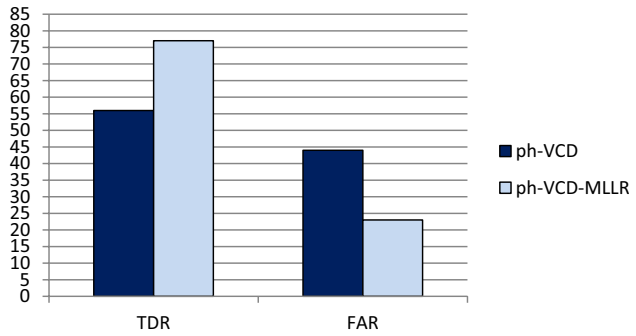

Fig. 5 The evaluation results for Ph-VCD (after pronouncing not-detected commands) based on TDR, FAR

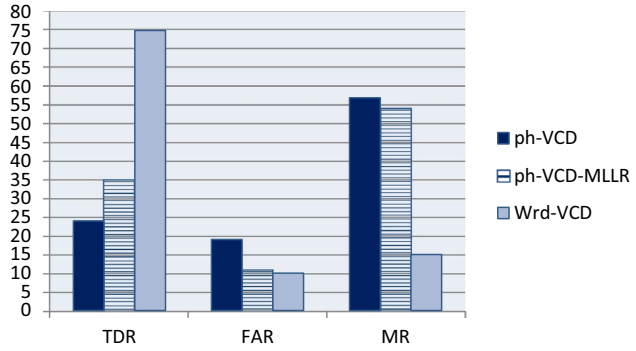

Fig. 6 The evaluation results for Wrd-VCD system based on TDR, FAR and MR

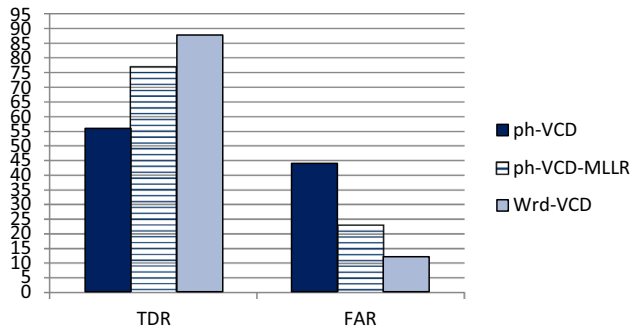

Fig. 7 The evaluation results for Wrd-VCD (after pronouncing not-detected commands) based on TDR, FAR

with those of Ph-VCD method without adaptation. Then, the best adaptation method is compared with the Wrd-VCD. These evaluations are presented in the following of this section.

4.1 The results of evaluating adaptation methods

Figure 4 shows the evaluation results for three methods (Ph-VCD, ph-VCD-MAP and Ph-VCD-MLLR) based on three evaluation measures (TDR, FAR, MR). When a miss rate is occurred, the speaker could pronounce the command again and expect that this time the VCD system detect it. Thus, the TDR and FAR are calculated for the cases that speaker is permitted to pronounce the miss commands, repeatedly, until all of the commands will be detected. These calculations have been done based on statistical conditions. The results are depicted in Fig. 5. Since the MAP adaptation method performed weaker than the MLLR method, just the results of MLLR adaptation method have been reported.

As Figs. 4 and 5 shows, the ph-VCD system has a very low TDR and FAR due to the differences between the test and train platforms. Using adaptation methods will improve the performance of the system. However, the MAP adaptation just improves the TDR and not the FAR. The MLLR method increases TDR about 21%, for the case that the speakers repeat the miss commands repeatedly, until they will be detected. Additionally, it decreases the FAR about 21% for the case that the speakers repeat the miss commands repeatedly, until they will be detected.

4.2 The results of evaluating Wrd-VCD

Figure 6 shows the evaluation results for three methods (Ph-VCD, ph-VCD-MLLR and Wrd-VCD) based on three evaluation measures (TDR, FAR, MR). Again, the TDR and FAR for the case that speaker is permitted to pronounce the miss commands repeatedly, until all of the commands will be detected, have been calculated. The results are presented in Fig. 7.

As Figs. 6 and 7 shows, the Wrd-VCD system has a very higher performance in comparison to Ph-VCD and Ph-VCD-MLLR. It is due to this matter that the train and test platforms in Wrd-VCD system are the same. The TDR of the Wrd-VCD system is about 40% greater than that of the Ph-VCD-MLLR system. Although the FAR of the Wrd-VCD

system is not improved, the MR of the system decreased, considerably (about 42%).

Based on time complexity evaluations, the RTF of the Ph-VCD is about 6.7 times faster than real time. It means that each 5-s commands takes about 0.75 s to detect. The RTF of the Wrd-VCD is about 4.2 times faster than real time. Each 5-s command in the Wrd-VCD takes about 1.2 s to detect. Thus, the Wrd-VCD is about 0.45 s slower than the phone-based one. However, both systems response considerably faster than real time with an acceptable false alarm rate.

The proposed framework in this paper differs from the other VCD systems (Fezari et al. 2012; Gupta et al. 2014; Hoque et al. 2014; Morris et al. 1993; Weinstein 1995) in the number and length of the voice commands. Although, it decodes more commands (at least six times more) with longer length (at least two times more), it performs as well as the other VCD systems.

5 Conclusion

In this paper, a VCD system for aerospace applications is proposed. The VCD system is composed of two parts: the keyword spotter and the voice command decoder. The output of the first part, which is the input to the second part, is some discriminative keywords with their exact locations in the input voice command. In the second part, a rule-based language model is proposed which decodes the input command based on the discriminative keywords and their locations. The output of the second part is the output of the system: the decoded voice commands. In addition to these contributions, a non-native database for aerospace applications is prepared which is about 3 h and 48 s. The whole database is labeled based on word unit. A fraction of the database (about 10 min) is labeled based on phone unit for adapting the phone-based HMMs trained on TIMIT (a native English database).

The mentioned keyword spotter is trained based on phone and word units using HMM-based method. The phone-based keyword spotter is trained on TIMIT and then, adapted using MLLR and MAP adaptation methods to compensate the differences between train and test platforms. The word-based keyword spotter is trained on 3 h of the prepared non-native database. Ph-VCD and its adapted versions (Ph-VCD-MAP and Ph-VCD-MLLR) and Wrd-VCD are evaluated using four evaluation measures: True detection rate (TDR), false alarm rate (FAR), miss rate (MR) and real time factor (RTF). The experimental results show that the Wrd-VCD decodes the voice commands with true detection rate equal to 88% and false alarm rate equal to 12% in average. Additionally, using speaker adaptation methods in the Ph-VCD improves the true detection and false alarm rates about 21 and 21%, respectively. Based on time complexity evaluations, each five-second commands takes about 0.75 and 1.2 s, in the word-based and Ph-VCD, respectively, to detect. Thus, the Wrd-VCD is about 0.45 s slower than the phone-based one. However, both systems response considerably faster than real time with an acceptable false alarm rate. Compared with the other existent VCD systems, although the proposed VCD system decodes more commands (at least 6 times more) with longer length [at least two times longer (if "word" is considered as the length unit)], it performs as well as them.

The keyword spotter plays an important role in the performance of the VCD system. Improving the detection rate of the keyword spotting part using better features or exploiting more accurate training methods, for example discriminative methods, will be considered in the future works. Additionally, increasing the variety of the database utterances will result in more complete dataset, which will be used for training more accurate keyword spotter. Preparing more suitable database will be considered in the future works.

References

Ahmed, A., Ahmed, T., Ullah, M., et al. (2012) Controlling and securing a digital home using multiple sensor based perception system integrated with mobile and voice technology. arXiv preprint arXiv:1209.5420.

Bahl, L., Brown, P., De Souza, P., et al. (1986) Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'86* (pp. 49–52). IEEE.

Benayed, Y., Fohr, D., Haton, J. P., et al. (2003a) Improving the performance of a keyword spotting system by using support vector machines. In *2003 IEEE Workshop on Automatic Speech Recognition and Understanding, 2003. ASRU'03.* (pp. 145–149). IEEE.

Butt, M., Khanam, M., Khiyal, M., Khan, A., et al. (2011) Controlling home appliances remotely through voice command. *(IJACSA) International Journal of Advanced Computer Science and Applications, Special Issue on Wireless and Mobile Networks*, 35–39. doi:10.14569/SpecialIssue.2011.010206.

Chen, C.-P., Bilmes, J. A., & Kirchhoff, K. (2002) Low-resource noise-robust feature post-processing on AURORA 2.0. In *Seventh International Conference on Spoken Language Processing*.

Chen, G., Parada, C., & Heigold, G. (2014) Small-footprint keyword spotting using deep neural networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4087–4091). IEEE.

Cornu, E., Destrez, N., Dufaux, A., et al. (2002) An ultra low power, ultra miniature voice command system based on hidden markov models. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (pp. IV-3800–IV-3803). IEEE.

Fernández, S., Graves, A., & Schmidhuber, J. (2007) An application of recurrent neural networks to discriminative keyword spotting. In *International Conference on Artificial Neural Networks* (pp. 220–229). Berlin: Springer.

Fezari, M., Boumaza, M. S., & Aldahoud, A. (2012) Voice command system based on pipelining classifiers GMM-HMM. In *2012*

International Conference on Information Technology and e-Services (ICITeS) (pp. 1–6). IEEE.

Fezari, M., & Bousbia-Salah, M. (2006) A voice command system for autonomous robots guidance. In *9th IEEE International Workshop on Advanced Motion Control* (pp. 261–265.). IEEE.

Firdaus, A. M., Yusof, R. M., Saharul, A., et al. (2015) Controlling an electric car starter system through voice. *International Journal of Science & Technology Research, 4*(4), 5–9.

Gupta, A., Patel, N., & Khan, S. (2014) Automatic speech recognition technique for voice command. In *2014 International Conference on Science Engineering and Management Research (ICSEMR)* (pp. 1–5). IEEE.

Hoque, E., Dickerson, R. F., & Stankovic, J. A. (2014) Vocal-diary: A voice command based ground truth collection system for activity recognition. In *Proceedings of the Wireless Health 2014 on National Institutes of Health* (pp. 1–6). ACM.

Juang, B.-H., & Katagiri, S. (1992). Discriminative learning for minimum error classification (pattern recognition). *IEEE Transactions on signal processing, 40*, 3043–3054.

Keshet, J., Grangier, D., & Bengio, S. (2009). Discriminative keyword spotting. *Speech Communication, 51*, 317–329.

Lamel, L. F., Kassel, R. H., & Seneff, S. (1989) Speech database development: Design and analysis of the acoustic-phonetic corpus. In *Speech Input/Output Assessment and Speech Databases*.

Li, J., Deng, L., Gong, Y., et al. (2014) An overview of noise-robust automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22*, 745–777.

Liu, W. K., & Fung, P. N. (2000) MLLR-based accent model adaptation without accented data. In *Sixth International Conference on Spoken Language Processing (ICSLP 2000), Beijing*.

Manikandan, M., Araghuram, S. D., Vignesh, S., et al. (2015). Device control using voice recognition in wireless smart home system. *International Journal of Innovative Research in Computer and Communication Engineering, 3*, 104–108.

Manos, A. S., & Zue, V. W. (1997) A segment-based wordspotter using phonetic filler models. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997. ICASSP-97* (pp. 899–902). IEEE.

Morris, R. B., Whitmore, M., & Adam, S. C. (1993). How well does voice interaction work in space? *IEEE Aerospace and Electronic Systems Magazine, 8*, 26–31.

Mporas, I., Ganchev, T., Siafarikas, M., et al. (2007). Comparison of speech features on the speech recognition task. *Journal of Computer Science, 3*, 608–616.

Ngo, K., Spriet, A., Moonen, M., et al. (2012). A combined multichannel Wiener filter-based noise reduction and dynamic range compression in hearing aids. *Signal Processing, 92*, 417–426.

Özkartal, S. G. (2015). Development of a system for human language commands and control for a quadcopter application. *Journal of Management Research, 7*, 1.

Povey, D., & Woodland, P. C. (2002) Minimum phone error and I-smoothing for improved discriminative training. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (pp. I-105–I-108). IEEE.

Principi, E., Squartini, S., Bonfigli, R., et al. (2015). An integrated system for voice command recognition and emergency detection based on audio signals. *Expert Systems with Applications, 42*, 5668–5683.

Rohlicek, J. R., Russell, W., Roukos, S., et al. (1989) Continuous hidden Markov modeling for speaker-independent word spotting. In *1989 International Conference on Acoustics, Speech, and Signal Processing, 1989. ICASSP-89* (pp. 627–630). IEEE.

Shokri, A., Tabibian, S., Akbari, A., et al. (2011) A robust keyword spotting system for Persian conversational telephone speech using feature and score normalization and ARMA filter. In *2011 IEEE GCC Conference and Exhibition (GCC)* (pp. 497–500). IEEE.

Szöke, I., Schwarz, P., Matejka, P., et al. (2005) Comparison of keyword spotting approaches for informal continuous speech. In *Interspeech* (pp. 633–636). Citeseer.

Tabibian, S., Akbari, A., & Nasersharif, B. (2013). Keyword spotting using an evolutionary-based classifier and discriminative features. *Engineering Applications of Artificial Intelligence, 26*, 1660–1670.

Tabibian, S., Akbari, A., & Nasersharif, B. (2014). Extension of a kernel-based classifier for discriminative spoken keyword spotting. *Neural processing letters, 39*, 195–218.

Tabibian, S., Akbari, A., & Nasersharif, B. (2015). Speech enhancement using a wavelet thresholding method based on symmetric Kullback–Leibler divergence. *Signal Processing, 106*, 184–197.

Tabibian, S., Akbari, A., & Nasersharif, B. (2016). A fast hierarchical search algorithm for discriminative keyword spotting. *Information Sciences, 336*, 45–59.

Tranter, S., Yu, K., Everinann, G., et al. (2004) Generating and evaluating segmentations for automatic speech recognition of conversational telephone speech. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP'04)* (p. I-753.). IEEE,

Vapnik, V. N., & Vapnik, V. (1998) Statistical learning theory. New York: Wiley.

Vaseghi, S. V. (2008) Advanced digital signal processing and noise reduction. Hoboken: Wiley.

Vergyri, D., Lamel, L., & Gauvain, J.-L. (2010) Automatic speech recognition of multiple accented English data (pp. 1652–1655). In *INTERSPEECH*.

Viikki, O., Bye, D., & Laurila, K. (1998) A recursive feature vector normalization approach for robust speech recognition in noise. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 733–736). IEEE.

Wang, R., Shen, Z., Zhang, H., & Leung, C. (2015) Follow me: A personal robotic companion system for the elderly. *International Journal of Information Technology (IJIT), 21*(1).

Watile, Y., Ghotkar, P., & Rohankar, B. (2015) Computer control with voice command using matlab. Computer, doi:10.17148/IJIREEICE.2015.3613.

Weinstein, C. J. (1995) Military and government applications of human-machine communication by voice. Proceedings of the National Academy of Sciences 92:10011–10016.

Yoshizawa, S., Hayasaka, N., Wada, N., et al. (2004) Cepstral gain normalization for noise robust speech recognition. In *Proceedings.(ICASSP'04). IEEE International Conference on Acoustics, Speech, and Signal Processing* (vol. 201, p. I-209-212). IEEE.

Young, S. J., Woodland, P., & Byrne, W. (1993) HTK: Hidden Markov Model Toolkit V1. 5. Washington D.C.: Cambridge University Engineering Department Speech Group and Entropic Research Laboratories Inc.