Hybrid of Neural Networks and Hidden Markov Models as a modern approach to speech recognition systems

Paweł Sokólski, Tomasz Rutkowski

Department of Control Systems Engineering, Faculty of Electrical and Control Engineering,
Gdańsk University of Technology, Poland

Abstract: The aim of this paper is to present a hybrid algorithm that combines the advantages of artificial neural networks and hidden Markov models in speech recognition for control purposes. The scope of the paper includes review of currently used solutions, description and analysis of implementation of selected artificial neural network (NN) structures and hidden Markov models (HMM). The main part of the paper consists of a description of development and implementation of a hybrid algorithm of speech recognition using NN and HMM and presentation of verification of correctness results.

Keywords: artificial neural networks, hidden Markov models, MFCC, speech recognition, control

Nowadays, a fast and reliable communication with electrical equipment plays important role. Despite the fact that the easiest and most intuitive form of communication and command is speech, the most common in communication with the devices are methods based on mechanical effects on the control such as keyboard, or joystick. The current knowledge allows the realization of voice control systems, which was not possible a few years ago. Therefore, there is a need to develop more efficient methods of human speech recognition, to ensure the reliability of communication between man and machine. This paper describes an approach using mel-cepstral coefficients (MFCC) as the basis for the analysis of the speech signal. Based on the resulting signal characterizing factors are identified appropriate elements (sounds / words) using a hybrid algorithm, combining the benefits of artificial neural network (NN) and hidden Markov models (HMM).

1. Speech signal and its

1.1. Speech signal

Speech is a complex acoustic signal (fig. 1). In human communication through speech a lot of information in addition to the dictionary meaning of words is provided. Intonation, speaking rate, and the pitch of the voice and emotional attitude determine the speaker's intentions. Any change in intonation or speed alters the nature of the signal and make it difficult to compare with the pattern. Due to intonation and speech rate changes it is difficult to analyze the signal neither in the time nor the frequency domain. An important problem is the existence of noise and ambient sound or speech of others. Despite the many difficulties which are inherent to computer analysis of the speech signal and automatic recognition, speech has unquestionable advantages as a part of human-machine interface. Four main advantages of communication by means of speech are specified [1]:

- speed of operation (statement can be formulated efficiently than any manipulation),
- no operator bound to any desktop, a set of keypads, keyboards, etc.,
- ability to efficient operate in the dark, under overload conditions, physical or mental stress,
- the natural and comfort control, releasing from having long-term training and apprenticeship.

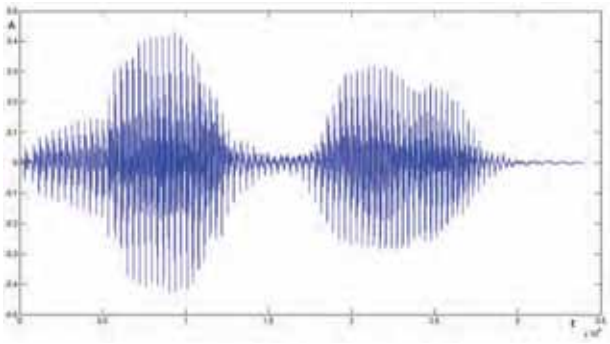

Fig. 1. Speech signal for the word “lewo” (left) [2]
Rys. 1. Sygnał mowy słowa “lewo” [2]

It turns out that the benefits of voice control are so significant that the methods and algorithms of speech recognition are very popular and are rapidly developed. The presence of noise, interference and distortion makes the signal analysis less accurate. It is not possible to register the ideal speech element pattern, because in the process of detection only imperfect pattern always available [1]. Therefore, sound recording in the highest quality has great practical importance in order to best analyze the characteristics of the speech signal and correctly classify it on the basis of the individual elements of speech. The speech signal can be divided into parts of varying duration, such as sentences, words, syllables, or sounds. There

are different approaches to dividing speech elements [2]. In this paper discussed are only two of them: the division of the whole word (in relation to sentences) and the division into phonemes (in relation to words). According to one definition, created for the purpose of technical considerations, phonemes are "class of sounds, the differences between which are purely personal (ie, due to the differences in pronunciation of individual persons) or contextual (ie, arising from the relation between a phone and it preceding phoneme or optionally the following one)" (Sapożkow MA, 1966). One phoneme can correspond to one letter, one letter may correspond to a specific number of phonemes and their collection, given phoneme can correspond to some of the letters or their collection. In the Polish language about 40 phonemes can be distinguished (with 32 letters of the alphabet) (Sapożkow MA, 1966).

Hybrid algorithms presented in this paper are used to recognize whole sentences (commands consisting of several words) and individual words. For example, two following sentences were analyzed: "thirty degrees to the right" and "sixty degrees to the left". While, examples of recognized individual words are "left" and "right".

1.2. Cepstral analysis

Due to the difficulties in the areas of speech analysis in time and frequency domains caused by changes in intonation and pace of speaking, it is necessary to use other methods that allow to specify the set of parameters that can be used in the speech recognition process. For this purpose the cepstral analysis was used. Cepstral analysis is defined as a result of the inverse Fourier trans-formats of the logarithm of the original signal spectrum. The purpose of this is the transition from the convolution of signals with the use of the Fourier transform (convolution in the time domain corresponds to multiplication in the frequency domain) to the sum of the logarithms of these signals in the domain of pseudo-time. Writing a mathematical transformation of the final form presented as a formula 1.

$$\hat{X}(T) = F^{-1}[\ln | F[x(t)] |] \tag{1}$$

where:

t – time
$F[x(t)]$ – Fourier transform of the function $x(t)$
$F^{-1}[X(f)]$ – Inverse Fourier transform of a function $X(f)$
$\hat{X}$ – the function result of the cepstral analysis

A property of the logarithm, whereby the multiplication is replaced by addition, is used. Application of the logarithm of the inverse transform of the signal in the frequency domain can be obtained in the time domain (pseudo-time) signal which is not signal convolution, but the sum of the signals (logarithm of the inverse transform spectra) of its composition, which helps separate analysis of the component signals.

1.3. Mel-cepstrum and mel-cepstral coefficients (MFCC)

Mel scale is the scale of the subjective assessment of sound. According to the Weber-Fechner law [1] subjectively perceived by the human being is not the arithmetic difference of stimuli, but their attitude. This causes that the subjectively perceived pitch is different from the frequency measured objectively. This means that twice the sound frequency is heard as sound twice as high. On this basis, the frequency scale which corresponds to the perception of the human ear was created. Because the human ear perception is selective calculation of the signal cepstrum associated with prolongation of the calculation and the addition of noise in the form of a large amount of irrelevant information. Cepstral speech signal analysis of the performed only for frequencies audible to the human ear is called mel-cepstral analysis. This approach can reduce the amount of data to be analyzed at the same time to a form which include all relevant information.

Cepstrum value for a given period for a given mel frequency is called mel-frequency cepstral coefficient – MFCC (fig. 2). Analyzing n frequencies n coefficients that characterize the part of speech are calculated (fig. 3).

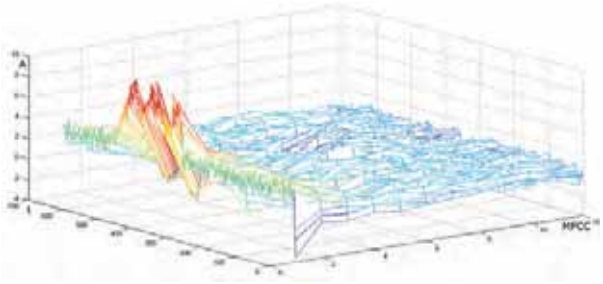

Fig. 2. Mel-frequency cepstrum of the word "lewo" (left) [2]
Rys. 2. Mel-cepstrum sygnału słowa „lewo" [2]

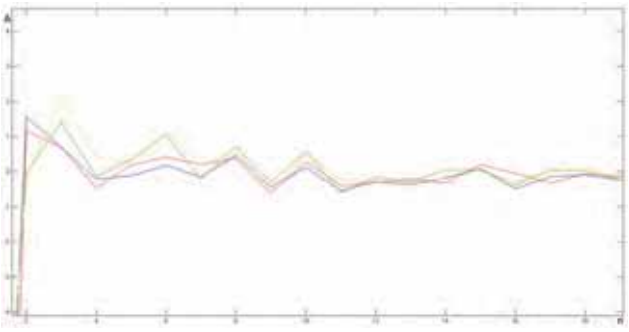

Fig. 3. Vector of mel-frequency cepsral coefficients (MFCC) for the word "lewo" (left) [2]
Rys. 3. Wektor współczynników mel-cepstralnych dla słowa „lewo" [2]

In order to process the raw speech signal for the algorithms presented in the following section, the following parameters related to the windowing signal were used:

- the length of the window 3 ms,
- window shift 1.5 ms,
- Hamming window,
- 20 MFCC coefficients determined on the window.

2. Artificial neural networks (NN)

2.1. Examined structures

Artificial neural networks were used for isolated word/phoneme recognition purposes. Neural networks are designed to the great extent reminiscent of human brain activity. They are able to "learn" to recognize patterns impossible to be described mathematically. What's more, they have the ability to generalize knowledge, which means that they can generalize the knowledge about patterns presented to them, to recognize patterns in samples which was not previously presented to them. Neural networks have a strong ability to recognize patterns of sound, and their ability to generalize knowledge can help to improve the quality of speech recognition systems. As network input vector mel-frequency coefficients, delta and double delta coefficients (first and second derivative of MFCC) characterizing individual words together with the logarithm of the signal energy were selected.

Two approaches to speech recognition using NN were used: recognition of isolated words as a whole, and recognition of individual phonemes for the use of the HMM algorithm to implement the hybrid NN-HMM discussed later in this article.

In order to select the neural network, which would be the most effective in the diagnosis of speech following structures were analyzed: neural network LVQ (Learning Vector Quantization) and neural network RBF (Radial Bias Function).

Network input vector consisted of MFCC together with the known derivatives. Output assigns the word to one of the defined by user classes. Sound samples recorded by one person in 50 samples per word (50 recordings/word as a training set and a set of 10 recordings/word as a set used for verification) were used to learn networks. A number of tests for the considered structures were carried out.

Operation of different network structures discussed above was tested in MATLAB [2]. Tests were performed using:

- LVQ Network: 61 inputs (20 MFCC, their first and second derivative and a logarithm of signal energy), 240 neurons in the competition layer, 8 outputs;
- RBF network: 61 inputs, 240 neurons, 8 outputs.

Tab. 1. Words recognition by LVQ and RBF networks results [2]
Tab. 1. Skuteczność rozpoznawania sieci LVQ i RBF [2]

	LVQ	RBF
Lewo	5	6
Prawo	5	4
Start	9	1
Stop	8	10
Stopni	2	10
Trzydzieści	9	0
Sześćdziesiąt	9	8
Dziewięćdziesiąt	8	6
AVERAGE:	6.87	5.62
Percentage of recognition:	69 %	56 %

The results obtained in the simulation are summarized in table 1 (detailed description of the network parameters and methods of learning can be found in [2]). As it can be seen, the highest percentage of recognition has LVQ network. During the tests it was noted that in the latter case some of the words radial network matched LVQ networks, or they were even better. The mean score indicates, however, the superiority of LVQ networks. Consequently, this network has been used in the later work in both isolated word recognition, and the individual phonemes recognition.

2.2. LVQ neural network

The LVQ network structure [3, 4] is presented in fig. 2.2. It is composed of input vector, competitive and linear layer and output vector.

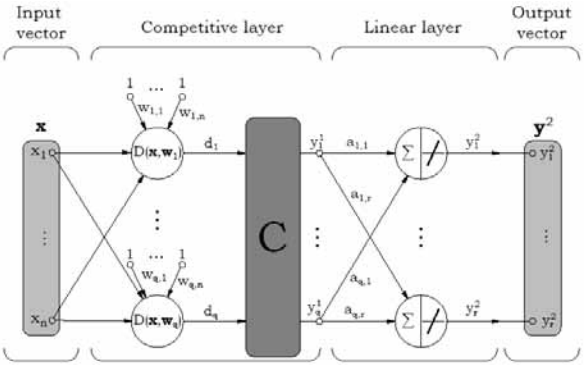

Fig. 4. LVQ network structure [3]
Rys. 4. Struktura sieci LVQ [3]

The competitive layer is composed of q neurons and is used to cluster input data into the $\Re^n$ input subspace, where n is number of elements in the input vector x. Position of the j-th cluster centre in the $\Re^n$ subspace is determined by the corresponding neuron weights $w_i = [w_{1,j},...,w_{n,j}]^T$. The Winner Takes All algorithm is used to calculate the competitive layer output $y^1 = [y_1{}^1,...,y_q{}^1]^T$. It is obtained by comparing the calculated distances $D(x, w_j)$ between input vector and weights of each neuron. The output corresponding to the winning neuron (minimum distance in Euclidean metric sense) is activated (it's value is set to 1), all other outputs are set to 0. In the linear layer clusters defined in the hidden layer are assigned to the output classes $y^2 = [y_1{}^2,...,y_r{}^2]^T$ where r is the number of network outputs. The outputs classes and are weights in linear layer are defined by user. That provides possibility that multiple clusters may be assigned to the one LVQ network output. Therefore it is possible to properly classify linearly inseparable data.

The Winner Takes All is used as LVQ network learning technique with Euclidean metric (it is most commonly used). For a given set of the i-th input data (x_i, t_i) is calculated metric distance D to the prototype, which is the weights vector w_j. The winner prototype w_j, which is closest to the input x_i in accordance to considered metric, is attracted to x_i if belongs to a class t_i or repelled

otherwise. Thus, the winner prototype w_i is updated according to the rule (2).

$$w_j = w_j \pm \alpha(x_i - w_i) \tag{2}$$

where: α — controls the rate of the convergence of the algorithm.

The sign in the rule (2) is positive, when the winner prototype belongs to the proper class, and negative otherwise.

3. Markov models (MM) and hidden Markov models (HMM)

3.1. Markov models (MM)

Markov Model represents a sequence of object states in with the structure of the transitions between them. Changing of state can take place only in accordance to the direction of the arrow lines on the graph. Thus impossible transitions are eliminated from consideration, which significantly simplifies the space in which may be a solution. In recognition of human speech, in order to simplify, it is assumed that each new state only depends on the actual one, and not on the previous state. In fact, it may be different, however, taking it into account significantly complicates calculations used in the analysis of models.

With this structure, it is possible to reduce the number of possible commands. Sequences of words not belonging to the model are ignored. This approach increases the effectiveness of the algorithm and protects against unwanted command execution (eg, an attempt to execute such a command, which further part of the algorithm could correctly interpret, but for some reason being outside the Markov model).

Figure 5 shows an example, simplified Markov model showing the way of state transition in example command sentence (polish language).

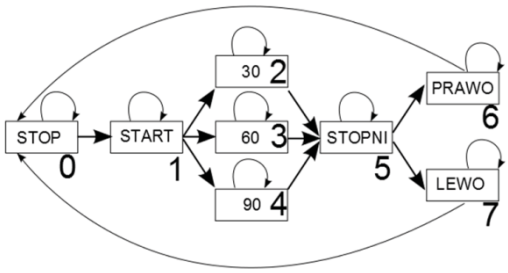

Fig. 5 Sample Markov model [2]
Rys. 5. Przykładowy model Markowa [2]

3.2. Hidden Markov models (HMM)

Markov models in case the state in which the property is located, and there is only a set of certain observations related to possible states is known are called the Hidden Markov The models (HMM – Hidden Markov Models called) [5]. In the case of HMM only to the probability of being in a given state, the probability of changes and the likelihood of observing the data for individual states are known. On this basis, with a number of observations in succession it is possible to determine the sequence of states which have the highest probability of occurrence.

It is possible to modeling variability of the speech signal in time, i.e. to determine the most likely order of various elements of speech (phonemes, sounds, words), and on this basis to say that the word was uttered with the greatest probability. We do not know what word the speaker said, but a number of observed sets of parameters is known. It is known (through research, analysis of samples of speech, etc.) with what probability parameters (observations) correspond to a given phoneme, and what is the probability that in a given word one phoneme occurs after the another (probability of a state change). On this basis, it can be specified which of the words with the highest probability corresponds to the observed sequence of speech parameters such as MFCC. This allows the model changes over time, and also take into account a word prolonged in time (the probability of remaining in a state for a specified period of time). It appears that not all phonemes occur with the same probability [1], making it possible to identify the more likely sequence. This feature speaks for the use of probability-based methods, such as HMM, for speech recognition.

Another feature of hidden Markov models is their ability to "learn" arrays of probabilities. Using suitable algorithms based on the training data in the form of strings of observations arrays are modified iteratively so that the response of the model coincides to the actual one. Figure 6 shows an example, simplified hidden Markov model showing the way of state transition in time with defined probability of state transition.

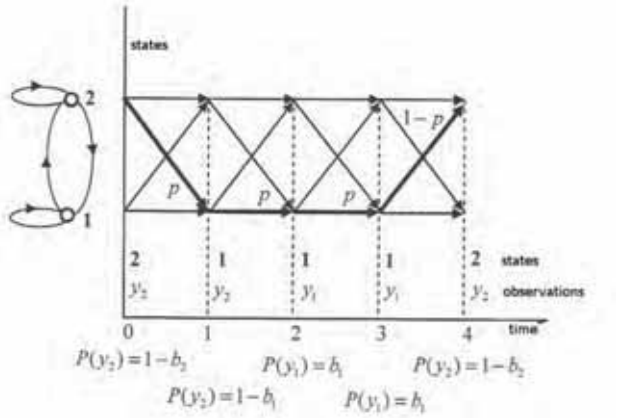

Fig. 6. Sample HMM structure – state transition in time [5]
Rys. 6. Przykładowy ukryty model Markowa – zmiana stanu w czasie [5]

4. Hybrid algorithm (NN-HMM)

Currently, hidden Markov models are the most commonly used method in speech recognition systems. They can perfectly model the time dependence between individual elements of speech. However, it appears that they have some limitations, which set a limit in increasing the efficiency of speech recognition. Therefore, the idea of hybrid

algorithms that combine the advantages of HMM and neural network with the ability to recognize complex patterns of sound was developed [6]. With the synergistic combination of these two algorithms it may be possible to improve the ability of automatic recognition of human speech. In one approach a neural network "learns" to recognize the different elements of speech, such as words, syllables, or phonemes. Using the extended training set, respectively, due to the ability of generalization knowledge, it is possible to achieve high recognition performance. Hidden Markov models can be "taught" the probability of occurrence of the sequence of states by statistical analysis of the training set. In this way, having a large amount of data, it is possible to obtain such weights variation of artificial neural networks and hidden Markov models probabilities to best the way acoustic and time dependences are modeled. This approach, being based on "learning", as opposed to the theoretical knowledge-based design, allows to catch subtle differences between the patterns and to grant insensitivity to unimportant differences occurring in them.

The big advantage of the hybrid NN-HMM algorithm is that hidden Markov models can be taught not with use of ideal sequence of vectors corresponding to the correctly identified elements of speech, but the sequence of vectors which are the actual responses of neural networks. This allows the HMM to learn to recognize correctly input data vectors in which the error occurred due to the action of NN. While training HMM the probability of recognizing a wrongly categorized speech element by the neural network is taught. This is a double protection against making a mistake and if an artificial neural network fails to recognize the patterns presented to it, but its effectiveness is good enough to teach HMM, a hybrid NN-HMM should be enable to recognize much higher percentage of voice commands than in the case of each of its component algorithms separately. A large percentage of NN errors may, however, make it not possible to distinguish between sets of different inputs and to teach hidden Markov models, which can reduce the effectiveness of the algorithm.

As previously mentioned, it is necessary to teach HMM with a comprehensive training set. Another problem resulting from a combination of NN and HMM algorithms is the use of hidden Markov models with incorrectly specified probability matrices, which may result from the use of too little training data. Because of that, despite the high efficiency of recognition of individual elements of the neural networks, sequences are poorly recognized by the HMM.

This paper presents two approaches: the entire command recognition by HMM on the basis of the recognition of isolated words recognition by NN and recognizing whole-word by HMM basing on the phonemes identified by the NN.

4.1. Command recognition from isolated words

In the first variant of the algorithm Markov models are responsible for determining the sentence which is most likely to have been spoken. The use of HMM for command recognition should allow to eliminate errors in distinguishing between successive words. However, it is not possible to increase the effectiveness of word recognition in the branches of a parallel model such as "left" and "right". These words have the same probability of occurrence in a given position in the command. While the NN-HMM algorithm should properly recognize the command, in which the recognition error occurred in not parallel branches, in the case where words with the same probability of occurrence are situated parallel in the model, recognition quality depends entirely on neural networks.

If the recognized word is the only one occurring on the specific place in the model, in which other word appearance is not possible, it is not possible in any way to increase the likelihood that the spoken command is different. This means that the wrong word recognition by artificial neural network in this case will not affect the outcome of the whole recognition algorithm.

As previously mentioned, the behavior of the HMM-based system has advantages over simple Markov models with strictly defined conditions of the transition between states. Only fully correct sequence would allow further analysis. While using hidden Markov models the occurrence of such irregularities is likely to result in the correct recognition.

Wrong word recognition by the neural network does not make it impossible to recognize the entire sequence by the hidden Markov models. In this approach to hybrid algorithms (NN recognize words, HMM construct command), as mentioned, a major problem is the case where alternatives are equally probable to appear. The mechanism based on the higher probability of the sequence is not able to determine which answer is correct, which may even have a negative impact on the recognition process.

In some cases there is no advantage in the operation of the hybrid algorithm but it allows, in the relation to the whole recognition problem, to compensate for errors certain. HMM do not have the actual probability matrices, and only "taught" their counterparts. The result is that in the process of learning one option may have a slightly higher probability than the other and will be favored during recognition process. HMM are taught based on the responses of neural networks, in which a different probability of a given state may occur. The problem of distinguishing between states with the same probability of occurrence is a drawback of hidden Markov models.

4.2. Recognition of words from phonemes

In the second approach individual phonemes in spoken word are analyzed. This solution is widely used in speech recognition systems. Division of words into phonemes can increase recognition accuracy, and can facilitate the introduction of new words to the dictionary of the recognition system. For the purposes of this paper was assumed that only 10 phonemes corresponding to the letters 'l', 'e', 'w', 'o', 's', 't', 'p', 'a', 'r', 'n', 'i', will be recognized, which should allow to construct a few simple words. The signal of silence lasting for a specified period of time was also added, which also can be seen as a form of a phoneme.

Thus, even if the fragment does not contain a speech signal, it will be classified by the system as a "phoneme of silence".

Split into individual phonemes was based on their duration. Knowing how many phonemes consists of the word, its duration is divided into the number of pieces and analyzed. Such an approach for extracting phonemes causes that not in all words phonemes are separated in the same way. Analyzed signal segments can be too short or too long. However, this is the easiest way to extract the corresponding phonemes and therefore has been used in the work.

Even if no phoneme is recognized by a neural network, hidden Markov models, basing on the length of the word, and the probability of phoneme occurrence, were able to correctly identify the sequence. It turns out that the fact that the majority of phonemes belongs to the word (advantage of at least one phoneme) is often enough to make for an equal length and probability of its occurrence is higher (even though many of the phonemes have not been completely identified). There is a possibility of error in favor of another word of the same length. Sometimes, the word is recognized correctly due only to the value of HMM probability matrix, which was founded by learning and does not reflect the exact probabilities that occur in reality. However, as the learned hidden Markov models were not based on actual sequence of phonemes (the actual probability of phonemes occurrence), but the response vectors generated by LVQ network, the answer is that the network response sequence is more likely to correspond to a given word. On the overall probability of recognition the matrix of the probability of transitions, which determine the probability of transitions between states has impact, and not only a probability of particular sequence of states (not all models of all states – words occur, and thus the likelihood of their occurrence in the taught model is negligible) is important. This means that for the same set of phonemes their sequence matters. Therefore, even if the neural network always or almost always be fails to identify a phoneme, but its position in the sequence, that is, the probability of transition into or out of (the transition of states for a particular set of states) is characteristic for a given word, HMM learn to recognize these patterns of transitions between phonemes and will correctly identify the word.

However, as mentioned earlier, HMM are taught not on the basis of the actual sequence, but based on the network response and the important is not only what phonemes were recognized, but also in what order. This means that with the same number of phonemes correctly identified by the neural network in the sequence of two of the most likely solutions, the result can be very different, depending on network errors. This illustrates the significant impact of the network being "wrong" on the solution. HMM "learns" patterns of errors and probabilities of errors during the recognition of the word. The use of large amounts of training data from the output of the network (the need for large amounts of training data to accurately calculate the probability matrix) can be very important in the correct recognition in situations when sequences recognized by the NN are similar.

4.3. Efficacy of NN-HMM algorithms

Both of the above approaches to hybrid algorithms combining artificial neural networks and hidden Markov models were implemented and verified in MATLAB. On the basis of voice recordings several attempts of recognition are made, and the results are summarized in the tables. Table 2 shows the effectiveness of word recognition of the first variant of the algorithm that is algorithm recognizing commands basing on isolated words.

Tab. 2. The effectiveness of recognizing sentences from words
Tab. 2. Skuteczność rozpoznawania poleceń na podstawie słów

Sentence	Words NN	Commands NN-HMM
30 stopni lewo	78 %	100 %
60 stopni lewo	34 %	70 %
60 stopni prawo	54 %	10 %

As it can be seen, despite the fact that artificial neural networks recognize only 78 % of the words presented to them in the command "30 degrees left", a hybrid algorithm recognized 100% of commands. The situation is the same for the next command, "60 degrees left", where the NN recognized 34%, while NN-HMM had 70 % effectiveness. This shows the benefits of combination of NN and HMM. Hidden Markov models may recognize sequences despite the relatively large errors of artificial neural networks.

Command "60 degrees right", however is an interesting case, which is recognized by the NN-HMM algorithm correctly only in 10 % of the cases while the artificial neural network to recognize 54 % of the words. This is due to the already mentioned drawback of hidden Markov models, involving the difficulty in contrast sequences of different states with equal probability of occurrence. In this case, most of the sample commands "60 degrees right" is classified as "60 degrees left", which is equally probable. This defect was the cause of developing another version of the algorithm, in which neural networks do not recognize the words, but the phonemes, and hidden Markov models make recognition of whole words.

Table 3 shows the results for the second version of the algorithm. As can be seen, also in this case the effectiveness of the recognition by the hybrid algorithm is greater than the neural networks alone.

Tab. 3. The effectiveness of recognizing words from phonemes
Tab. 3. Skuteczność rozpoznawania słów na podstawie fonemów

word	phonemes NN	Word NN-HMM
lewo	68 %	90 %
prawo	51 %	70 %

However, in contrast to the word recognition based on isolated words, individual combinations of phonemes are more complex and confusing issue of equiprobable states do not occur, or occurs to a small extent.

The results illustrate the benefits of the hybrid NN and HMM algorithms and support the use of this type of algorithms in recognizing of human speech.

5. Summary

People have communicated with each other by speech for the millennia. The possibility of using the ease with which they use it and intuitiveness of this method to communicate in effective control allows to issue commands and expanse usability of many devices.

Currently used speech recognition algorithms use the hidden Markov models or neural networks. This paper presents an example of hybrid approach combining the advantages of these methods. Hybrid algorithm, free of NN or HMM defects, allows to increase the efficiency of speech recognition. In this way it is possible to issue commands more reliable, and thus extend the applications of this type of control. But it turns out that the algorithm of this type has its own drawbacks, the existence of which must be taken into account during the design.

The paper specifies the situations in which the use of such algorithms is justified and their drawbacks, which make them inefficient in some applications.

It is important that the base of samples used for learning neural networks (and hidden Markov models on the base of NN answers) was large enough to properly "teach" NN-HMM algorithm. To allow the practical application of this algorithm it would be necessary to collect a large number of recordings, including recordings from different speakers.

The results obtained in this study fall within the scope of Master thesis at the Faculty of Electrical and Control Engineering of the Technical University of Gdansk [2].

5.1. Bibliography

1. Tadeusiewicz R., *Sygnał mowy*, WKŁ, Warszawa, 1988.
2. Sokólski P., *Sieci neuronowe i modele Markowa jako elementy hybrydowego algorytmu rozpoznawania mowy dla potrzeb zadań sterowania głosem*, Politechnika Gdańska (Gdańsk University of Technology), Gdańsk 2012.
3. T. Kohonen, *Learning vector quantization Neural Networks*, Vol. 1, suppl. 1, 1988.
4. Hagan, M.T., H.B. Demuth, and M.H. Beale, *Neural Network Design*, Boston, MA: PWS Publishing, 1996.
5. Kwiatkowski W., *Metody automatycznego rozpoznawania wzorców*, WAT, Warszawa 2001.
6. Tebelskis J., *Speech Recognition using Neural Networks*, Carnegie Mellon Univiersity, Pittsburg 1995. ■

Hybryda sieci neuronowych i ukrytych modeli Markowa jako nowoczesne podejście do rozpoznawania mowy

Streszczenie: Celem artykułu jest przedstawienie algorytmów hybrydowych łączących zalety sztucznych sieci neuronowych i ukrytych modeli Markowa w zastosowaniach rozpoznawania mowy dla potrzeb sterowania. W zakres opracowania wchodzi przegląd stosowanych obecnie rozwiązań, opis i analiza implementacji wybranych struktur sieci neuronowych (NN) oraz ukrytych modeli Markowa (HMM). Główną część artykułu stanowi opis opracowywania hybrydowego algorytmu rozpoznawania mowy wykorzystującego NN i HMM oraz prezentacja wyników weryfikacji poprawności działania.

Słowa kluczowe: sztuczne sieci neuronowe, ukryte modele Markowa, MFCC, sterowanie

Paweł Sokólski, MSc Eng

PhD candidate at Gdansk University of Technology at the faculty of Electrical and Control Engineering and student of Mechatronics at the faculty of Mechanical Engineering. In professional work focuses on industrial informatics and control engineering in cathodic protection.

e-mail: sokolski.p@gmail.com

Tomasz Rutkowski, PhD Eng

Received his PhD degree in automatic control from the Faculty of Electrical and Control Engineering of the Gdańsk University of Technology. His current research interests include advanced control algorithms, estimation algorithms, computational intelligence techniques and industrial control systems.

e-mail: t.rutkowski@eia.pg.gda.pl