ECONTECHMOD. AN INTERNATIONAL QUARTERLY JOURNAL – 2019, Vol. 08, No. 4, 19 – 27

Identification of Bee Colony Acoustic Signals using the Dynamic Time Warping Algorithm

Yu. Polyniak, D. Fedasyuk, T. Marusenkova

Lviv Polytechnic National University, Lviv, Ukraine
yurii.polyniak@gmail.com, fedasyuk@ gmail.com, tetyana.marus@gmail.com

Received November, 18 2019: accepted January, 11 2019

Abstract**.** The work deals with automated recognition of the current state of a bee colony, for continuous monitoring of processes running in a bee hive is of key importance in beekeeping. The dynamic time warping algorithm is considered as a method of analyzing acoustic signals produced by a bee colony. Upon such an analysis one can make inferences about the current state of the colony. We have developed a software module for audio-signal identification, which is to be used as a part of an automated bee colony monitoring system, and a software tool for verification of the module. We evaluated the efficacy of the algorithm, the probability of bee colony states correctly recognized using acoustic signals produced by the colony and consumed computational resources by the example of a queen bee's sounds recorded during swarming. The dependencies of the signal processing time and the successful pattern recognition probabilities on the frame sample rate and frame size are presented.

Keywords**:** Acoustic Signal, Signal Recognition, Dynamic Time Warping, Fast Fourier Transform, Automated Bee Colony Monitoring.

1 INTRODUCTION

A number of complex processes are constantly running in a bee hive. Some of them require an immediate reaction and assistance of a beekeeper to solve potential problems. An automated bee colony monitoring system might replace a long-term vigilance of a human beekeeper and thus help detect upcoming critical situations. Therefore, issues of synthesizing methods and tools for development of such a system are of key importance for beekeeping.

Today a variety of bee colony monitoring systems exist. However, most of them are limited to tracking parameters like temperature, humidity and carbon oxide which are typically not sufficient for correct identification of processes currently running in a hive. In order to enhance the efficacy in recognition of events likely to happen in a bee hive some systems analyze sounds as well. Such an analysis usually boils down to evaluation of frequency distribution, which proves to be insufficient as well. There exist systems that implement more sophisticated signal identification methods. However, these systems focus on some specific states of a bee colony (for instance, swarming) and are not applicable for identification of other states.

The work is aimed at determining the process of bee sounds recognition. We discuss details of adaptation of the dynamic time warping algorithm to the problem of identifying bee colony states by sounds the colony produces.

2 LITERATURE OVERVIEW AND PROBLEM STATEMENT

One of the first bee colony monitoring systems was Apidictor [1]. The system incorporates a low-pass filter for detecting a change in sounds that happens two-three weeks before swarming. Ferrari and his colleagues [2] analyzed sounds recorded using a computer sound card and multidirectional microphones. They also tracked temperature and humidity. During last decade monitoring systems have evolved substantially due to wireless sensor networks [3]. The latter consist of embedded systems that acquire data from different sensors, process the data and send them to a computer connected to a remote database. Kviesis et al [4] presented a sample implementation of such a network whose main device, a local server, interacts with each device placed in a hive and thus serves as a gateway. Rybochkin [5, 6] has published a series of papers on the subject of beekeeping automation. He considered conditions under which acoustic noise produced by a bee colony can be considered a stationary ergodic random signal. In [7], in order to calculate the power distribution of a short signal, the author used analog frequency filtering with selection of the most informative frequency range. Then the whole frequency range under scrutiny was divided into several frequency bands. The spectral density was averaged for all those bands. Different states of a bee colony were then identified by comparison of the averaged spectral densities for all the bands. Qandour and Ahmad [8] presented a process of audio-signal identification that assumes frequency filters and fast Fourier transform (FFT) to be used for preparation of a signal spectrogram. Later on, they extract some important characteristics from the spectrum like the peak frequency, spectral centroid and others that are used by classification methods. A similar approach is used by Cejrowski [9], who applied machine learning algorithms to processing acoustic signals (including support vector machine algorithm). Meikle [10] proposed a system that performs monitoring upon readings from a set of different sensors, including prerecorded sounds and vibrations of a hive.

We are aimed at development of a software module based on dynamic programming methods that would allow us to analyze sounds produced by a bee colony. Besides, the efficacy of the software module should be verified on a set of prerecorded sounds of a queen bee during swarming.

3 ACOUSTIC SIGNALS ANALYSIS OVERVIEW

The structure of the acoustic signal analyzing process can be represented by an activity diagram (Fig. 1).

The chosen acoustic signal processing method assumes the following steps to be taken. An input audio signal is represented as a byte stream split into fixed-size vectors called frames. Frames are queued for processing as they become available. Then the frame vector is multiplied by some selected window function and passed to the Butterworth filter. Then the vector is subjected to FFT, which results in a vector of complex numbers that contain spectral characteristics for the corresponding frequencies. The complex number modulus of each vector item represents the magnitude whereas its argument denotes the phase. In order to reduce the size of data being analyzed we first ignore the signal out of the target range, and after that calculate the amplitude for each frequency. The obtained vector of magnitudes is normalized and quantized in accordance with the preset amount of levels. For comparing the prepared magnitude vector with signal patterns we use the dynamic time warping (DTW) algorithm. As a result, for each sample we obtain a DTW distance between it and one of the samples. The least DTW distance indicates the pattern most similar to the audio-fragment being analyzed. We consider the frames to be conforming the sought-for signal if their distances do not exceed the preset threshold value.

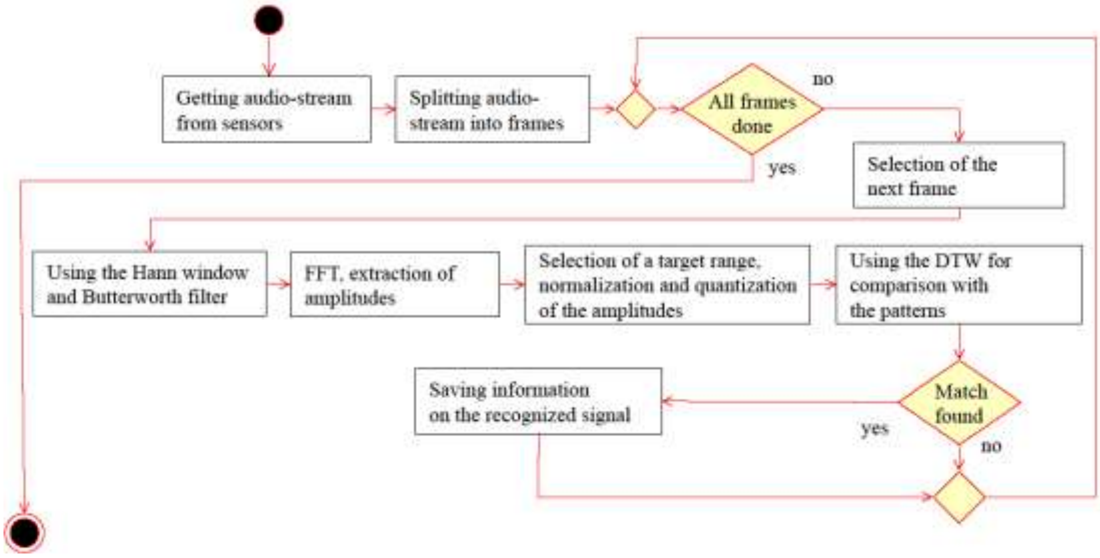

Fig. 1. An activity diagram for acoustic signal recognition

A fast Fourier Transform (FFT) algorithm is a substantial analytic method of processing audio signals. FFT is an optimized version of the discrete Fourier transform (DFT). In contrast to the main DFT implementation whose computational complexity is $O(N^2)$, FFT has a complexity reduced to $O(N \log N)$.

As a result, we will have fragments extracted from an audio-signal being analyzed for which spectra will be determined. Two algorithms parameters are of key importance: 1) sample frequency, fs, which is the rate of sampling a time-continuous audio-signal during its time-discretization, and 2) the block length, N_{FFT} – the number of samples included into a fragment being analyzed. For FFT the block length is 2^N (N is some natural number).

The throughput (Nyquist frequency), *fn*, determines the maximum frequency to be found by the FFT algorithm:

$$fn = \frac{fs}{2}. \quad (1)$$

The time constant, *TC*, is determined on the basis of the preset discretization frequency and block length:

$$TC = \frac{N_{FFT}}{fs}. \quad (2)$$

The frequency resolution, *df*, defines the distance between adjacent detected frequencies. Alternatively, it can be considered as the amount of FFT blocks formed out of a one-second-long fragment of an audio-signal being analyzed:

$$df = \frac{fs}{N_{FFT}}. \quad (3)$$

The sample frequency depends on the signal being analyzed whereas the block length can be set as an algorithm parameter. The sample frequency resolution is another FFT parameter. The bigger the FFT block length, the more accurate the resulting frequency spectrum but the slower the whole computation process. On the contrary, the smaller the FFT block length is, the faster FFT can be accomplished. Another way of speeding up the process is reduction of the sample frequency resolution by omitting some part of frames.

DFT assumes that the measured signal is an integer number of periods, so the two endpoints of the time waveform are interpreted as if they were connected together. In practice this assumption is unlikely to be fulfilled, i.e. the signal values at the ends of each block do not coincide. These discontinuities between signal blocks cause spectral leakages. A spectral leakage is an appearance of parasite frequencies actually absent from the original signal. In order to fight discontinuities and avoid spectral leakages, one applies a window function. A window function is a function that has zero values everywhere outside some specific interval. Multiplying a signal being investigated by a window function, one obtains a modified signal, zeroed everywhere outside the specified interval. Thus the signal values at the ends of blocks are forced to be equal. On the other hand, use of windows means the loss of signal information, since the parts of the signal in the middle of each window matter more than the signal at the boundaries of the window.

One should place high emphasis on comparison between spectrograms and the patterns. Here we propose to benefit from the Dynamic Time Warping algorithm (DTW for short). The DTW algorithm was initially developed for speech recognition in 1960s, then spanned across a wide range of areas. To name a few, today it is used in computer vision and computer animation, data mining, online signature matching, gestures recognition, etc. The algorithm "aligns" two temporal sequences, i.e. minimizes the effects of shifting and distortion in time between them and thus allows us to detect their similarity. As the algorithm name implies, two-time series under scrutiny are "warped" non-linearly in the time dimension to figure out a measure of their similarity. The optimal sequence of transformations between the two-time series will be computed (a so called warping path), along with the distances between them.

Let us take two numerical sequences, (a_1, a_2, ..., a_n) and (b_1, b_2, ..., b_m) whose length are not necessarily equal. The DTW algorithm starts with computation of pairwise distances between the time series. By "distances" they usually mean Euclidean distances, however, alternative metrics may be used instead.

Finding similarity may be achieved by comparing the numerical forms or spectrograms of the signals. In any case comparison assumes compensation for the unequal lengths of the sequences and non-linear nature of sound. The DTW remedies the stated problems by finding such a deformation that ensures the optimal distance between two time series of unequal lengths. In this work we make use of the DTW variant for comparison of two spectrograms.

The method assumes an audio-signal to be split into a number of fixed-size intervals (frames). For each frame its frequency spectrum will be found using FFT. The spectrum of a frame is a numerical vector whose index and value represent the frequency and its corresponding amplitude, respectively. By turns, a spectrum vector for each frame being analyzed is compared with spectrums of the patterns. As a result, we obtain a set of DTW distances that characterize similarity between the frame and each of the patterns. Smaller distances indicate closer similarity. Therefore, we select the least distance. If the latter does not exceed some preset threshold, we conclude that the frame matches the corresponding pattern.

As of many dynamic programming algorithms, the complexity of DTW algorithm is $O(N^{2v})$ where N – is the sequence length and v is the size of the dictionary. In the case of large sequences two problems rise: storage of bulky spectrogram matrices and numerous computations of distances.

In order to speed up the algorithm and enhance its efficacy at recognition of bee sounds, one analyzes only the frequency range containing frequencies typical of the sought-for signals instead of considering the whole spectrum resulting from the FFT. Besides, the amplitudes for the frequencies from the selected subrange are normalized and quantized. Therefore, time series will be compared within a fixed set of possible spectrum values.

There exists an enhanced version of the DTW algorithm, called FastDWT, which provides solutions to the above-stated problems. The solution consists in splitting the state matrix into 2, 4, 8, 16 and so on smaller matrices by iterative dividing the input sequence into two parts. Thus, all calculations are done to these smaller matrices and warping paths computed for them.

The DTW algorithm has its drawbacks. Firstly, the complexity $O(N^{2v})$ is not suitable for bulky dictionaries. The size of the dictionary influences the outcome a recognition process. Not surprisingly, the larger dictionary is, the more chances of successful recognition results are. Simply put, the DTW has poor scalability. Secondly, it is difficult to single out two items out of two different sequences if one takes into account the existence of multiple channels with different characteristics. Nevertheless, the DTW remains a simple algorithm, available for improvements and applicable to problems of recognizing relatively simple sounds.

4 IMPLEMENTATION OF A RECOGNITION PROCESS RESULTS

Selection of optimal analysis methods is of key importance for a bee colony monitoring system. The system should be able to process an input audio-signal in real time and find matches with predefined patterns corresponding to different bee colony states. The whole analysis process is expected to be performed by an embedded system directly. A remote server should receive the pattern recognition results. Such architecture conforms to the approach of local data analysis.

In accordance with the goal of this work, we are aimed at developing a software module for analysis of bee colony acoustic signals (Fig. 2) in order to identify the current state of the colony (for instance, swarming). The module being developed should be able to deal both with a continuous audio-stream and prerecorded audio-fragments. Besides, the module should be presented as a library, which can be included in a bee colony monitoring system. Since the module is not to be used independently, we have developed an additional Windows-compliant PC tool for its analysis and verification.

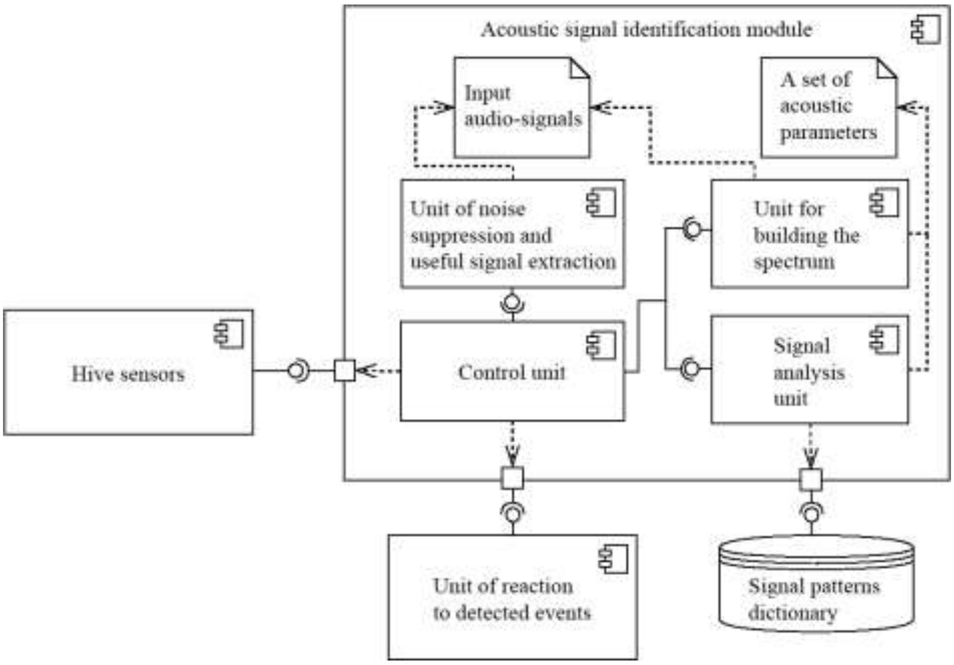

Fig. 2. The structure of the software module for acoustic signals recognition

As a result, the application depicts the audio-fragment that has been processed (Fig. 3) as a spectrogram (Fig. 4). The values of spectrum amplitudes are represented as a bitmap image where the horizontal pixel position is determined by the sequential frame number (the frame position in time), its vertical position corresponds to the spectrum frequency and its brightness is calculated upon the amplitude corresponding to the frequency. The coordinate grid represents the time increase of 1 second along the horizontal axis and the frequency increase of 1000 Hz along the vertical one.

The fragments of audio-signals whose DTW-distances do not reach the threshold are marked as levels in the upper part of the spectrogram. Higher levels indicate smaller DTW-distances, which means that the fragment under scrutiny gets closer to one of the patterns from the dictionary.

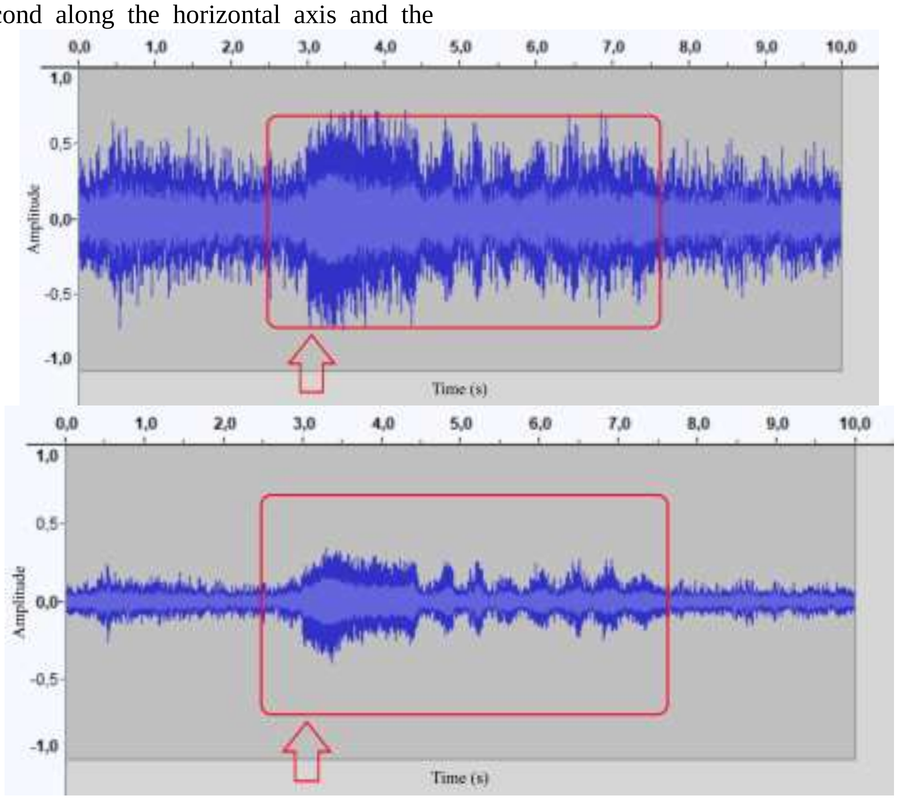

Fig. 3. Representation of a signal being analyzed as a wave (before and after noise filtering)

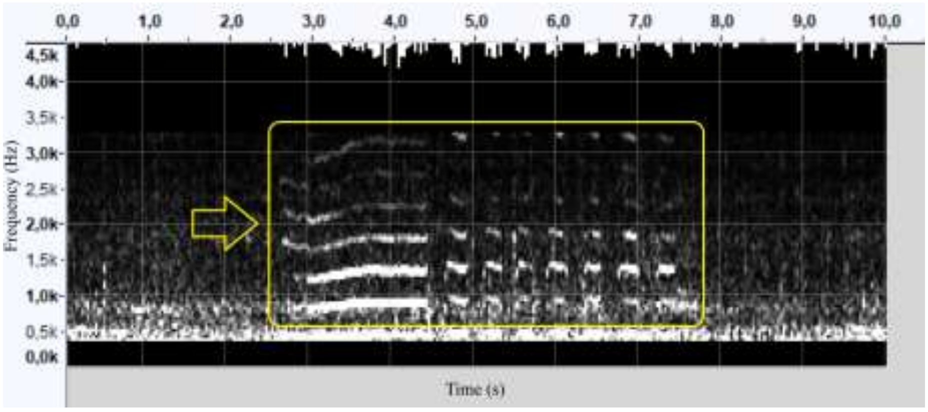

Fig. 4. The spectrogram of the processed audio-file fragment

5 INVESTIGATION INTO THE PATTERN RECOGNITION PROCESS AND ANALYSIS OF THE OBTAINED RESULTS

Having analyzed the efficacy of the developed software module for recognizing patterns in audio-signals, one can evaluate whether usage of the DTW algorithm for bee colony monitoring is applicable or not. Thus, we have evaluated the efficacy of our software module over a set of pre-recorded test audio-files under different parameters of the recognition algorithms. We considered the following parameters: 1) frame size for the FFT algorithm, 2) frame sample frequency, 3) size of a pattern set for the DTW algorithm, 4) the threshold value for the DTW algorithm.

The basic metrics to be analyzed are the signal processing time (at the fixed selected hardware resources), and the ratios N_α/N_S and N_β/N, where N_α is an amount of successfully recognized acoustic signal blocks, N_β is the amount of false alarms, N_S is the amount of signal blocks, N is the amount of blocks being analyzed.

The probabilities of the errors of type I (α) and type II (β) correspondingly are determined by the following formulae:

$$\alpha = 1 - \frac{N_\alpha}{N_S} \tag{4}$$

$$\beta = \frac{N_\beta}{N} \tag{5}$$

The probability of correct signal recognizing *p(S)* can be evaluated by the formula:

$$p(S) = 1 - (\alpha + \beta) \tag{6}$$

During our experiments we have processed four audio-files of the same length, which contained acoustic signals of queen bees in four different hives. The signals differ by their noise-to-signal ratios, presence of parasitic signals and signal strengths (Fig. 5). The regions containing the acoustic signals of queen bees are marked on the file spectrograms.

We have investigated into the efficacy of the process of recognizing patterns in acoustic signals for different FFT frame sizes. The bigger an FFT frame size, the larger data bulk is used for forming each frame. As a result, data processing slows down. On the other hand, the resolution of a spectrogram increases, which means that the probability of successful pattern recognition increases as well. Each test audio-record was processed by the developed software module at the size of an FFT block 128 to 2048 samples (the signal is split into frames with the frequency 30 frames per second). We are seeking for the minimum allowable FFT block size at which it is still possible to achieve an applicable pattern recognition probability. The experimental results for each audio-record are summarized in Table 1.

We have calculated the ratios between successfully and incorrectly recognized acoustic signal blocks and the whole analyzed audio-signal under scrutiny, along with the processing time. The detected dependencies are depicted in Fig. 6 and Fig. 7. It was detected that the FFT frame size of 1024 samples is sufficient for the probability of successful recognition equal to 80.42%. Using a block of 2018 samples, one can increase the pattern recognition probability by 5.44% while deteriorating the processing time by 95.1%. On the other hand, usage of blocks with sizes less than 1024 samples reduces the pattern recognition probability to less than 70%, which is a much unsatisfying result. Upon the obtained figures we conclude that the best tradeoff between the processing speed and successful recognition probability is an FFT frame size of 1024 samples.

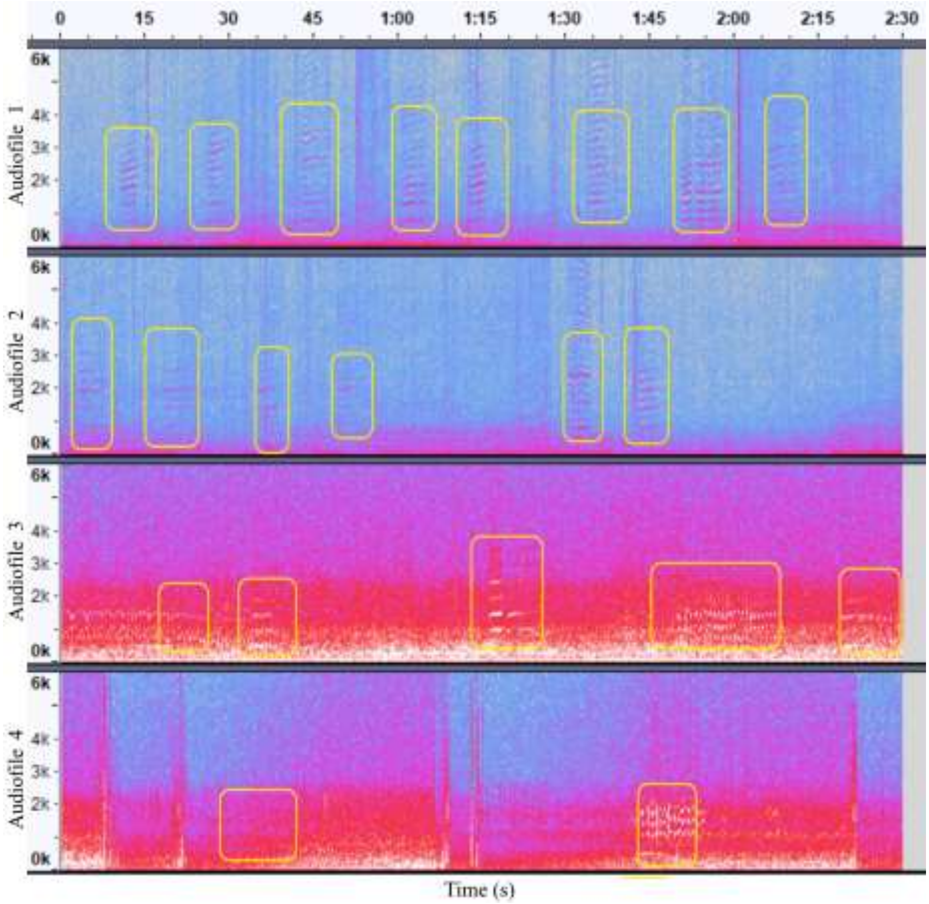

Fig. 5. Spectrograms of test audio-files

Table 1. The results of investigation into the efficacy of the software module at different FFT frame sizes.

Frame size / Amount of sought-for frames	Processing duration, s	Amount of properly recognized frames, N_α	Amount of false matches, N_β	***Probability of recognition, pS %***	Processing duration, s	Amount of properly recognized frames, N_α	Amount of false matches, N_β	***Probability of recognition, pS %***
	File 1				File 2			
	543				304			
128	2.84	320	41	***58.02***	2.60	192	53	***61.98***
256	3.17	348	43	***63.13***	3.11	203	55	***65.55***
512	4,75	412	56	***74.63***	5.11	211	60	***68.07***
1024	10.79	441	54	***80.02***	10.05	246	63	***79.52***
2048	19.98	476	72	***86.06***	20.16	263	72	***84.91***
	File 3				File 4			
	289				121			
128	2.47	164	68	***55.23***	2.53	74	21	***60.69***
256	2.92	178	71	***60.01***	3.12	82	29	***67.12***
512	4.88	183	76	***61.63***	5.05	91	37	***74.38***
1024	10.10	234	81	***79.51***	10.27	101	38	***82.63***
2048	20.41	246	87	***83.19***	19.85	104	42	***85.02***

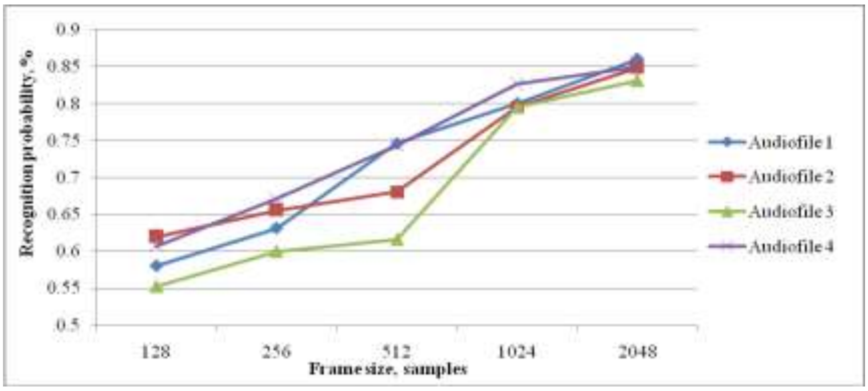

Fig. 6. Dependency of the signal recognition probability on the frame size

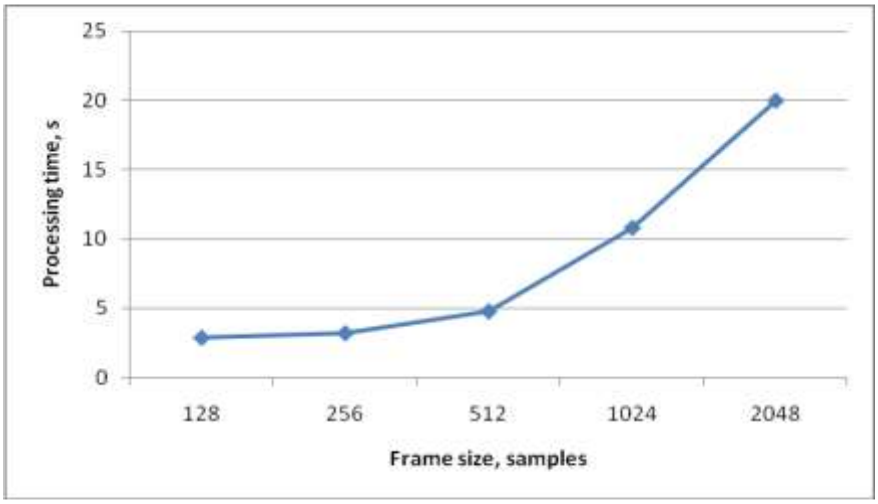

Fig. 7. Dependency of the signal processing time on the frame size

We have studied the efficacy of the recognition process for various frame sample rates. The frame sample rate is the amount of frames out of an audio-fragment to be analyzed per second. Since the length of each block is fixed, some part of data remains unprocessed and discarded. Consequently, the processing rate can be increased significantly by sacrificing the recognition quality. Each test audio-record is processed by the developed software module at frame sample rates ranging from 5 to 40 frames per second, with the size of FFT block fixed to 1024 samples. The experimental results are summarized in Table 2.

As a result of investigation we have calculated the probabilities of recognizing patterns in an audio-signal under scrutiny and consumed computational resources including processing time (Fig. 8 and Fig. 9). Probabilities of pattern recognition lies in a rather narrow range from 78.49% – 81.11% for the frame sample frequencies being analyzed. The difference in the bee colony state recognition probabilities for adjacent frequencies is only 0.41%, on average. The average processing time increases by 21.34% for adjacent frame sample frequencies. However, when switching the frequency from 5 frames per second to 10 frames per second, one might expect to increase the probability of successful pattern recognition only by 1.2%. Consequently, the optimal frame sample frequency is 10 frames per second, which is sufficient for organization of a recognition process with the successful recognition probability not worse than 79.69%.

6 CONCLUSIONS

The work considers the process of analyzing bee colony acoustic signals and its implementation as a software module for pattern recognition in bee sounds. The efficacy of the developed software module was studied over a set of test audio-files. We focused primarily on two metrics: the successful recognition probability and processing time. It was shown that the most suitable FFT frame size is 1024 samples whereas the most efficient frame sample rate is 10 frames per second. Having set the above-stated parameters to the recommended values, one might expect to successfully recognize acoustic sounds typical of a bee colony with the probability not worse than 79.69%.

The developed software module is intended to be a part of a complex automated bee colony monitoring system, as a component of an embedded system placed directly on a hive. Currently we are working on optimization of the DTW version and enhancement of the recognition process by noise suppression.

Table 2. The results of investigation into the efficacy of the software module at different frame sample rates.

Sample rate, frames per second	Processing time, s	Processing duration, s	Amount of properly recognized frames, N_α	Amount of false matches, N_β	*Probability of recognition, pS %*	Processing time, s	Processing duration, s	Amount of properly recognized frames, N_α	Amount of false matches, N_β	*Probability of recognition, pS %*
	File 1					**File 2**				
5	2.05	88	70	12	***77.95***	1.87	46	36	8	***77.19***
10	4.02	170	136	18	***78.8***	3.86	93	74	20	***78.24***
15	5.97	277	223	28	***79.26***	5.97	153	123	32	***78.97***
20	7.65	366	296	37	***79.64***	7.62	188	151	35	***79.15***
25	9.71	470	381	43	***79.92***	9.7	254	206	53	***79.69***
30	11.32	543	441	54	***80.02***	11.35	304	246	63	***79.52***
35	13.35	620	504	58	***80.19***	13.26	328	267	72	***80.03***
40	15.74	780	635	71	***80.23***	15.36	421	347	87	***80.97***
	File 3					**File 4**				
5	1.85	50	40	10	***78.67***	1.89	21	17	6	***80.15***
10	5.73	96	78	27	***79.45***	5.93	42	35	16	***82.27***
15	5.87	145	118	41	***79.56***	5.88	60	50	19	***82.49***
20	7.53	181	149	56	***80.45***	7.72	84	70	26	***82.47***
25	9.88	231	190	64	***80.54***	9.43	103	86	32	***82.64***
30	11.94	289	235	81	***79.51***	11.39	121	101	38	***82.63***
35	13.06	329	268	92	***79.71***	13.27	137	115	43	***83.12***
40	15.18	373	304	105	***79.75***	15.76	166	140	50	***83.5***

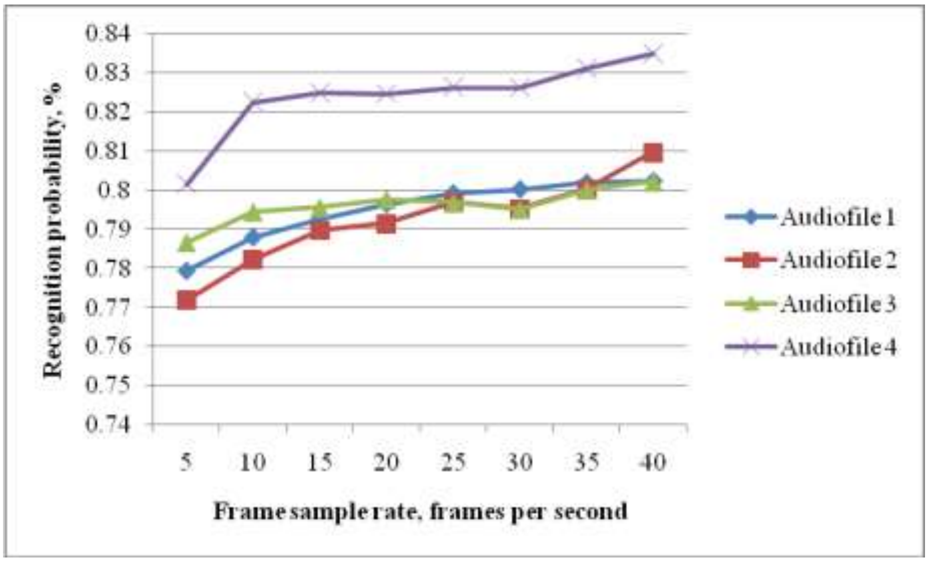

Fig. 8. Dependency of the signal recognition probability on the frame sample rate

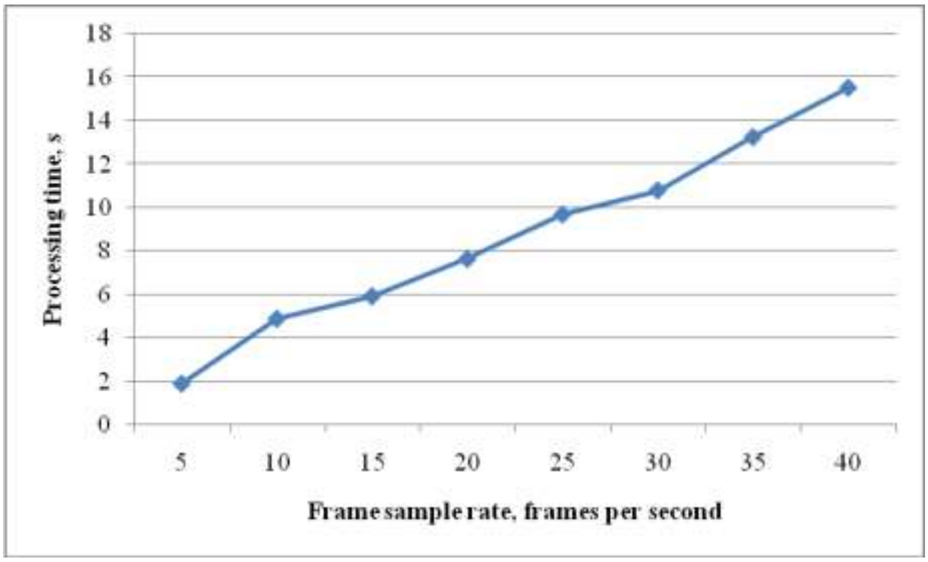

Fig. 9. Dependency of the signal processing time on the frame sample rate

REFERENCES

1. **Woods E.F. 1957**. Means for Detecting and Indicating the Activities of Bees and Conditions in Beehives. 2,806,082. U.S. Patent. 1957 Sep 10.
2. **Ferrari, S., Silva, M., Guarino, M., Berckmans, D. 2008**. Monitoring of swarming sounds in bee hives for early detection of the swarming period. Comput. Electron. Agric. 64(1), 72–77.
3. **Gil-Lebrero, S., Quiles-Latorre, F., Ortiz-López, M., Sánchez-Ruiz, V., Gámiz-López, V., Luna-Rodríguez, J. 2017**. Honey bee colonies remote monitoring system. Sensors 17(1), 55. doi:10.3390/s17010055.
4. **Kviesis, A., Zacepins, A., Durgun, M., Tekin, S. 2015**. Application of wireless sensor networks in precision apiculture. In Proceedings of the 14th International Scientific Conference Engineering for Rural Development (ERDev), pp. 440–445. Latvia University of Agriculture.
5. **Rybochkin A.F., Zakharov I.S. 2004**. Computer systems in beekeeping. Kursk State Technical University, Kursk (in Russian).
6. **Rybochkin A.F., Zakharov I.S. 2009**. System analysis of acoustic signals of a bee colony using code messages. Kursk State Technical University, Kursk (in Russian).
7. **Rybochkin A.F. 2004.** Monitoring and controlling the vital functions of bee colonies (in Russian)
8. **Qandour, A., Ahmad, I., Habibi, D., Leppard, M. 2014**. Remote beehive monitoring using acoustic signals. Acoustics Australia 42(3), 204–209. doi:10.1007/s40857-015-0016-5
9. **Cejrowski, T., Szymanski, J., Mora, H., Gil. D. 2018**. Detection of the bee queen presence using sound analysis. In 10th Asian Conference on Intelligent Information and Database Systems, pp. 297–306. Springer, Heidelberg. doi:10.1007/978-3-319-75420-8_28.
10. **Meikle, W., Holst, N. 2014**. Application of continuous monitoring of honeybee colonies. Apidologie 46(1), 10–22 https://doi.org/10.1007/s13592-014-0298-x.