VOICE COMMAND CONTROL FOR MOBILE ROBOTS

M.Hrnčár

Department of Control and Information Systems
Faculty of Electrical Engineering, University of Žilina

Abstract

Digital signal processing techniques and today's computing capabilities allow the computers to "understand" human speech. This paper describes the EllaVoice application, which is the user-dependant, isolated voice command recognition tool. It was created in MATLAB, based on dynamic programming and it could serve for the purpose of mobile robots control. The paper deals with the application of selected techniques like cross-words reference template creation or endpoints detection.

1 Introduction

The aim of this research is to describe the main aspects influencing the creation and function of voice-command system. For this purpose, I decided to create the voice command recognition tool EllaVoice which can be utilized also for the mobile robots control. The dynamic programming technique was chosen because of its wide use in isolated words recognition. Although this technique is quite old, there have been several variations or improvements of it proposed by Rabiner [5] and many others. Its advantage is in the easy way of training where the templates of the whole commands are used, which makes it very suitable in case of the automatic speech recognition systems with small dictionary (~ 10 commands). The main problem is in the preparation of reliable reference templates for the set of commands to be recognized [2].

The whole recognition process could be divided into two tasks. The first one is the isolated words endpoint detection. Its accuracy influences the second task – reference templates creation and the decision itself. The organization of this paper is as follows. In the next section we describe the problem of endpoint detection based on different features. Section three is devoted to the algorithms of recognizer. Section four describes the application in MATLAB and brings the overview of user interface. Finally, in the section five, we summarize the outcomes.

2 Endpoint detection of isolated words

The problem of locating the beginning and the end of speech utterance in an acoustic background of "silence" is important in many areas of speech processing. The good endpoint-locating algorithm can locate the region of the speech utterance to be recognized. The problem of locating the endpoints of the utterance is not trivial one expect in the case of acoustic environments with very high signal-to-noise ratio.

The main problem causes weak fricatives, especially at the beginning or at the end of the words. Another problem could be caused by the final nasals or trailing off of the certain voiced sounds. This comes from the model of vocal tract and the paper [1] illustrates more examples. It is important to realise that our goal is just to isolate enough of the word so that a reasonable analysis and recognition can be performed. Thus, it is not necessary to isolate the exact point where the word begins or ends, although the more accurate determination provides the better performance of the speech recognition system.

Based on these facts, we can summarize our requirements into these two points:

- reliable location of the acoustic events,
- effective processing capable to be applied to various background conditions.

The combination of zero-crossing rate (ZCR) and energy functions seems to be the most appropriate and effective tool for successful fulfillment of stated requirements. This combination provides us fairly accurate results. Both functions are applied at short-time speech segments, usually with the duration between 10 and 20 ms. In this case, the sampling rate was 22050 Hz and the 20 ms microsegment was chosen, so we have got 441 samples in one microsegment.

The zero-crossing function will be defined as

$$Z_n = \sum_{k=2}^{441} \left| \mathrm{sgn}[s(k)] - \mathrm{sgn}[s(k-1)] \right| , \tag{1}$$

where $s(k)$ is the discrete value of a speech signal in time (within the microsegment) and $sgn[s(k)] = 1$ for $s(k) \geq 0$, $sgn[s(k)] = -1$ for $s(k) < 0$ respectively. There exist several definitions of short-time speech energy. I have chosen the root mean square measurement described by Eq. (2) because of its advantageous attributes.

$$E_n = \sqrt{\frac{1}{441} \sum_{k=1}^{441} s(k)^2} \ . \tag{2}$$

The algorithm of endpoint detection must cover different conditions. We utilize the fact that during the first 100 ms of the recording there is no speech present. Thus, during this interval, the statistics of the background noise are measured – the average zero-crossing rate and average energy. Based on the average zero-crossing rate and a fixed threshold (of 51 zero-crossings during 20 ms microsegment) the algorithm determines the final value of zero-cross threshold *IZTC*. Two energy thresholds *ITL* (lower) and *ITU* (upper) are set from the measured noise energy and peak energy respectively.

The algorithm works as follows. It finds the points where the first and second thresholds were exceeded, based on the energy criteria. Now, it searches for the one, where the energy exceeded the *ITL* and then *ITU* threshold without falling down under the *ITL* again. If it is found, such point is labeled as the possible start (N_1). Analogically, we find the possible end (N_2) as the point where the function fells down below *ITL* after falling down below *ITU*, without exceeding *ITU* again. The endpoint locations we got are fairly conservative. Now we apply the zero-crossing criteria. This criterion is applied to the 240 ms (i.e. duration of 12 microsegments) region before N_1 and the algorithm counts the number of microsegments in which the zero-crossing rate was exceeded. If the number was two or more, the starting point is moved to $\hat{N}_1$, otherwise is kept at N_1 . This is illustrated in Fig. 1. The same procedure is done for N_2 but here were not the conditions for the position change fulfilled. On the *x*-axis, *n* represents the sequential index of the microsegment.

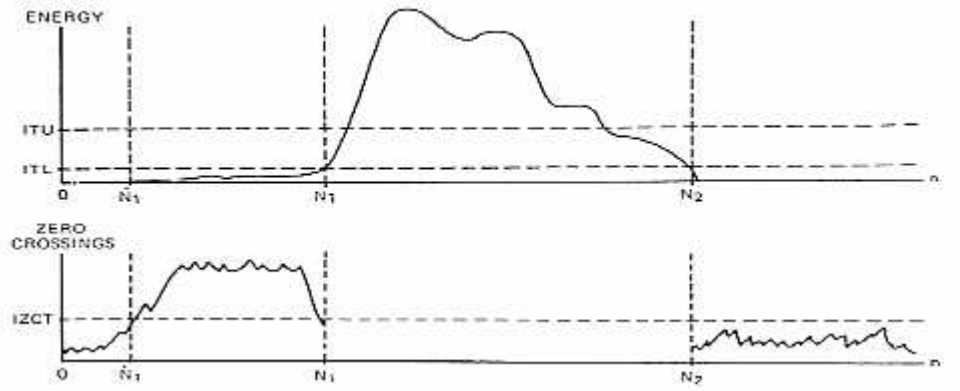

Figure 1: The illustration of endpoint detection process based on ZCR and short-time energy criteria

The algorithm has also to be able to detect and to deal with limit events, e.g. the truncation of the beginning or the end part of the word during the audio acquisition. If the conditions do not allow detecting the endpoints correctly, it results in the change of “check bit” value. All the features have been included into one m-file and can be called as a function. We need to send the audio file in wave form together with its sampling frequency to the input and we get the probable positions of the endpoints in ms (this values are empty if the “check bit” value has been changed, i.e. we are not able to determine the endpoint positions). Moreover, we get the estimated signal-to-noise ratio calculated from the background conditions before and during the presence of speech according to the Eq. (3)

$$SNR = 20.\log \frac{\max|s+n| - \max|n|}{\max|n|} \quad , \tag{3}$$

where the fraction denominator represents the maximal value of the environmental noise (obtained from the first 100 ms of the audio signal) and the nominator represents the maximal value of the speech signal (obtained from the interval where both speech and the background noise were present).

3 Improved Dynamic Time Warping algorithm

The technique of dynamic programming for the time registration of a reference and a test pattern has found widespread use in the area of isolated word recognition. This problem is important because the time scales of a test and a reference pattern are generally not perfectly aligned. In most cases, a nonlinear time warping is required to compensate for local compression or expansion of the time scale. For such cases, the class of algorithms known as Dynamic Time Warping (DTW) methods has been developed. These algorithms all assume that the input is a feature vector from an isolated word whose endpoints are, at least approximately, known [5]. Hence, the problem of DTW can be formulated as a path finding problem over a finite grid as shown in Fig. 2. We denote the reference pattern as a sequence of frames, $R(m)$, $m = 1,2, ..., M$, where $R(m)$ is, in general, a multidimensional feature vector that describes the characteristics of the m-th frame of the spoken word. Typically, a frame (microsegment) of data encompasses from $10 \div 50$ ms of data and Mel Frequency Cepstral Coefficients (MFCC) and their derivates are used for its description. We denote the test pattern as a sequence of frames, $n = 1, 2, ..., N$, where $T(n)$ is also a multidimensional feature vector. The optimal path

$$m = w(n) \tag{4}$$

is minimizing the distance between the "images" of reference and test words. As you can see, there is a global region of the DTW function movement. It is closely described in [3].

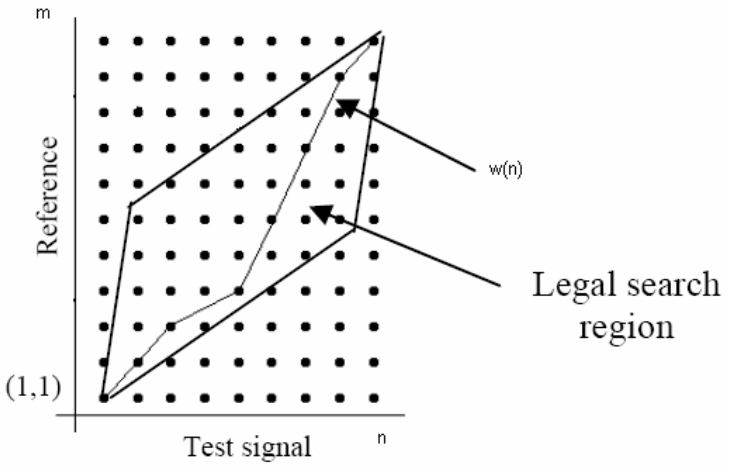

Figure 2: DTW function and the global path region

The distance function is used to obtain the optimal warping path. A general form for such a function is

$$D(A,B) = \min_{\{i(k),j(k),K\}} \left[\frac{\sum_{k=1}^{K} d[i(k), j(k)] \hat{W}(k)}{N(\hat{W})} \right] \quad , \tag{5}$$

where $d[i(k), j(k)]$ is the local distance between $n = i(k)$ microsegment of the tested signal image and $m = j(k)$ microsegment of the reference signal image, $\hat{W}(k)$ is the weighting function and $N(\hat{W})$ is a normalization factor. The DTW algorithm can be divided into three steps:

- the initialization, for k = 1,
- the recursion, representing the computation kernel and introducing the function of party accumulated distance,
- the final distance calculation.

The traditional way of preparing the reference templates is by judiciously selecting one example for each word (needed to be recognized) and considering it as a reference template for that word. The disadvantage of using a single reference template is that it is not robust to the speech signal variability. That is because it is almost impossible for a person to repetitively speak a word exactly in the same way. To overcome this problem without incurring more computations in the recognition phase, a technique is developed to prepare more robust templates, called crosswords reference templates (CWRTs). Using these templates has improved the recognition accuracy, as it is prepared from multiple examples rather than just one example.

A few examples ($3 \div 5$ examples is normally sufficient) for each word have to be prepared beforehand. Then, the average length of the extracted templates is calculated. Next, the template with the length nearest to the average length is chosen to be the best template. This later template is considered as the initial reference template. Then the other templates are time aligned by the DTW process such that their lengths will be equal to the chosen initial template. Finally, the final reference template will be created by averaging the time-aligned templates across each frame. This technique is described in more detail in [2]. Fig. 3 illustrates the CWRT preparation as derived from three examples.

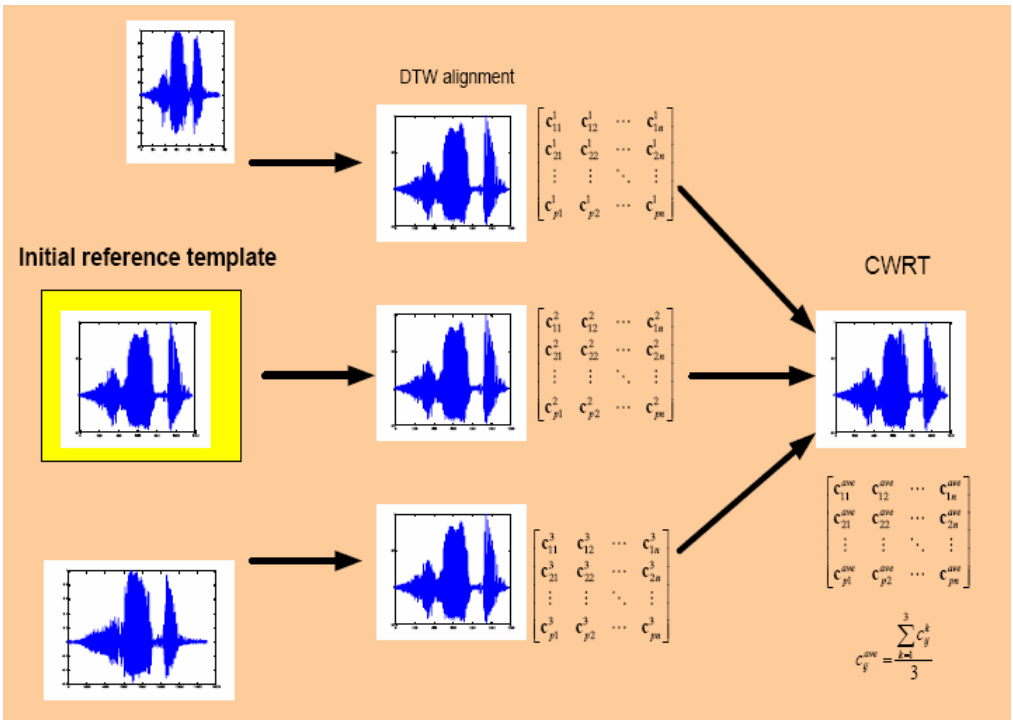

Figure 3: Scheme of CWRT preparation from three utterances

One disadvantage of the DTW is the fact that it always try to assign the spoken word to one from the dictionary (with the minimal distance). This assignment is also done in the case if the spoken word (command) is not included in the dictionary, so it negatively influences the safety of such a system. Therefore I have tried to implement some kind of safety-related function. Based on the observations at given sampling frequency and microphone used for the recording of utterances, I was able to set two rules. The first one is the maximal overall "distance", which can be accepted for given set of voice-commands. If this distance is exceeded, the spoken command will not be assigned to any of the commands in the dictionary. If this condition is fulfilled, the system compares the distances to two most probable words from the dictionary and the decision is made only in case that the ratio between these distances is at least 1,33, i.e. the recognized word is not too similar to the other one in

the dictionary. The implementation of all the algorithms stated above to EllaVoice application is described in the next section.

4 The description of EllaVoice application

The Automatic Speech Recognition (ASR) on a computer system is trying to simulate the activity of Human Auditory System (HAS) and brain. The classificator can work either in the training or classification mode [7].

In the training mode, every speaker (in case of user dependant system) pronounces every word (command) one or more times. The human voice, as analog acoustic signal, is acquired by the microphone and digitalized by A/D converter at given sampling rate. The digital signal is effectively transformed with the help of chosen mathematical techniques (e.g. FFT or MFCC) to the vectors of numerical values called patterns. The classification principle is illustrated in Fig. 4. We usually require such a system to work real-time, i.e. the decision should be made within 0,5 s [3].

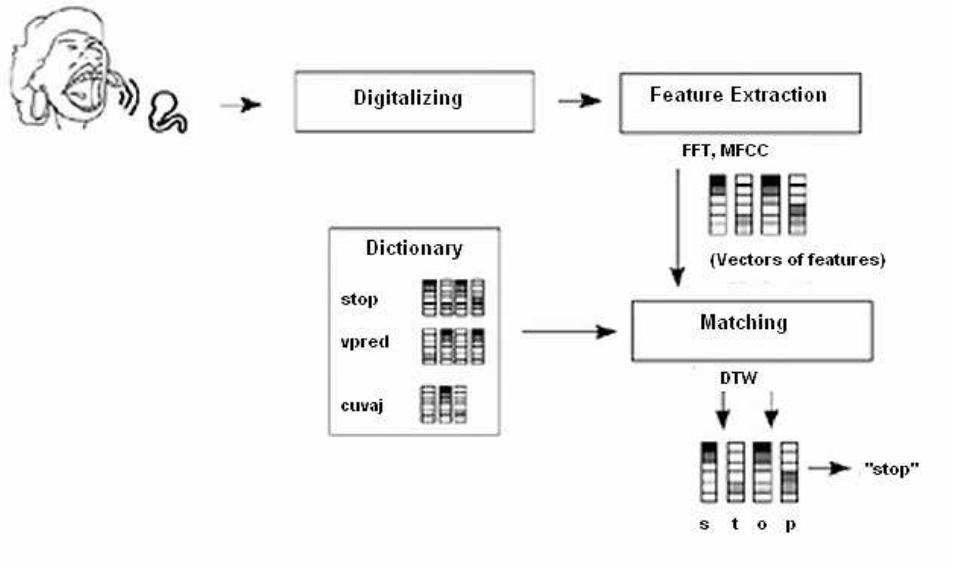

Figure 4: Scheme of Automatic Speech Recognition system

EllaVoice application also works on the same principle. Thus, it is divided into two modules. The first one is a training module. This module has its own Graphical User Interface (GUI) named EllaVTrain which is illustrated in Fig. 5 together with MATLAB Command Window in the background. It allows user-friendly recording and manipulation with wav-files as well as the creation of Cross Words Reference Templates. Let's briefly describe the interface.

The user interface includes several buttons and the graphical window which displays the recorded signal together with its time scale. The signal interval between the determined endpoints is highlighted in red. The user can set the path and the filename of CWRT as well as the number of samples from which the reference template is going to be created. The "REC NEXT" button serves for the push-to-talk recording of the command. The "PLAY" button allows the user to check the endpoint detection also by audio perception – the red highlighted interval of the signal is played. If the user is not satisfied with the current sample, he can record the new sample by pressing the "REC NEXT" button again. Otherwise, he confirms the sample by pressing the "ACCEPT" button. The number in right bottom corner is increased by one. If this number reaches the required value for CWRT creation, the CWRT can by simply created by pressing the activated "CREATE REF. SAMPLE AND SAVE" button under the graphical window (in Fig. 5 inactive). The CWRT is created and its values are saved to the text file with given filename. The "RESET" button serves for reset of all data, except of file path. This is used when we want to start creating CWRT of another command. The auxiliary window Figure 123 represents the closer analyze of recorded signal. The ITL and ITU threshold values are displayed to make the coefficients change easy in case of another microphone use.

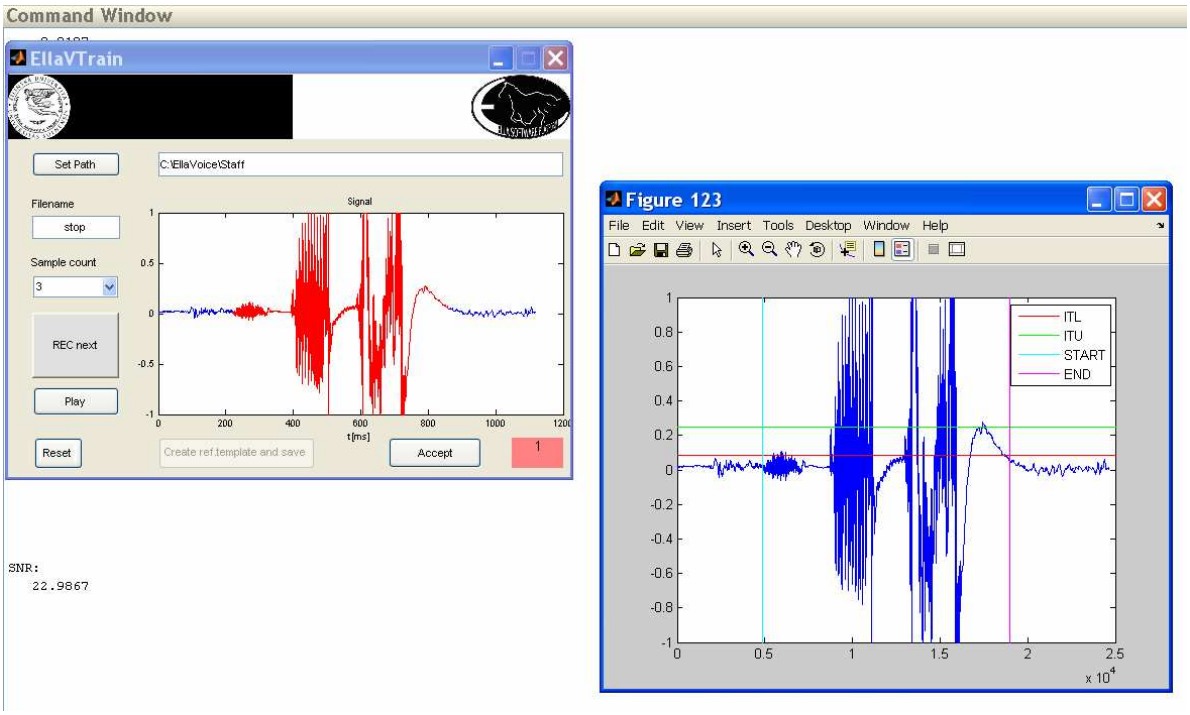

Figure 5: The training GUI (left) with the signal analysis window (right)

The second module represents the core of the program. You have to set the active user whose commands have previously been defined and to choose whether to activate the safety-related function, first. Then you can, after pressing return key, say you voice command to be recognized to the microphone. The program calculates signal-to-noise ratio and evaluates your command according to "standard" distance evaluation used in DTW method. In case of active safety-related function, the evaluation is more complex.

Finally, we had to deal with the problem how to integrate the output from EllaVoice application in the MATLAB environment to our Ella SW platform. This platform originated as we had problems to find software package capable of meeting all our current and future requirements for the complex environment for the simulation of mobile robot control. It was written in C++ GLSL and Lua programming languages and it is based on OpenGL SDL and Boost libraries [6]. It is compatible with Linux and Windows operating systems. Application consists of multiple different interface modules providing the interaction with user. The virtual reality model of our six-leg walking robot in Ella SW platform and the robot prototype are illustrated in Fig. 6.

Figure 6: Six-leg walking robot prototype and its virtual model

As far as an effort to create the dynamic library from the existing code in MATLAB for following implementation to the Ella SW platform has seemed to be too complicated, we have chosen different attitude. To every word in the dictionary a specific symbol is associated. When the

recognizer comes to the decision it writes the specific symbol of the chosen (i.e. most probable) word from the dictionary to the text file. This file is shared in the network and its modification is checked 10 times per second from the far-end computer on which Ella SW platform is running. The command is recognized based on the specific symbol and is sent to the control system of either virtual model or directly to the prototype of the robot. This simple way allows multiplatform running of our voice command recognition application. It allows the remote control of the robot under the condition of real-time operation as well. Nowadays, the system is being tested and its successful detection rate is evaluated.

5 Conclusion

EllaVoice application, which is the user-dependant, isolated voice command recognition tool, was created for the purpose of mobile robot control and described in this article. The effectiveness of the described algorithms for the small set of voice commands (~ 10) is now tested at the group of 25 people. It seems that for the good recognition rate achievement, it is needed to use the same microphone and computer every time. The more complex results will be published in my dissertation work.

Of course, there still will be a space for possible improvements. The adaptive filter addition can improve the performance of the system by the background noise suppression. For the purpose of endpoint detection, there exist also other methods like e.g. energy-entropy method which has its own advantages. Finally, the DTW method is nowadays in the shadow of Hidden Markov Models methods providing better performances. However, the aim of my research is not to create a perfect voice recognition system but to describe the issues of voice-command system creation based on the selected methods. These have been previously chosen in regard to our requirements.

Acknowledgement

This paper is supported by cultural and educational grant agency within project KEGA № K-057-06-00: Innovation of methodology of laboratory education, based on modeling and simulation in MATLAB, with educational models through e-learning.

References

[1] L. R. Rabiner, M. R. Sambur. *An Algorithm for Determining the Endpoints of Isolated Utterances*. The Bell System Technical Journal, USA, 1975

[2] H. A. Waleed, D. Chow, G. Sin. *Cross-words Referece Template for DTW-based Speech Recognition Systems*. TENCON Conference, Volume 4, 2003

[3] J. Psutka. *Komunikace s počítačem mluvenou řečí*, Academia Praha 1995. ISBN 80-200-0203-0

[4] X. Huang, A. Acero, H. Hon. *Spoken Language processing*. Prentice Hall PTR, New Jersey 2001. ISBN 0-13-022616-5

[5] C. Myers, L.R. Rabiner, A. E. Rosenberg. *Performance Tradeoffs in Dynamic Time Warping for Isolated Word Recognition*. IEEE Transactions on Acoustic, Speech and Signal Processing, Volume 28, 1980

[6] T. Michulek. *Using virtual reality to develop six legged walking robot control system*. Applied computer science, Volume 3, 2007

[7] M. Hrnčár. *Aspects of voice-command creation in Human-Machine Interface.* Dissertation project, Žilina 2007

M.Sc. Martin Hrnčár
University of Žilina, Faculty of Electrical Engineering, Univerzitná 1, 010 26 Žilina, Slovakia
e-mail: martin.hrncar@fel.uniza.sk