Voice Signal's Noise Reduction Using Adaptive/Reconfigurable Filters for the Command of an Industrial Robot

Moisa Claudia**, Silaghi Helga Maria**, Rohde L. Ulrich * '*** ,
Silaghi Paul****, Silaghi Andrei**** and Silaghi A. Marius*

*Department of Electrical Engineering,
University of Oradea, Faculty of Electrical Engineering and Information Technology,
1, University, 410087 Oradea , Romania, E-mail: masilaghi@uoradea.ro ,

**Department of Electrical Drives and Automation,
University of Oradea, Faculty of Electrical Engineering and Information Technology,
1, University, 410087 Oradea , Romania, E-mail: : hsilaghi@uoradea.ro , moisa.claudia@live.com

*** Department of Communications, Technical University of Cottbus, GERMANY
Email: rohdeu@tu-cottbus.de

**** "Politehnica" University of Timisoara
2, Vasile Parvan Bv, Timisoara , Romania

Abstract - This paper presents a voice signal's noise reduction using adaptive / reconfigurable filters for ordering an industrial robot using isolated words. Finally, some results of the simulation are presented.

Key-Words: Voice processing, Signal filtering, Industrial robot, Speech recognition, Command, Noise reduction

1 Introduction

Nowadays, vocal command is becoming more and more fashionable for industrial automation. Rather than implementing a control panel which may not be user friendly and consequently difficult to use, a speech recognition system is very attractive for basic commands such as: START, STOP, PAUSE, etc.

In [3] we have proposed an application for speech recognition, designed to recognize isolated words. In laboratory environment, we have proved that the recognition rate is satisfactory. The performance is however worse compared to modern Hidden Markov Model (HMM)-based speech recognition software [4]. Important is the fact that our application is computationally efficient and thus would need less resources for field deployment.

We have tested our application in the field. Compared to laboratory environment, speech signal on site was altered by noise coming from various sources such as: motors, machine tools and operators. In order to keep our application in the real-environment scenario, we need to add a noise-filtering stage before the actual speech recognition takes place.

In the process of transmitting a signal from source to receiver, i.e. from operator to the machine in-built microphone, the signal is distorted by the perturbation of multiple factor. On one hand we are talking about the effect of the transmission channel, in our case the air which transmits the acoustic signal. On the other hand, the signal is affected by additive noise in the channel.

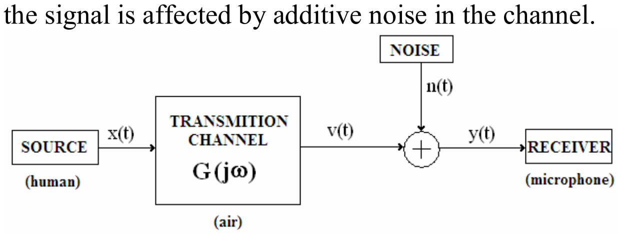

Fig. 1. The block diagram of a Transmission System

A block diagram of the acoustic signal path from source to receiver is shown in figure 1. Suppose a source emits a signal x (t), which is carrying the information. The transmission channel can be modeled as a filter defined by G(jω). The effect of the signal transmission channel is then the convolution of the acoustic signal and the channel impulse response, as defined by the following equation:

$$v(t)=x(t) * g(t) \quad (1)$$

Additive noise in the channel (motors, machines tools, operators – all which can be found in an industrial environment) is modeled as a signal n(t). The noise is additive, thus the received signal is described by the equation:

$$y(t)= x(t) * g(t) + n(t) \quad (2)$$

The receiver will need to extract the useful information carried in the received signal. This implies a filtering operation in order to eliminate the effects of the channel and additive noise in the channel. So the solution would involve designing an optimal adaptive filter, which, if applied the received signal y(t) would produce a signal as close as possible to the original signal x (t).

2 Wiener filter

To solve the filtering problem, we have chosen the Wiener filter. It works on the signal spectrum separating the useful signal components from the noise. It starts from the premises that some spectral components contain much useful information, while others contain more noise. Therefore, noisy spectral components are those that will be blocked.

The Wiener filtering foundations are formulated on minimum quadratic error (Least square error) linear adaptive filtering. Wiener filter can be of the infinite pulse response (IIR) or finite impulse response (FIR). In either implementation, each sub-band gain is determined by the ratio of useful information and noise content in the spectral components.

Determination of the Wiener filter equations

Further on, we describe the Wiener Filter considering a FIR filter model. The filter input-output relationship is given by:

$$\hat{x}(m)=\sum_{k=0}^{p-1} w_k y(m-k)=w^T y \quad (3)$$

where w is the vector of Wiener Filter Coefficients, y(n) is the noisy signal input to the Wiener Filter and x^(m) is the filter output signal. In the frequency domain, equation (3) translates to:

$$\hat{X}(f)=Y(f)\cdot W(f) \quad (4)$$

The estimation error signal, E (f), is defined as the difference between the desired signal, X (f), and the signal obtained after filtration is:

$$E(f)=X(f)-\hat{X}(f)=X(f)-Y(f)\cdot W(f) \quad (5)$$

The target is to minimize the error signal E(f). This leads to the determination of the Wiener filter coefficients as:

$$W(f)=\frac{P_{xy}(f)}{P_{yy}(f)} \quad (6)$$

Where $P_{yy}(f)=E[Y(f)Y^*(f)]$ and $P_{xy}(f)=E[X(f)Y^*(f)]$ represent the power spectrum of Y (f) and cross-power spectrum of Y (f) and X (f) respectively.

3 The filtering function

The wiener filtering function has been implemented in Matlab. The inputs to the function are the audio signal to be filtered, the sampling frequency and length of the signal initial silence. It is considered that any audio signal is started by some samples of silence where only the noise is present. Upon this assumption, the noise spectrum is extracted to be later used in the actual filtering. Then, effective signal filtering is based on Overlap-Add method [5].

Primarily, the function performs a pre-accentuation of the audio signal. After this, the signal is divided into overlapping frames. The frame sharing is based on a number of samples which can be chosen arbitrarily.

The following operation is the Fourier transform of each frame and the power spectrum. The initial period of silence is calculated as an average from noise power spectrum. Based on these values we obtain effective filtration. On each filtered frame of the audio signal, the following operations are performed.

1. Testing that there is an actual speech signal (voice activity). Speech Detector - Voice Activity Detector – [3] works on the principle of spectral distance between the current frame spectrum and noise spectrum calculated from the initial silent period (spectral magnitude distance). The algorithm detects the input parameters of the speech signal which are to be examined, the amplitude of the noise spectrum, the number of noise frames preceding the current frame of the last voice activity detection and the noise margin. Based on these parameters, the function decides if the current frame contains speech or just noise and increments or resets the number of previous noise frames.

2. After obtaining new voice signal frames, the Overlap-Add method is applied in order to obtain the filtered audio signal. It should be noticed that the operations on the frames are made in the frequency domain. The reconstruction of the signal will be made in the frequency domain.

3. After the reconstructed signal was obtained in frequency spectrum, the inverse Fourier transform is applied to obtain the time domain signal. Finally, the signal obtained, represents the filtered version of the original audio signal.

4 Experimental results

The application was implemented in Matlab and was tested in a real industrial environment. The test signal was recorded in a noisy industrial environment for the vocal command: "START". The implemented filter is first subject to a set of simple tests in an environment with fewer noise sources. Then, its performance is tested in laboratory environment with artificially generated noise. Finally, we deploy the Wiener filter on site an test its performance in real circumstances.

4.1 Vocal command with ambient noise (in time and frequency)

The signal was then filtered with the implemented Wiener filter. The signal representation if time-domain and frequency-domain, for both the original and filtered signals, is shown in figures 2-5.

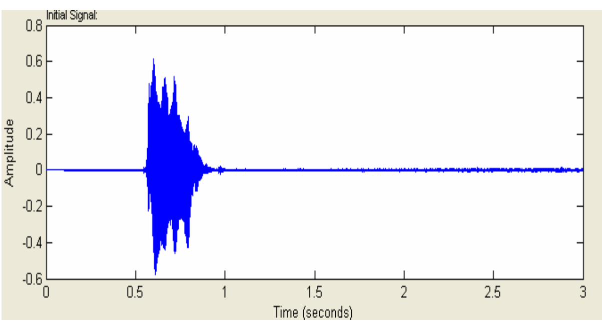

Fig. 2. Voice signal with ambient noise in time

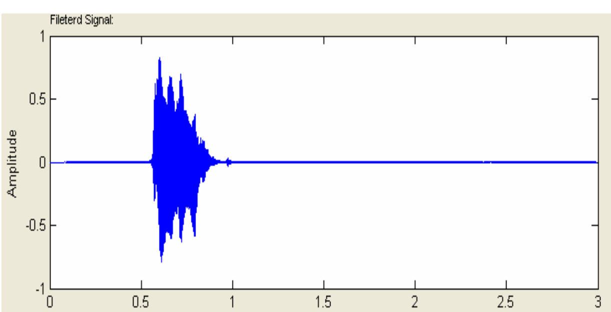

Fig. 3. Voice signal with ambient noise filtered in time

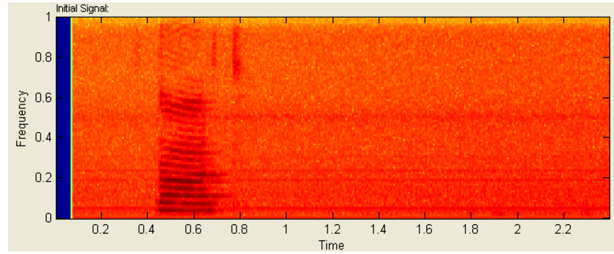

Fig. 4. Voice signal with ambient noise in frequency

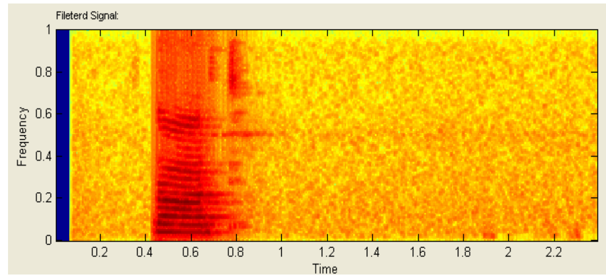

Fig. 5. Voice signal with ambient noise filtered in frequency

4.2 VOCAL COMMAND WITH ON SITE WITH HUMAN VOICES NOISE

Next, the signal was recorded in the presence of other operators. Thus, the noisy environment consists of human voices. The signal representation if time-domain and frequency-domain, for both the original and filtered signals, is shown in figures 6-9. Figures 7 and 9 prove that the Wiener filter achieves the expected results in the given scenario.

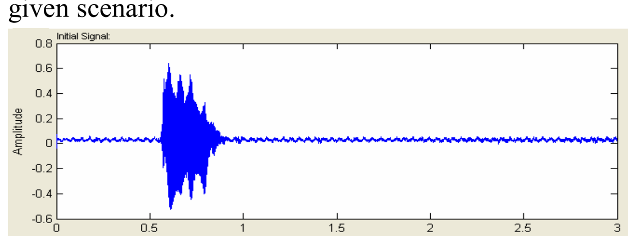

Fig. 6. Voice signal with on site with human voices noise in time

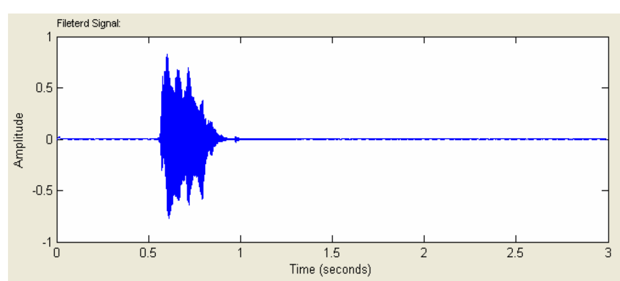

Fig. 7. Voice signal with on site with human voices noise filtered in time

ISBN: 978-1-61804-027-5

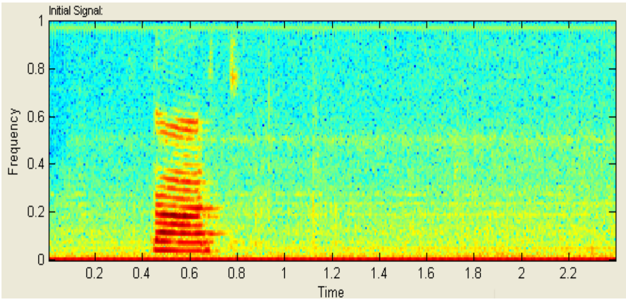

Fig. 8. Voice signal with on site noise with human voices in frequency

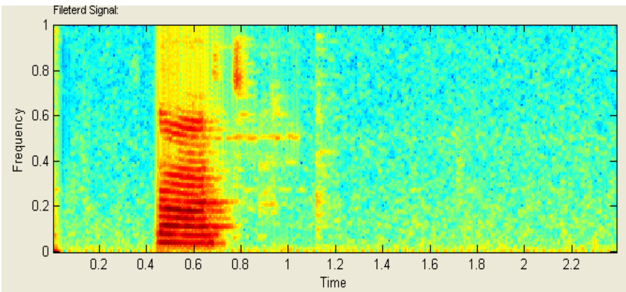

Fig. 9. Voice signal with on site noise with human voices filtered in frequency

Next, we want to test the actual performance of the Wiener filter. Therefore we simulate noise and add it over the speech signal.

4.3 VOICE COMMAND WITH GAUSSIAN SIMULATED NOISE

The signal was artificially altered with generated Gaussian noise, with a noise amplitude value of 6% from the signal peaks. The time-domain representations and the corresponding spectrograms prove the efficiency of the Wiener filter.

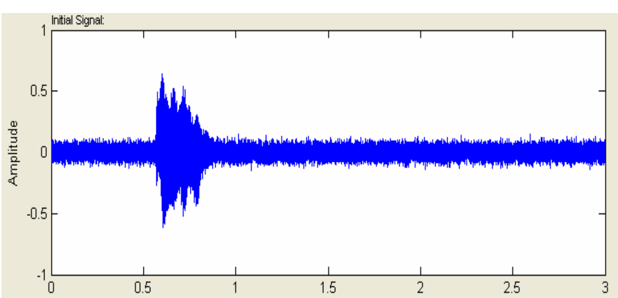

Fig. 10. Voice signal with Gaussian simulated noise in time

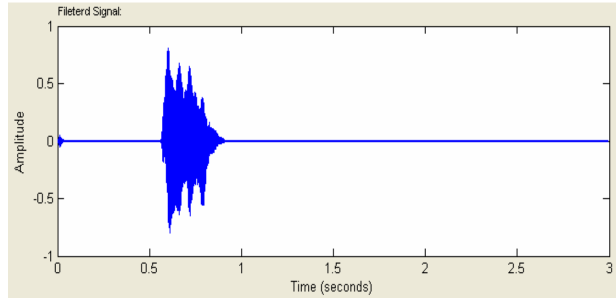

Fig. 11. Voice signal with Gaussian simulated noise filtered in time

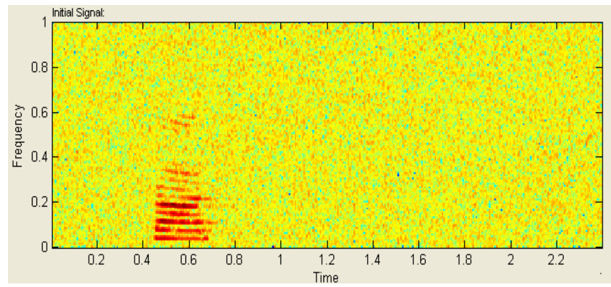

Fig. 12. Voice signal with Gaussian simulated noise in frequency

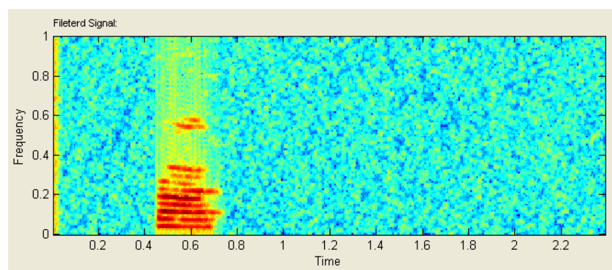

Fig. 13. Voice signal with Gaussian simulated noise filtered in frequency

In a next attempt, noise amplitude was increased to 8% of the signal peaks. Again, the filter performs well and cleans the signal.

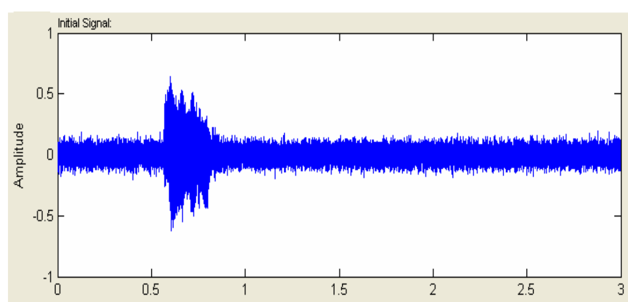

Fig. 14. Voice signal with high Gaussian simulated noise in time

ISBN: 978-1-61804-027-5

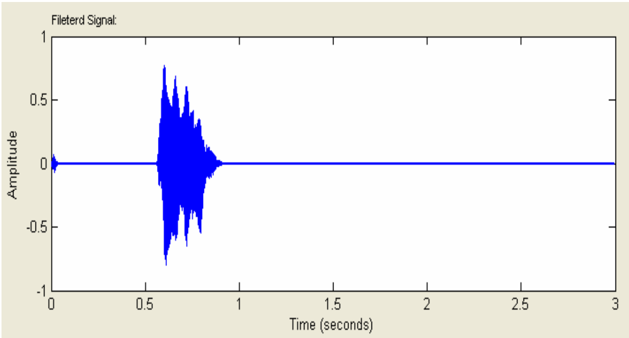

Fig. 15. Voice signal with high Gaussian simulated noise filtered in time

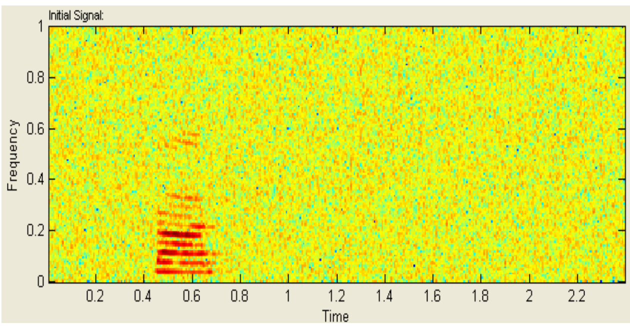

Fig. 16. Voice signal with high Gaussian simulated noise in frequency

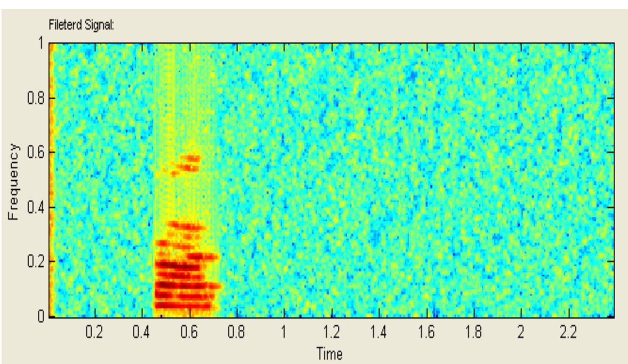

Fig. 17. Voice signal with high Gaussian simulated noise filtered in frequency

4.4 VOCAL COMMAND WITH ON SITE WITH HUMAN VOICES NOISE

Finally, we take the Wiener filter on site. In the presence of human speech noise, the filter provides good results, as shown in figures 18-21.

In the presence of noise generated from machines, the signal isn't completely cleared from the unwanted noisy components, see figures 23 and 25.

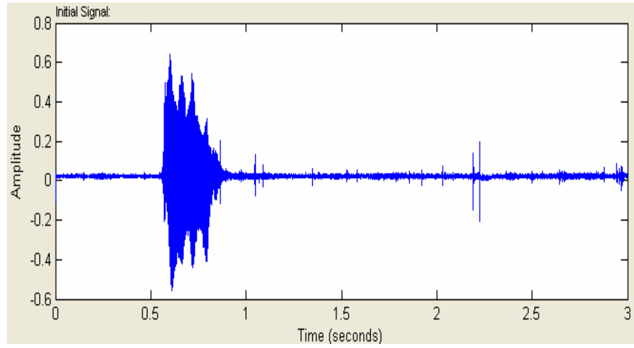

Fig. 18. Voice signal with on site with machine tools' noise in time

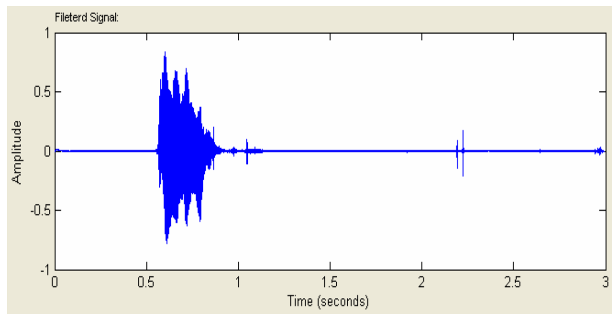

Fig. 19. Voice signal with on site with machine tools' noise filtered in time

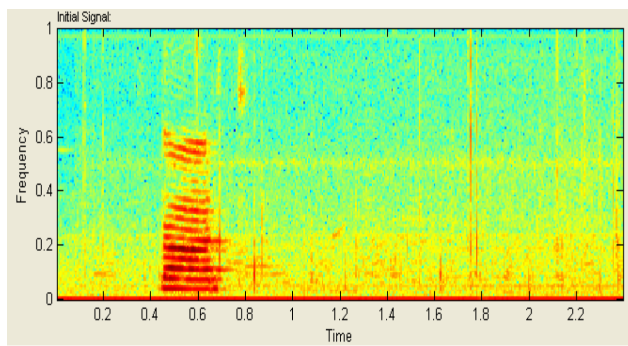

Fig. 20. Voice signal with on site with machine tools' noise in frequency

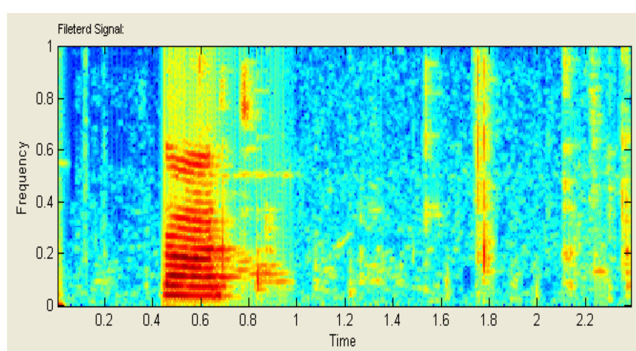

Fig. 21. Voice signal with on site with machine tools' noise filtered in frequency

However, it is obvious that the signal spectrum is well enough recovered so that it is usable for further processing.

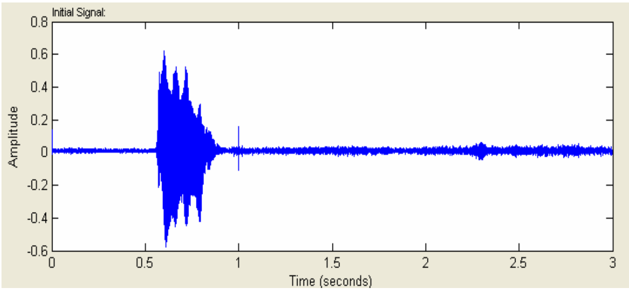

Fig. 22. Voice signal with on site noise in time

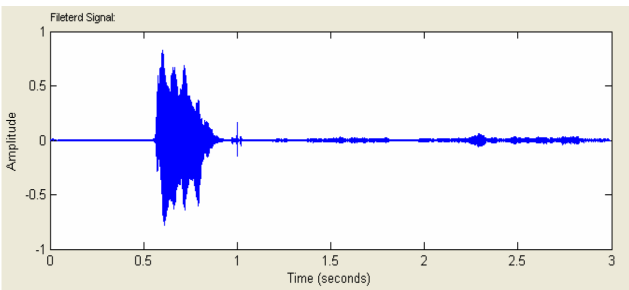

Fig. 23. Voice signal with on site noise filtered in time

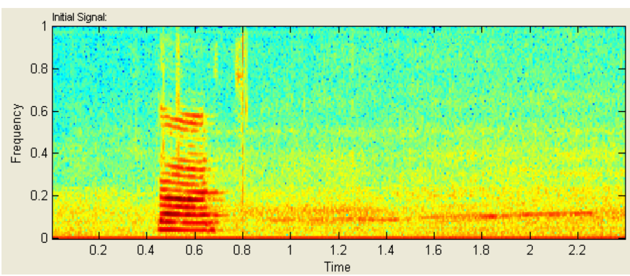

Fig. 24. Voice signal with on site noise in frequency

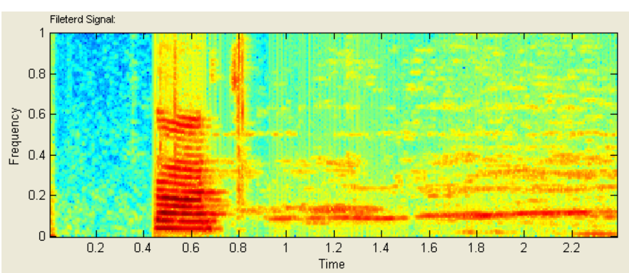

Fig. 25. Voice signal with on site noise filtered in frequency

5 Conclusion

Based on the experiments and the results, some conclusions can be made on Wiener filtering.
The time-domain representations and the corresponding spectrograms prove the efficiency of the Wiener filter.
First, it can be concluded that the Wiener filtering method is applicable to audio signals and the results are satisfactory even in very noisy environments (like industry). A second conclusion refers to the maximum amplitude of the noise that can be attenuated without compromising the sound. Regarding the noise from the industrial environment, there will be very small portions of noise, which will be interpreted as voice-activity, but they will not affect the command given to the robot. We need a compromise between performance and processing speed, which can be set accordingly to the noisy environment where the robot will be used.

References:

[1] Steven W. Smith, Ph.D. - *The Scientist and Engineer's Guide to Digital Signal Processing,* http://www.dspguide.com/

[2] Saeed V. Vaseghi - *Multimedia Signal Processing Theory and Applications in Speech, Music and Communications,* John Wiley& Sons, Ltd, 2007

[3] C. Moisa, H. Silaghi, A. Silaghi - *Speech and Speaker Recognition for the Command of an Industrial Robot*, WSEAS Timisoara, 2010

[4] M. Giurgiu,G. Toderean, B Szekely - *An approach to Isolated Word Recognition with Markov Models,* Bucharest, 1995

[5] Steven W. Smith – *FFT convolution and the overlapp-add method*, 2007

[6] H. Silaghi and V. Spoiala *Electrical Drives Design*, University of Oradea Publishing House, 2009

[7] V. Spoială, Helga Silaghi, D. Spoială, *Automatic Control System Based on Field Orientation Control for a Two-Phase Asynchronous Motor*, Journal of Computer Science and Control Systems, 2009, University of Oradea, Vol.2, Nr.2, ISSN 1844-6043