Application of Software Defined Radio for Noise Reduction Using Empirical Mode Decomposition

Sapan H. Mankad[1] and S.N. Pradhan[2]

[1] Institute of Technology, Nirma University, Ahmedabad
sapanmankad@nirmauni.ac.in
[2] Institute of Technology, Nirma University, Ahmedabad
snpradhan@nirmauni.ac.in

Abstract. Software Defined Radios (SDR) are aimed at reducing the efforts required specifically in Wireless Communication. Many hardware devices are currently being used for communicating via radio waves. The function of SDR is to replace possibly all the hardware stuff by software which results in great flexibility and portability. This concept has opened new windows to the world of digital communication. Today there exists many flavors of SDR. This paper focuses on the open source GNU Radio and its capabilities. The GNU Radio Project serves as a reference for experiments in the area of signal processing and communications. This paper deals with utilizing the capabilities of software radios to improve the quality of the incoming signal. Our objective was to improve the received signal by reducing noise and thus enhancing the overall communication quality. We propose the use of Empirical Mode Decomposition (EMD) method embedded into GNU Radio. The idea presented here is to include the EMD functionality in GNU Radio toolkit so as to ensure reduction of error for better communication. We have integrated the capabilities of Empirical Mode Decomposition into GNU Radio and found improvements in the simulated environment.

1 Introduction

Mobile communications have expanded the horizons of signal processing in the modern era of communication technologies. Several aspects of digital and analog signal processing affect the performance of communication. To achieve benefits, the minimum expected performance has to be met by concerned organization or telecommunication companies or their signal processing logic. Nowadays, when the planet is connected all over, *mobility* has also become common to all. The issues related to mobility can weaken the performance of the signals being transmitted. In addition, in the technologically rapidly emerging world, new applications arrive at quick rate which everyone wants to avail. *Reconfigurability* is another factor associated in this concern. The target is that the incoming signal should reach with minimum possible attenuation to the destination. Mechanisms to smoothen the received signal have to be *portable* or ready to use. This can be beneficial not only to improve the performance but also to achieve better signal reception quality.

D.C. Wyld et al. (Eds.): Advances in Computer Science, Eng. & Appl., AISC 166, pp. 113–121.
springerlink.com

In general terms, signal processing is done through hardware based radio consisting of the components as illustrated in Figure 1. The use of hardware circuitry limits the researchers to make any dynamic change frequently. This difficulty led to search some alternative way to accomplish the same task more fairly. This is how SDR came into existence. It brings the capabilities of radio functionalities with signal processing functionalities to achieve reconfigurability and portability.

Software Defined Radio - this term was coined by Joseph Mitola[13] in 1991. As described in [1], "a basic SDR system may consist of a personal computer equipped with a sound card, or other analog-to-digital converter, preceded by some form of RF front end." Most of the signal processing tasks are handed over to the general-purpose processor instead of utilizing special-purpose hardware, thereby producing a transceiver that can receive and transmit different radio protocols or waveforms based solely on the software used. SDR fecilitates as a single wireless device which supports a wide range of functionalities.

GNU Radio Project[12], founded by Eric Blossom is an open source software radio community that makes it possible to add reconfigurability to existing signal processing packages. GNU Radio can be considered as a signal processing toolbox which can be customized as per the need. Signal Processing blocks are written in C++ and mapped into Python using simplified wrapper and interface generator (SWIG). GNU Radio Companion (GRC) is a GUI which can be used for convenience in programming.

A signal, when transmitted over a channel, becomes corrupted and reaches its destination with some alteration. Unless the communication is critical, this alteration may be acceptable. In order to ensure correct delivery of data, mechanisms to reduce the noise have to be devised. Use of filters is one such method, but it requires techniques to approximate noise pattern in the signal. Huang et al.[6] proposed Empirical Mode Decomposition that decomposes a signal into different monotonic signals, commonly

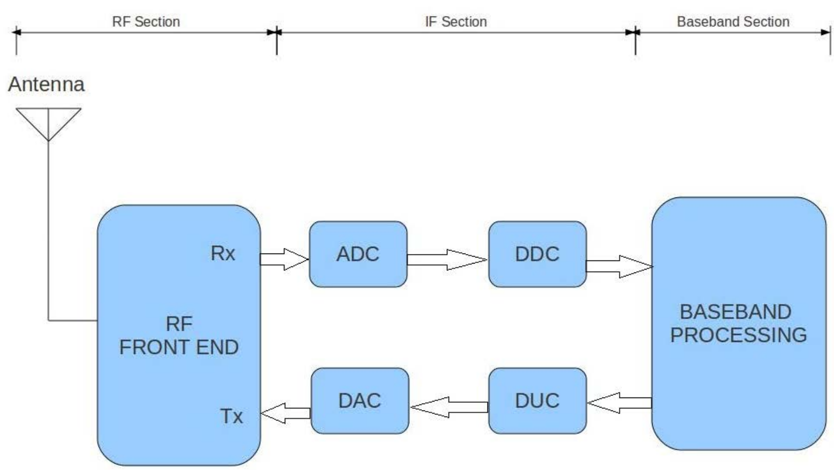

Fig. 1. Block Diagram for Hardware Defined Radio

known as intrinsic mode functions. This method is extremely helpful to separate a signal into several mono-component signals, which may be processed later on.

2 Motivation

Many researchers have thoroughly gone through the use of GNU Radio (an open source SDR) and contributed their findings in the area of wireless protocol testing or implementing various receivers. But a little research has been done in the area of writing a custom signal processing application for this tool. The challenge was to find a suitable way for noise reduction from an incoming signal, thus reducing the bit error rate (BER). It was found that till date, there haven't been any inbuilt signal processing block in GNU Radio to support this task except usual filtering blocks (which in turn, requires input parameters such as frequency or the pattern structure of the noise, which may not always be possible to estimate). Especially in wireless domain, one cannot predict the exact noise pattern in advance. Our work was centered around exploring or establishing some method that could help solve this problem. Empirical Mode Decomposition is useful to decompose the input signal into monotonic signals, which may later be useful to identify the noisy components in the signal.

Our target was to make it possible to utilize the capabilities of SDR to reduce the hardware based tasks and maximize software based computations so as to achieve reconfigurability. SDR, thus in real terms, enables *R*econfigurable *A*daptive *D*ynamic *I*nput *O*utput.

3 Empirical Mode Decomposition

Mathematical formalization of Empirical Mode Decomposition, as mentioned in [10], is described in Section 3.1.

3.1 Mathematical Concept

The Empirical Mode Decomposition (EMD) is an iterative process which decomposes real signal $f(t)$ into simpler signals (modes).

$$f(t) = \sum_{j=1}^{M} \phi_j(t) \tag{1}$$

Each monocomponent signal ϕ_j, with amplitude $r(t)$, should be representable in the form

$$\phi(t) = r(t) sin\theta(t) \tag{2}$$

These monocomponent signals ϕ, called Intrinsic Mode Functions (IMF), are produced by Empirical Mode Decomposition. EMD decomposes the signal into finite number of IMFs. Moreover, these IMFs reflect the intrinsic and reality information of the analyzed signal. Therefore, EMD method is a self-adaptive signal-processing method that is suitable for the analysis of non-linear and non-stationary process[7].

3.2 Intrinsic Mode Function

A function $\lambda(t)$ is defined to be an intrinsic mode function[10], of a real variable t, if it satisfies two characteristic properties:

1. λ has exactly one zero between any two consecutive local extrema.
2. λ has zero local mean.

Part A of Figure 2 shows several IMFs in increasing order generated after applying EMD on a speech segment. Part B shows corresponding FFTs of the produced IMFs. This shows that the IMF generation takes place in the decreasing order of their frequencies.

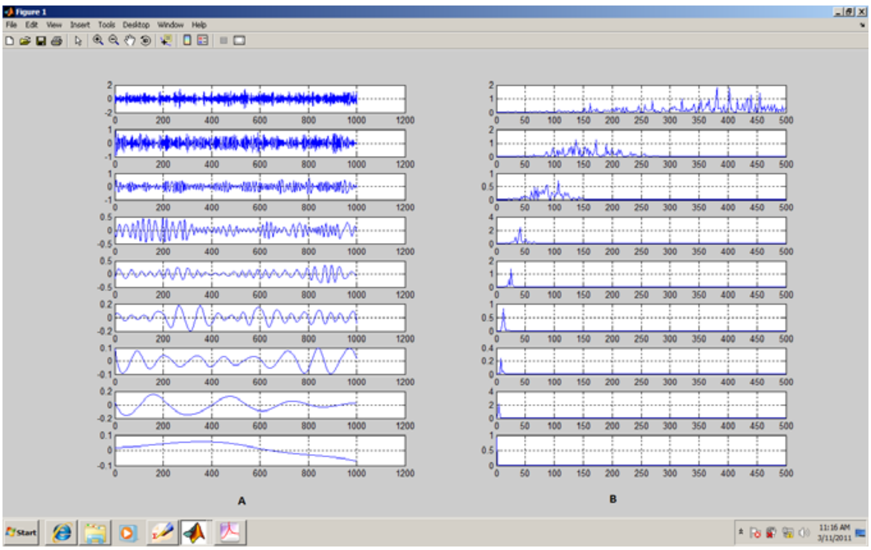

Fig. 2. Intrinsic Mode Functions and their Fourier transforms

The entire method for performing EMD is done through the sifting process. Cubic Spline interpolation is used to link local maxima and minima to form upper envelope and lower envelope of the signal respectively. The mean of these two envelopes is subtracted from the original signal. EMD is obtained after applying this process repeatedly. The sifting algorithm is highly adaptive; it is also unstable. A small change in data can often lead to different EMD[11].

4 Related Work

The results of [5] show that the IMF plots reveal that when noise is added to a clean speech signal, the first few IMFs contain most of noise energy and some of the speech. Our goal is to distinguish which IMF contain the speech or noise. This decomposition pushes a significant amount of the speech energy to latter IMFs along with some residual noise. The reconstruction process is given in Equation 3, which involves combining the n IMFs and the residual $r[n]$.

$$x[n] = \sum_{i=1}^{n} IMF[n] + r[n] \tag{3}$$

Issues such as the stopping criterion for the sifting process or determining a specific spline interpolation for EMD are crucial for the efficiency of the algorithm. The results may vary due to highly adaptive nature of the sifting algorithm and ad hoc nature of using cubic splines. In [3], the cubic splines were replaced by B-splines, which gives an alternative way for EMD. But again this modification does not resolve those issues. The convergence problem has been addressed in [11] using iterating filters, but it provides similar results. In [7], it is shown that the IMFs defined by their energy difference tracking method meet the orthogonality condition and reflect the intrinsic and reality information of the analyzed signal.

The authors in [9] discuss a very good comparison of different assessing alternatives for the sifting process and introduce the use of rational splines which results into tradeoffs relative to the original cubic spline method. They are succeeded to reduce the over- and undershooting problems, but at the expense of more IMFs and more sifting.

In [4], the authors discuss the assets of the EMD as its sparseness and completeness. They also explain the weakness of this algorithm regarding its dependancy on the sifting convergence criterion or interpolation method.The authors describe in [8] the use of Empirical Mode Decomposition for denoising signals in an efficient manner. They suggest that it should be possible to separate out the noise portion from the incoming signal. A very good documentation in [2] is provided on how to write a custom block inside GNU Radio.

5 Scenario for Experiment

5.1 Initial Setup

A typical communication link includes, at a minimum, three key elements: a transmitter, a communication medium (or channel), and a receiver. The ability to simulate all these functions is required to successfully model any end-to-end communication system. As our goal was oriented towards improvement of the signal using Empirical Mode Decomposition, we used Matlab at the initial stage to test EMD and its effect over the noisy signals.

5.2 Simulation on Speech Signal

Since Empirical Mode Decomposition performs better with non-stationary signals, we chose speech signal for our simulation.

The simulation result of [5] shows that for low E_b/N_o, EMD method improves BER performance by approximately $3dB$ gain. These BER improvements can help solving the problem related to call drop outs and improves overall QoS.

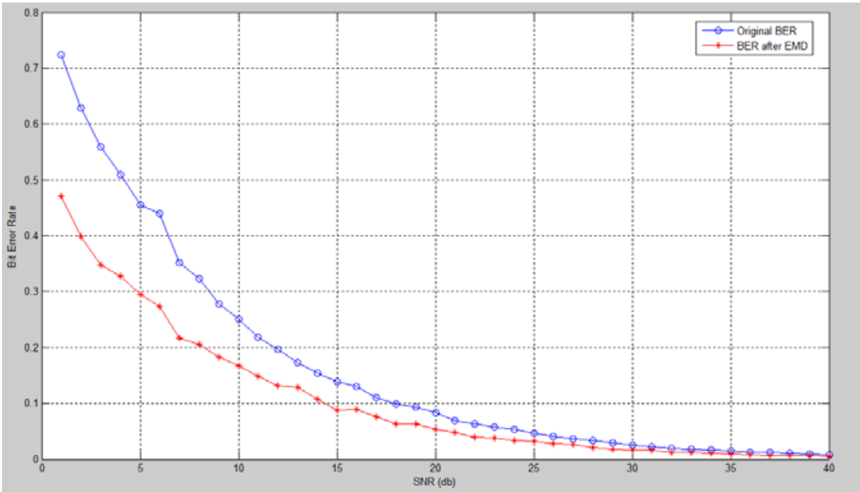

Fig. 3. Effect of EMD over Speech Signal

At the initial stage, we used a sample speech segment. The signal was passed through AWGN channel with varying SNR values. The EMD processed signal and unprocessed signal show significant difference. This is depicted using Figure 3.

5.3 Simulating GSM Signal

At present, GSM and CDMA are maximally used mobile communication signals. Our approach was to explore the effect of EMD as a denoising technique on different types of noise and embedding its functionalities into software defined radio.

At first, we constructed GSM signal by passing a random binary pattern through GMSK modulator, provided inside GNU Radio. This signal was corrupted with three different kinds of noise:gaussian noise, uniform noise and random noise. After applying EMD on this noisy signal, we compared the corresponding outputs produced after the experiment. It is observed that initial IMFs exhibit noise components in the signal more dominantly, hence our approach was to subtract the first few IMFs from the output of EMD. We started with removing only the first IMF from the output, which would contain the maximum noise energy. We refer to this approach as *Case I*.

We carried out simulation by varying different parameters such as type of noise, type of splines used in EMD and using first two IMFs for subtraction. Figure 4 describes the flow graph of the simulation. The analysis of results is shown in Table 1. It is clearly seen that EMD provides a good alternative to reduce bit error rate of the incoming signal.

5.4 Results

This simulation was handled for 2000 samples of GMSK signal and the results show that EMD performs better in presence of Gaussian Noise and it does not have any

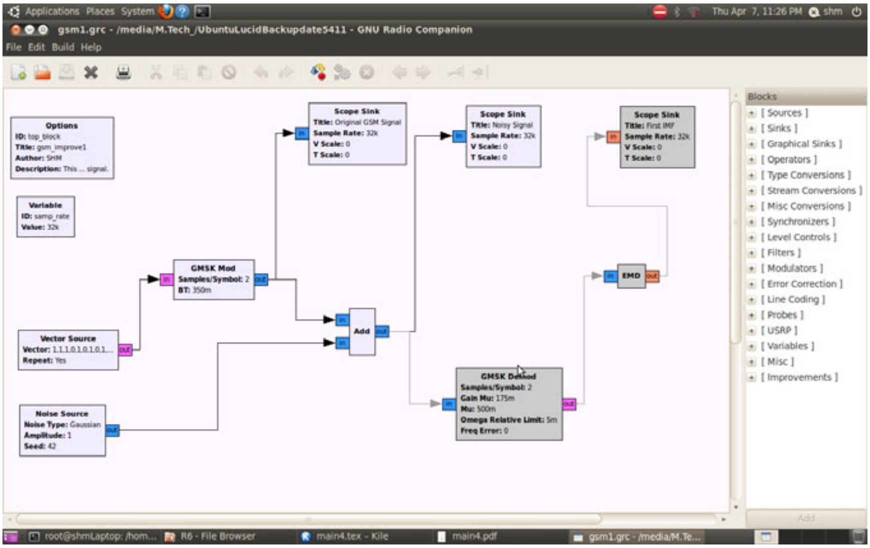

Fig. 4. Set up of Experiment in GNU Radio Companion

Table 1. Comparison of BER with and without EMD removing only the first IMF

Type Of Noise	Time(sec)	BER	BER with EMD
Gaussian	33.5	0.908	0.83
Uniform	19.5	0.914	0.869
Random	17	0.873	0.883

improvement during random noise. For GSM signals, it shows that for increasing SNR values, the BER remains the constant. After performing the simulation, we came to know that EMD processed signal is less noisy as compared to the unprocessed signal.

The underlying observation indicates that EMD should be avoided while random noise is active. We conducted another experiment with a little change in the scenario. We extracted first two IMFs from the noisy signal and deducted them from it. We refer to this approach as *Case II*. The results after this modification are depicted in Table 2.

The result after the modification shows that EMD has the same effect over Gaussian and Uniform noise, whereas there is a significant improvement over random noise in this case. This is because the most of the noise energy is concentrated in the initial IMFs. This may vary for different signals, but usually it may be contained within the first three IMFs.

The naive algorithm for EMD uses cubic natural spline. We implemented *Case II* approach using nearest neighbor spline and linear spline interpolation and their results are presented in Table 3 and Table 4 respectively. Results show that cubic interpolation is the most suitable approach for all the cases.

Table 2. Comparison of BER with and without EMD removing first two IMFs

Type Of Noise	Time(sec)	BER	BER with EMD
Gaussian	33.5	0.91	0.76
Uniform	19.2	0.914	0.734
Random	16.7	0.873	0.76

Table 3. Comparison with Nearest Neighbor Spline

Type Of Noise	Time(sec)	BER	BER with EMD
Gaussian	33.8	0.90	0.76
Uniform	19.4	0.92	0.74
Random	16.8	0.876	0.76

Table 4. Comparison with Linear Spline

Type Of Noise	Time(sec)	BER	BER with EMD
Gaussian	33.8	0.90	0.76
Uniform	19.2	0.92	0.74
Random	16.7	0.875	0.76

6 Conclusion

Empirical Mode Decomposition provides better results for signals having low signal to noise ratios. In addition, it is also observed that this algorithm works well when we remove the first two IMFs from the signal under process.

In presence of the uniform noise and gaussian noise, EMD exhibits about 20% improvement, whereas in case of random noise, for *Case I*, its performance is deteriorated by about 2%. Again, the cubic interpolation gives better result (about 1 to 2% improvement) over other methods.

Empirical Mode Decomposition provides a basis for separating out the noise dominated components from the signal. This is very useful in the data critical applications where maximal error free communication is expected.

7 Future Scope

Simulation of GSM signal for the experiment provides a good insight that wireless communication can be made noise prone at some extent using the method discussed in this work.

In this paper, we presented the advantage of empirical mode decomposition for denoising a signal. This work can be extended by implementing variants of EMD technique and providing more parameters to choose while performing the denoising using

EMD inside GNU Radio. Implementation can be more realistic if any hardware such as USRP is attached to GNU Radio so as to receive the OTA (over the air) files and process these live signals directly.

References

1. Bard, J., Kovarik V.J. : *Software Defined Radio- The Software Communications Architecture.* John Willey and Sons. (2007)
2. Blossom, Eric. How to write a signal processing block. Retrieved from www.gnu.org/software/gnuradio/doc/howto-write-a-block.html
3. Chen, Q., Huang, N., Riemenschneider, S. and Xu, Y. A B-spline approach for Empirical Mode Decompositions. *Advances in Computational Mathematics*, 24:171–195, 2006. 10.1007/s10444-004-7614-3.
4. Equis Sébastien and Jacquot Pierre. The empirical mode decomposition: a must-have tool in speckle interferometry? *Opt. Express*, 17(2):611–623, Jan 2009.
5. Guo, Z., Z. Xu, F. Wang and B. Huang,, 2010. Empirical Mode Decomposition for BER improvement in cellular network. Inform. Technol. J., 9: 146-151.
6. Huang, N.E., et al. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1971):903–995, March 1998.
7. Junsheng, C., Dejie, Y., and Yu, Y. (2006). Research on the Intrinsic Mode Function (IMF) criterion in EMD method. Mechanical Systems and Signal Processing, 20(4), 817-824. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/S0888327005001494
8. McLaughlin, S., Kopsinis, Y. : Empirical Mode Decomposition based denoising techniques.
9. Peel, M.C., G.G.S. Pegram and T.A. McMahon. : *Empirical Mode Decomposition : Improvement and Application.*
10. Sharpley, R.C., Vatchev, V. : *Analysis of the Intrinsic Mode Functions*. Industrial Mathematical Institute, Universitu of South Carolina.
11. Wang, Y., Lin L. and Zhou, H. Iterative filtering as an alternative algorithm for empirical mode decomposition. *Advances in Adaptive Data Analysis*, 1(4):543–560, 2009.
12. Official Gnu Radio Website, http://www.gnuradio.org.
13. Official Website of Joseph Mitola. http://web.it.kth.se/ maguire/jmitola/.