

# A New Model Based Design of SDR for Reducing Error Transmission

S. Hari Krishnan, K.Vidhya

**Abstract:-** In this paper, a SDR model-based concept approach to the implementation of SDR digital communication. The wireless communication systems are a popular prototyping platform. We propose to create a secure, private wireless channel between two SDR (Software Defined Radio) terminals. It is not the safest option to transfer sensitive data wirelessly via the web. We, therefore, offer an alternative short distance use of the internet for the transmission of secured data. We have implemented different modulation and encoding methods and we have compared their performance with Matlab with the main aim of reducing the BER bit error rate. An entire digital communication system is installed and validated via on-air demonstration that includes a transmitter with an encoder and a receiver. For hardware efficiency, synchronization algorithms including offset stage, offset carrier frequency and recovery time are also optimized.

**Index Terms:** SDR, BER, Mat Lab, carrier frequency offset.

## I. INTRODUCTION

A digital communications system (DCS) is designed and tested on a Asynchronous integrated circuit (ASIC) during the traditional development process. The high unrepeatable engineering costs of an ASIC, however, make system prototyping prohibitive. The hardware is modified to perform various operations if necessary. SDR is a middleware with radio and devices connected with the hardware interface. This unique ability gives SDR a greater technological extent. Model design is a promising way to develop quickly a communication system on an SDR platform. Several works are underway in order to implement the communication systems using design models for educational and research purposes targeted at SDR. However, the focus was mainly on the theoretical study of feasibility without practical implementation. While many other works are being done on FPGA communications systems implementations, most focused on hardware using a special algorithm. The modulation of the BPSK symbols, for instance, was developed and implemented using the Xilinx Zynq board targeting SDR [6] Root Raising Cosine (RRC) [5], interpolation filters, and then signal transmission. The modulator and demodulator are designed in Modulation BPSK symbols. The BPSK symbols have also been retrieved with a similar structure.

## II. TRANSMITTER MODEL

On the SDR based Matlab tool, the System Diagram of the BPSK transmitter is mainly composed of a convolution

encoder, symbolic mapping and pulse shaping. No prior coding knowledge is required, since all the units are available in Simulink. It is worth mentioning, These devices are optimally designed to produce HDL code and process the signal of baseband to convert the data into baseband waveform. The SDR radio board then converts the baseband waveform and air transmits the RF band signals.

### A. Convolutional coding

The convolutional encoder architecture. Five sub modules include pilot-payload scatter, pre QPSK pilot training, pilot-payload meeting and payload training. The convolution encoder is available in five submodules. In this module, the data and valid signal is divided when the source data, inclusive of payload and Pilot are present. A pair of two sequential bits of BPSK formation module in a considered as an output with unsigned two-bits and the input is generated from the symbol mapping component. The pilot/payload fraction then divides and divides the payload and pilot bits, so that the encoder submodule only codes the payload. The polynomial of the convolutional encoder is, for example, (101,111). Since, the convolutional encoder is used for doubling the data rate of payload and hence the necessity of doubling the pilot rate is required. Two sequential bits of the pre-BPSK pilot and payload formation module in an output of unsigned two-bit format and symbol mapping element input format. The encoded and pilot data are recast in the payload or pilot meeting module following the pre-BPSK training unit.

### B. Symbol mapping

Gray mapping is considered as a multilevel modulation scheme, where two successive values only change a bit. Gray codes are commonly used in combination with FEC codes for facilitating error correction. The receiver can retrieve the constellation point that flows into adjacent signal area utilizing a mapping procedure. A BPSK modulator block is used in the symbol mapping module to perform the gray mapping, where the data is mapped into two-bits [00, 01, 11, 10] and its complex symbols, which is represented as  $[1+i, -1+i, 1-i, -1-i]$ .

### Fec (forward error correction):

FEC module [7] is used to correct errors on the receiving end in wireless communication. Such errors must have occurred because the medium between Transmitter and receiver has interference, noise or various other impairments. Forward error correction (FEC) is an enhanced data reliability digital signal processing technique.

Revised Manuscript Received on September 10, 2019.

S. Hari Krishnan, PhD Scholar/ECE, Saveetha School Of Engineering, Chennai, Tamilnadu, India.

(E-mail: [s.harikrishnanvpm@gmail.com](mailto:s.harikrishnanvpm@gmail.com))

Dr.K.Vidhya, Professor/ECE, Saveetha School Of Engineering, Chennai, Tamilnadu, India.

(E-mail: [vidhyak.sse@saveetha.com](mailto:vidhyak.sse@saveetha.com))

This is done by inserting redundant data before transmission or storage, which is called an error correction code. Without a reversed channel FEC gives the recipient the ability to resolve all the errors to request the transfer of data.

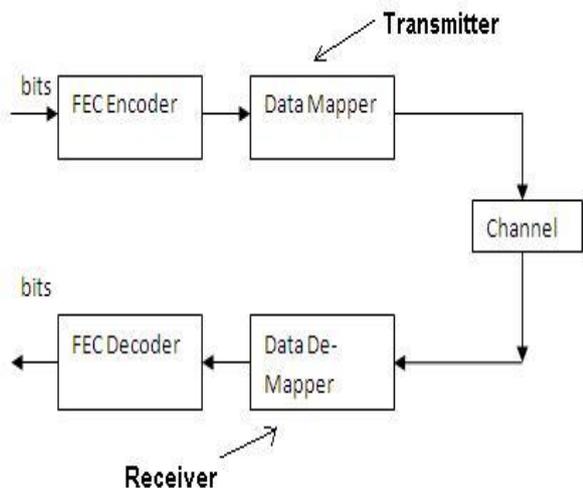


Fig.1 Block Diagram of FEC

**Types:**

- Convolution coding
- Turbo coding
- Viterbi coding

**CONVOLUTIONAL ENCODER:**

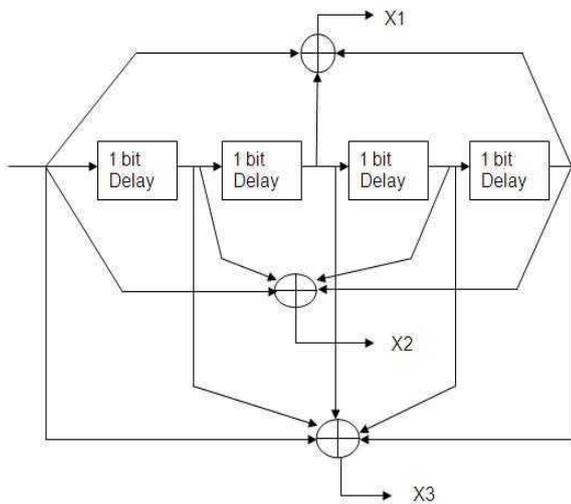


Fig 2.Block Diagram of Convolutional Encoder

Convolutional Code [8] is an error correction code which, by sliding the into a data stream from Boolean polynomial function and generates parity symbols. The encoder's "convolution" is represented by sliding application over the data, resulting in the term "convolutional coding." Starting with *k* memory registers, each of the input bit is used to encrypt data. The registers initially starts with 0 unless or otherwise specified. The encoder consists of an *n* modulo-2 adders.

The encoder outputs n number of symbols that utilizes a generator polynomials and the values in the other registers. These symbols are transmitted after following the code rate. The transmitted bit now shifts all the registration values and wait until the next input bit. If the input bits remain, the

encoder tends to move to zero status until all registers (flush bit termination) return.

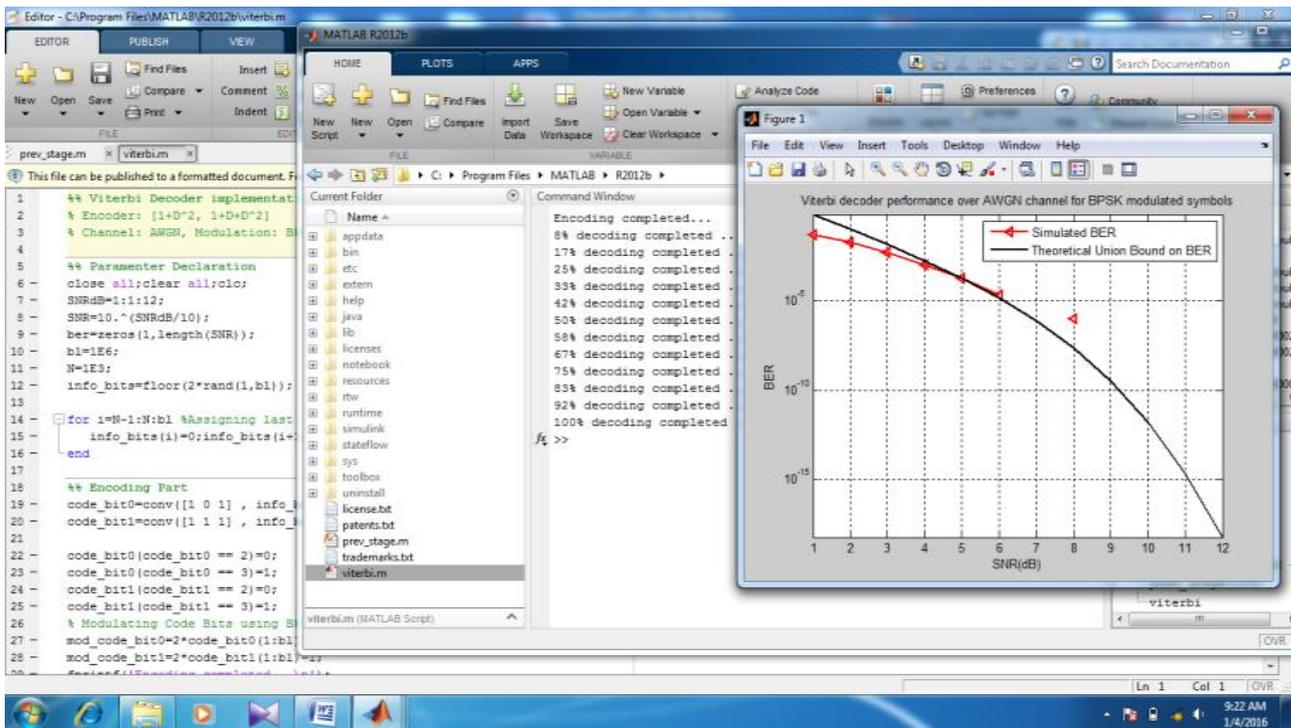
*Viterbi decoder:*

A Viterbi decoder [9] uses the process of Viterbi principle for decoding a bitstream encoded with trellis code. Other algorithms are available for decoding the stream using Fano algorithm. Two metrics are employed by the decoding algorithm: the branch and path metric. The branch metric is used for measuring the transmitter and receiver distance.

*Viterbi decoding steps:*

- Generate random BPSK Modulated symbols [10] +1,-1.
- Encode using convolutional encoder with a half rate generator polynomial [7,5] octal code.
- The encoded bits are passed through AWGN channel.
- The received soft and hard bits are sent through the Viterbi decoder.

### III. SIMULATION RESULT



### IV. CONCLUSION

In this article, a SDR model design that consists of a communication system with the property of channel coding and all the system computing units are implemented. All units are arranged in the form of modules in our design and it is modified based on the data flows and interfaces are matched to the appropriate circuits. The SDR Matlab simulation is also shown by over-air transmission. Simulation assessments are made to evaluate the performance of a model-based design. The result shows that the proposed method is effective than other methods.

### V. REFERENCES

- 1) R. W. Stewart *et al.*, "A low-cost desktop software defined radio design environment using MATLAB, simulink, and the RTL-SDR," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 64-71, Sep. 2015.
- 2) R. W. Stewart, K. W. Barlee, D. S. Atkinson, and L. H. Crockett, *Software Defined Radio Using MATLAB & Simulink and the RTL-SDR*. Glasgow, U.K.: Strathclyde Academic Media, 2015.
- 3) E. Grayver, *Implementing Software Defined Radio*. New York, NY, USA: Springer, 2012.
- 4) Adrat, M., Bernier, S., Buchin, B., Dingman, K., Kovarik, V., Marks, J., ... & Quintana, A. (2017). A technical review of sca based software defined radios: Vision, reality and current status. *Journal of Signal Processing Systems*, 89(1), 85-95.
- 5) Gorlov, A., Gelgor, A., & Nguyen, V. P. (2016). Root-raised cosine versus optimal finite pulses for faster-than-Nyquist generation. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems* (pp. 628-640). Springer, Cham.
- 6) Kamaleldin, A., Mohamed, A., Nagy, A., Gamal, Y., Shalash, A., Ismail, Y., & Mostafa, H. (2017, May). Design guidelines for the high-speed dynamic partial reconfiguration based software defined radio implementations on xilinx zynq fpga. In *2017 IEEE*

*International Symposium on Circuits and Systems (ISCAS)* (pp. 1-4). IEEE.

- 8) Marcotte, R. J., & Olson, E. (2016, May). Adaptive forward error correction with adjustable-latency QoS for robotic networks. In *2016 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 5283-5288). IEEE.
- 9) Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017, August). Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 1243-1252). JMLR. org.
- 10) Adiono, T., Aska, Y., Fuada, S., & Purwita, A. A. (2017). Design of an OFDM system for VLC with a Viterbi decoder. *IEIE Transactions on Smart Processing and Computing (SPC)*, 6(6), 455-465.
- 11) Bhagyashree, K., Ramakrishna, S., & Kumar, P. (2019). Analysis of PAPR for Performance QPSK and BPSK Modulation Techniques. In *Emerging Research in Computing, Information, Communication and Applications* (pp. 621-627). Springer, Singapore.