

# Simulation & Performance Study of Wireless Sensor Network (WSN) Using MATLAB

Qutaiba Ibrahim Ali  
Computer Eng. Dept.  
Mosul University  
Mosul, Iraq  
[qut1974@gmail.com](mailto:qut1974@gmail.com)

Akram Abdulmaowjod  
Computer Eng. Dept.  
Mosul University  
Mosul, Iraq  
[akram.dawood80@gmail.com](mailto:akram.dawood80@gmail.com)

Hussein Mahmood Mohammed  
Computer Eng. Dept.  
Mosul University  
Mosul, Iraq  
[husmah1984@gmail.com](mailto:husmah1984@gmail.com)

**Abstract**—A wireless sensor network consists of spatially distributed autonomous sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants. Different approaches have used for simulation and modeling of SN (Sensor Network) and WSN. Traditional approaches consist of various simulation tools based on different languages such as C, C++ and Java. In this paper, MATLAB (7.6) Simulink was used to build a complete WSN system. Simulation procedure includes building the hardware architecture of the transmitting nodes, modeling both the communication channel and the receiving master node architecture. Bluetooth was chosen to undertake the physical layer communication with respect to different channel parameters (i.e., Signal to Noise ratio, Attenuation and Interference). The simulation model was examined using different topologies under various conditions and numerous results were collected. This new simulation methodology proves the ability of the Simulink MATLAB to be a useful and flexible approach to study the effect of different physical layer parameters on the performance of wireless sensor networks.

**Keywords:** Wireless sensor networks, MATLAB Simulation, Signal to Noise Ratio, Interference, Bluetooth.

## I. INTRODUCTION

A wireless sensor network consists of spatially distributed autonomous sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance. They are now used in many industrial and civilian application areas, including industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, healthcare applications, home automation, and traffic control [1, 2, 3].

A smart sensor node is a combination of sensing, processing and communication technologies. Figure 1 shows the basic architectural components of a sensor node. The sensing unit senses the change of parameters, signal conditioning circuitry prepares the electrical signals to convert to the digital domain, the sensed analog signal is converted and is used as the input to the application algorithms or processing unit, the memory helps processing of tasks and the transceiver is used for communicating with other sensors or the base stations or sinks in WSN.

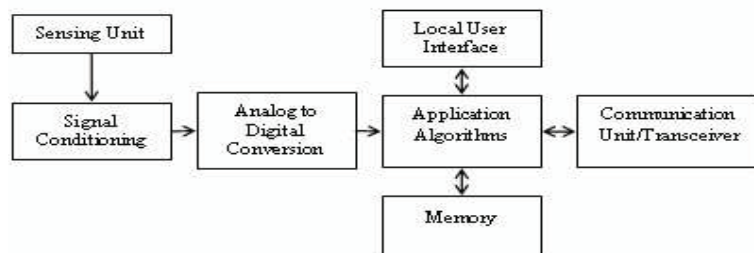


Figure 1. Basic architectural components of a smart sensor

Sensors can monitor temperature, pressure, humidity, soil makeup, vehicular movement, noise levels, lighting conditions, the presence or absence of certain kinds of objects or substances, mechanical stress levels on attached objects, and other properties. Their mechanism may be seismic, magnetic, thermal, visual, infrared, acoustic, or radar. A smart sensor is also capable of self-identification and self-diagnosis. The mechanisms of smart sensors work in one of three ways: by a line of sight to the target (such as visual sensors), by proximity to target (such as seismic sensors), and by propagation like a wave with possible bending (such as acoustic sensors) [4, 5].

Sensor networks are predominantly data-centric rather than address-centric. In such a network, queries are directed to a region containing a cluster of sensor nodes rather than specific sensor addresses. Given the similarity in the data obtained by sensors in a dense cluster, aggregation of the data is performed locally. That is, a summary or analysis of the local data is prepared by an aggregator node within the cluster, thus reducing the communication bandwidth requirements. Aggregation of data increases the level of accuracy and incorporates data redundancy to compensate node failures. A network hierarchy and clustering of sensor nodes allows for network scalability, robustness, efficient resource utilization and lower power consumption which are some of the key issues in WSN [6].

Fundamental objectives of sensor networks are reliability, accuracy, flexibility, cost effectiveness and ease of deployment. Key characteristics and benefits of WSN (Wireless Sensor Networks) are outlined below [7]:

- Sensing accuracy: The utilization of a larger number and variety of sensor nodes provides potential for greater accuracy in the information gathered as compared to that obtained from a single sensor.
- Area coverage: This implies that fast and efficient sensor network could span a greater geographical area without adverse impact on the overall network cost.
- Fault tolerance: Device redundancy and consequently information redundancy can be utilized to ensure a level of fault tolerance in individual sensors.
- Connectivity: Multiple sensor networks may be connected through sink nodes, along with existing wired networks (e.g. Internet). The clustering of networks enables each individual network to focus on specific areas or events and share only relevant information.
- Minimal human interaction: Having minimum human interaction makes the possibility of having less interruption of the system.
- Operability in harsh environments: Sensor nodes, consisting of robust sensor design, integrated with high levels of fault tolerance can be deployed in

harsh environments that make the sensor networks more effective.

- Dynamic sensor scheduling: Implying some scheduling scheme, sensor network is capable of setting priority for data transmission.

## II. LITERATURE REVIEW

Different approaches were used for simulation and modeling of SN and WSN (more than 40 active simulators in this area). They operate in different levels of simulating: hardware emulation, operating system and application level. The supported hardware platforms are mainly TmoteSky [8], MSB (Modular Sensor Board) [9], MICA [10], MICA2 [11], MICAZ [12] and etc. Operating system is mostly either Contiki [13] or TinyOS [14] therefore the programming language for sensor nodes is Contiki C or NesC [15] respectively. Application-level programs are usually written in C++, Java or Python.

Examples of the important and widely used simulators for each of the simulating levels includes: COOJA as a Contiki simulator[16], Bonnmotion: mobility simulator[17], OMNeT++: a general-purpose popular simulator[18], Castalia and MiXiM: two strong WSN specialized simulators based on OMNeT++[19,20], TOSSIM: TinyOS simulator and MICA2 emulator[21,22], Simulators 7 Avrora: instruction-level emulator for AVR processor and MICA2 hardware[23], SENS: a detailed environment simulator[24], NS2: a general purpose popular simulator[25] and OPNET: a general-purpose simulator with WSN support[26].

In this paper, Simulink MATLAB was adopted to be the simulation tool of wireless sensor network (WSN). The main advantage of the suggested method is to determine the effect of the different channel parameters (i.e., Signal to Noise ratio, Attenuation and Interference) on the system behavior.

## III. THE PROPOSED WSN SIMULATION METHODOLOGY

The environment in which we build our simulation model was MATLAB. MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. The name MATLAB stands for matrix laboratory. Simulink is an environment for multidomain simulation and Model-Based Design for dynamic and embedded systems. It provides an interactive graphical environment and a customizable set of block libraries that facilitates the designing, simulation, implementation, and testing a variety of time-varying systems, including communications, controls, signal processing, video processing, and image processing[27].

### A. Simulating a simple WSN in Simulink MATLAB

In order to demonstrate the concepts of the suggested simulation methodology, a simple WSN model was built as shown in figure 2. This network consists of three sensors (slaves) sending their measured data samples to a master node.

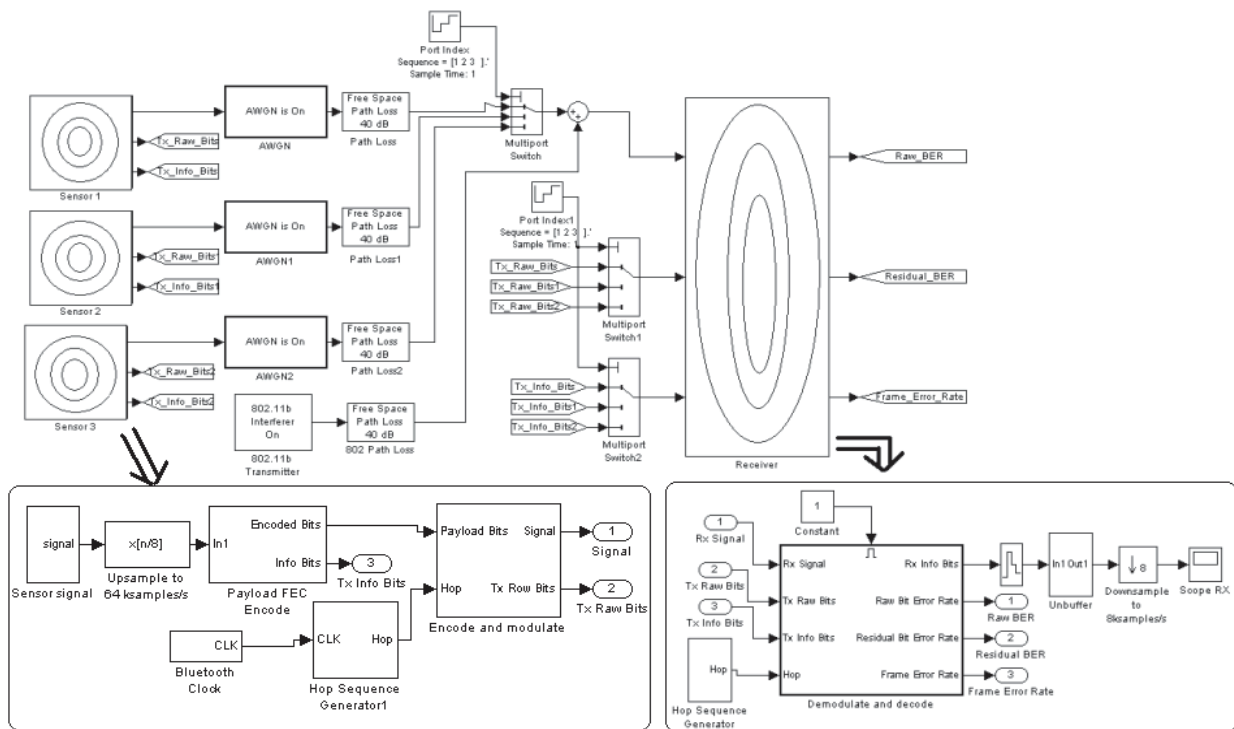


Figure 2. Simple WSN model architecture

The architecture of the system could be explained as follows:

The transmitter:

This system of transmission consists of bluetooth technology that is considered as the backbone of transmission operation.

Bluetooth is a short-range radio link technology that operates in the 2.4 GHz Industrial, Scientific, and Medical (ISM) band. In this system we modulate the signal using Gaussian frequency shift keying (GFSK) over a radio channel with maximum capacity of 1 Mbps. The transmitter consists of the following blocks:

- Sensor signal stage: It is represented by the sensor which senses the physical signal such as temperature, pressure...etc, then transducer them into an electrical signal. In addition, this stage includes the A/D converter which converts the signal from Analog to Digital using 256 quantization level.
- Up-sample to 64ksamples/s: Up-samples the input to a higher rate by inserting zeros between samples.
- Payload FEC encode: Encodes the data to enable error correction – an FEC encoder may include a

binary convolutional encoder followed by a puncturing device.

- Bluetooth Clock: Each Bluetooth device has a free-running 28-bit Bluetooth clock. The clock ticks 3,200 times per second or once every 312.5  $\mu$ sec, representing a clock rate of 3.2 KHz.
- Hop Sequence Generator: For devices to communicate with each other, they must transmit and receive on the same frequency at the same time. The hop sequence generator generates a sequence of hop frequencies in the range 0 to 78. It can generate either the connection state hop sequence, a random white sequence, or be fixed.
- Encode and modulate: The 366 data bits are transmitted at 1 Mbps and modulated using Gaussian frequency shift keying (GFSK). GFSK effectively transmits +150 kHz signal relative to the carrier for a 1bit, and a 150 kHz signal for a 0 bit. The carrier signal is generated in the Simulink model by a baseband MFSK block set to 79 symbols and a separation of 1MHz. If a hop frequency value 0 is input, a -39MHz complex sinusoid is generated. If a 1 is entered, a -38 MHz complex sinusoid is generated and so on. In the model, the hop sequences are generated by a simple random number generator, not

using the actual method specified in the standard. The transmitter is turned off after 366 bits using a Gain block to multiply the frame with a mask of 36600 ones and 26500 zeros.

The medium consists of the following blocks:

- **AWGN Channel:** The AWGN Channel block adds white Gaussian noise to a real or complex input signal. When the input signal is real, this block adds real Gaussian noise and produces a real output signal. When the input signal is complex, this block adds complex Gaussian noise and produces a complex output signal.
- **Path Loss:** Reduce the amplitude of the input signal by the amount specified. The loss can be specified directly using the "Decibels" mode, or indirectly using the "Distance and Frequency mode". The reciprocal of the loss is applied as a gain, e.g., a loss of +20 dB, which reduce the signal by a factor of 10 corresponds to a gain value of 0.1.
- **802.11b interferer:** Add signals that have the same frequency of the data signal to make interference between the data signal and other signals.
- **Multipoint Switch:** The Multipoint Switch block chooses between a number of inputs. The first input is called the control input, while the rest of the inputs are called data inputs. The value of the control input determines which data input is passed through to the output port.

The receiver consists of the following blocks:

- **Hop Sequence Generator:** same as mentioned earlier.
- **Demodulate and decode:** This block is used to extract the original information-bearing signal from a modulated carrier wave, and to recover the information contents in it.
- **Zero-Order Hold:** The block samples and holds its input for the specified sample period. The block accepts one input and generates one output, both of which can be scalar or vector. If the input is a vector, all elements of the vector are held for the same sample period.
- **Un-buffer:** The block un-buffers an Mi-by-N frame-based input into a 1-by-N sample-based output. That is, inputs are un-buffered row-wise so that each matrix row becomes an independent time-sample in the output. The rate at which the block receives inputs is generally less than the rate at which the block produces outputs.
- **Down-sample to 8ksamples/s:** Down-samples the input to a lower rate by deleting the repeating samples.

- **Scope RX:** To display the received signal and compare it with the original signal to discover the system behavior.

As known, a piconet can includes up to seven slaves and one master. In this example three signals are sent from three sensors (slaves) to the receiving component (master) representing one piconet , the information obtained by the sensors are used to estimate the Bluetooth performance as well as to study the media effect. Noise and interference are added to the signals in order to simulate the channel effect and measure Bit Error Rate (BER) and Frame Error Rate (FER). Figure 3 shows the original signals that sent from the three sensors.

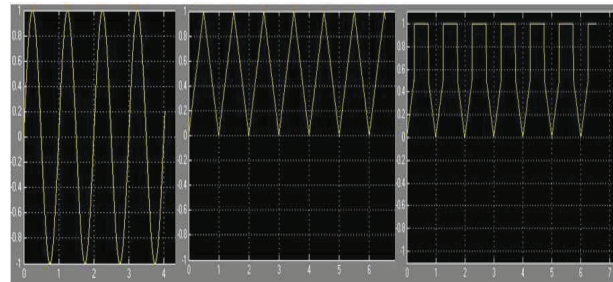


Figure 3. Original sensors signals

Figure 4 shows received signals with different Signal to Noise Ratio (SNR) for simple model.

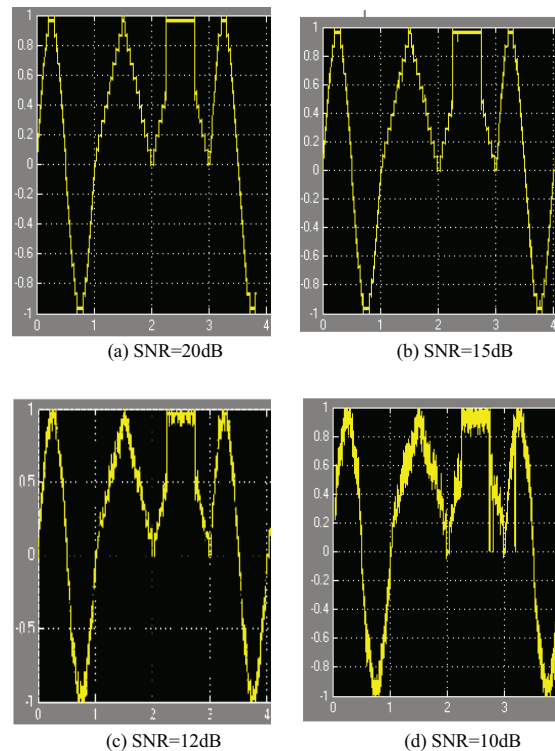


Figure 4. Received signals with different SNR for simple model

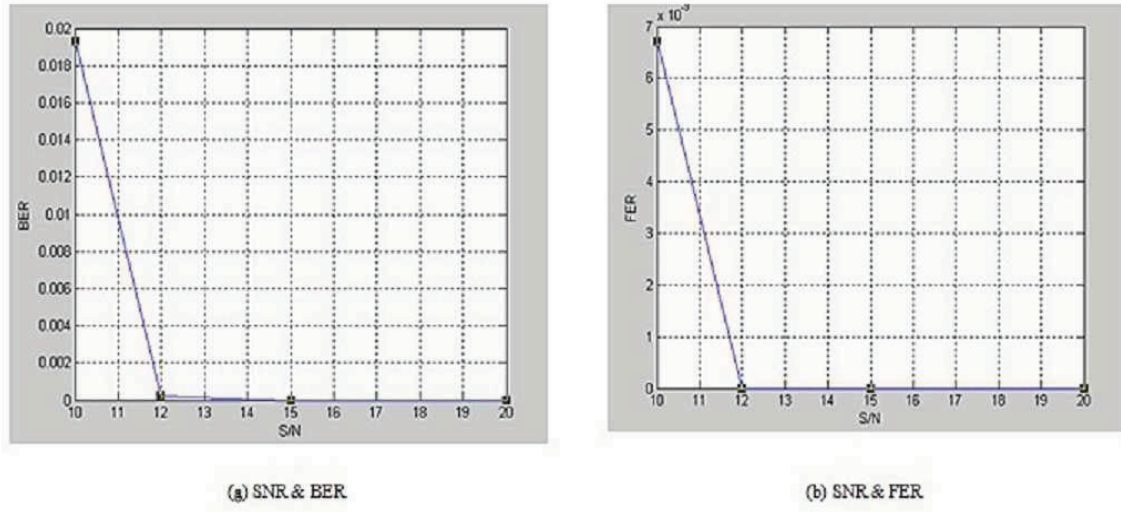


Figure 5. Relationship between SNR & (BER, FER) for simple model

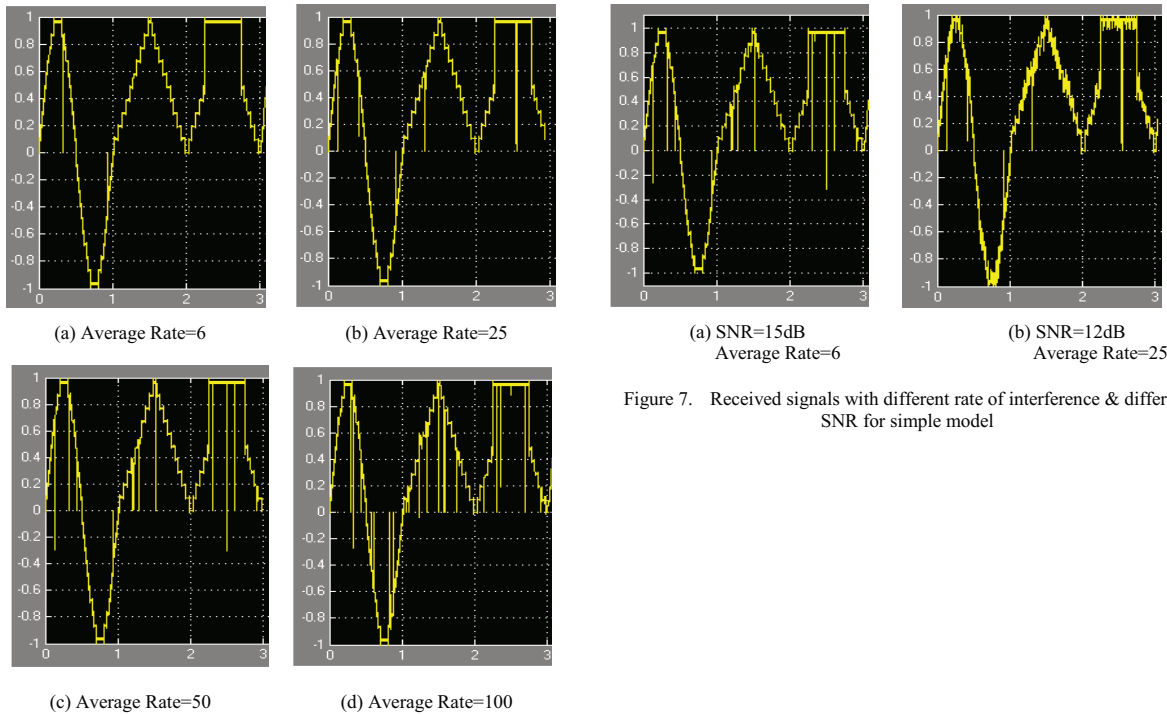


Figure 6. Received signals with different rate of interference for simple model

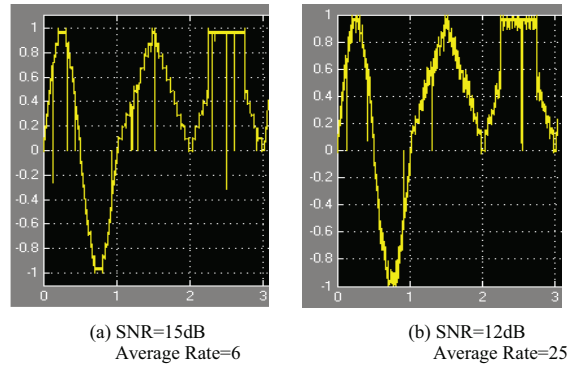


Figure 7. Received signals with different rate of interference & different SNR for simple model



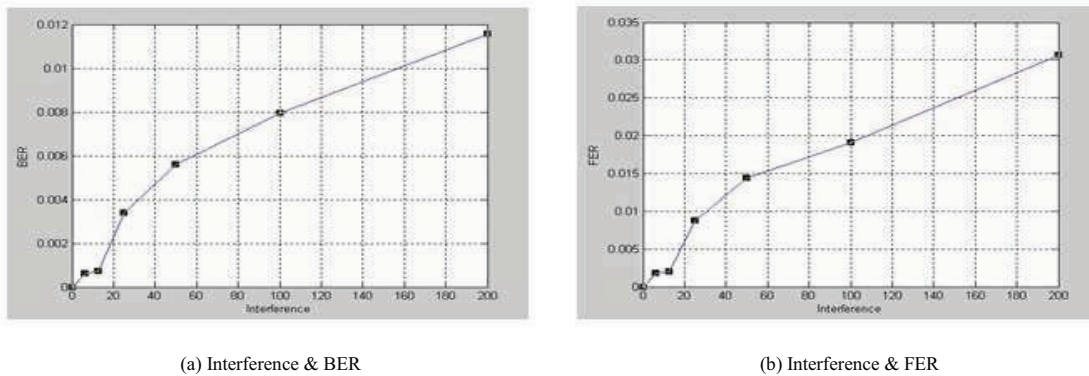


Figure 8. Relationship between interference & (BER, FER) for simple model

**B. More complex Example**

As known from Bluetooth operation, each piconet consists of one master and seven slaves and each master of a specific piconet may acts as a slave for another piconet which means the ability to expand the network to respond to more than seven sensors.

In this example two piconets are connected, so that the first piconet consists of three sensors connected to the master, and the later is connected as a slave to the second piconet. The second piconet consists of two slaves and one master as shown in Figure 9 and Figure 10.

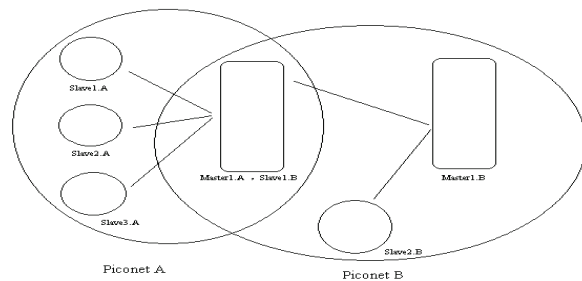


Figure 9. WSN model with masters & slaves

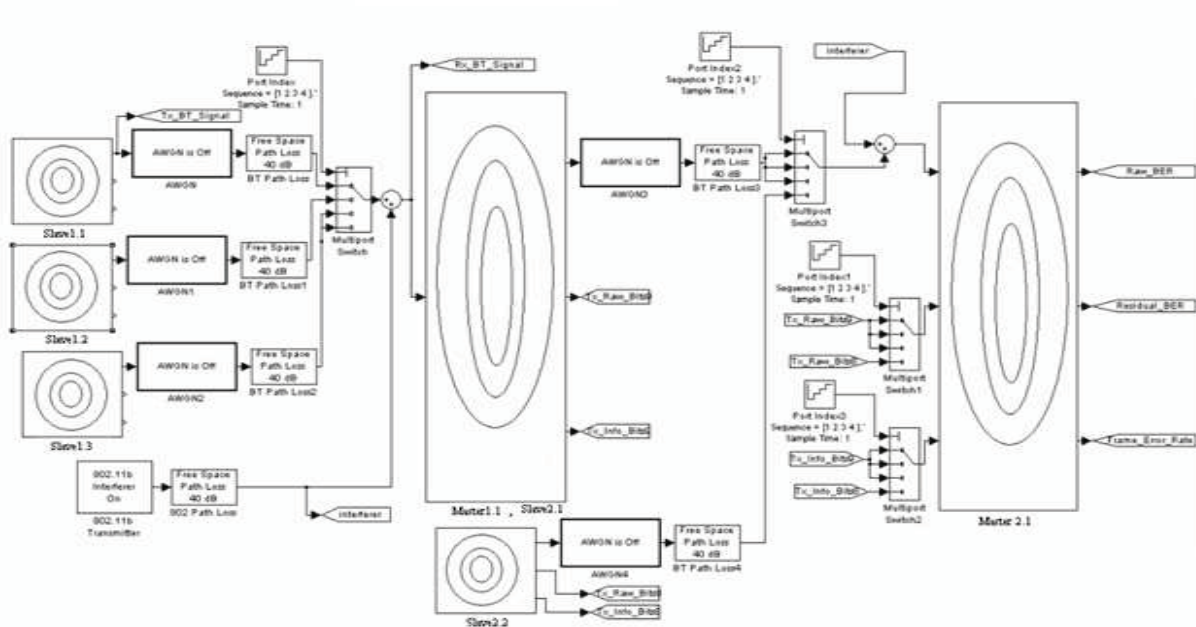


Figure 10. A complex WSN model architecture

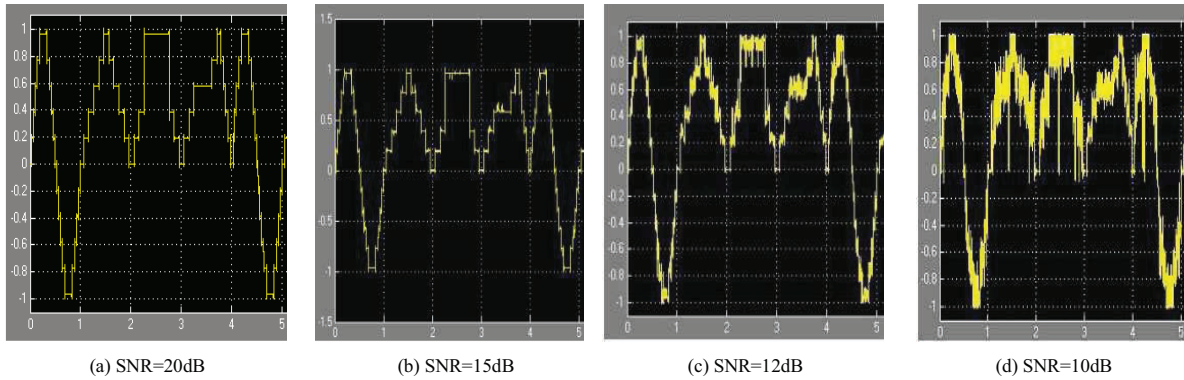


Figure 11. Received signals with different rate of interference for complex model

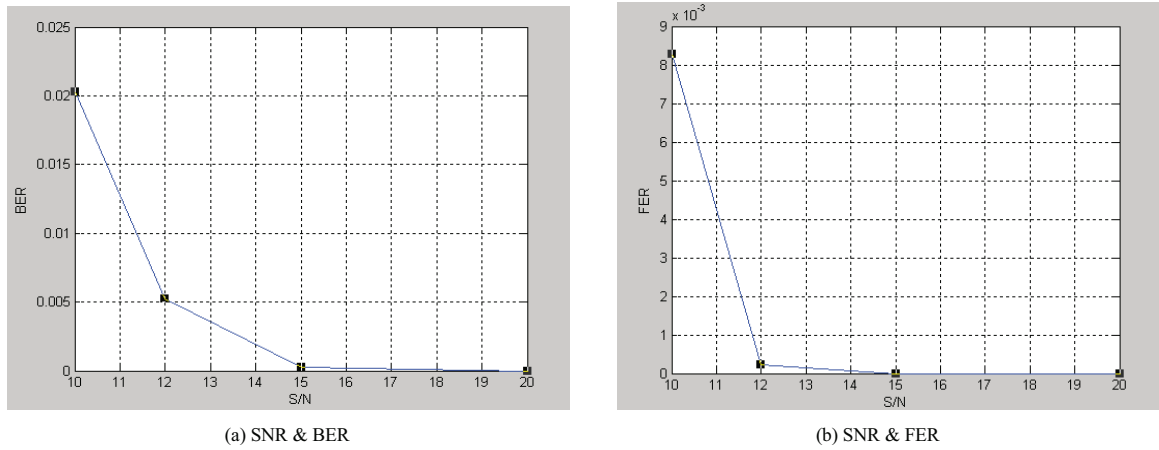


Figure 12. Relationship between SNR & (BER, FER) for complex model

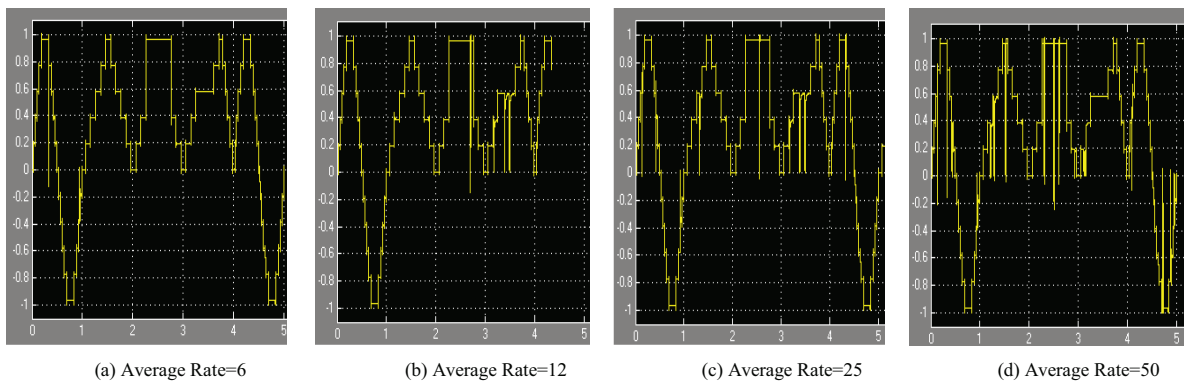


Figure 13. Received signals with different rate of interference for complex model

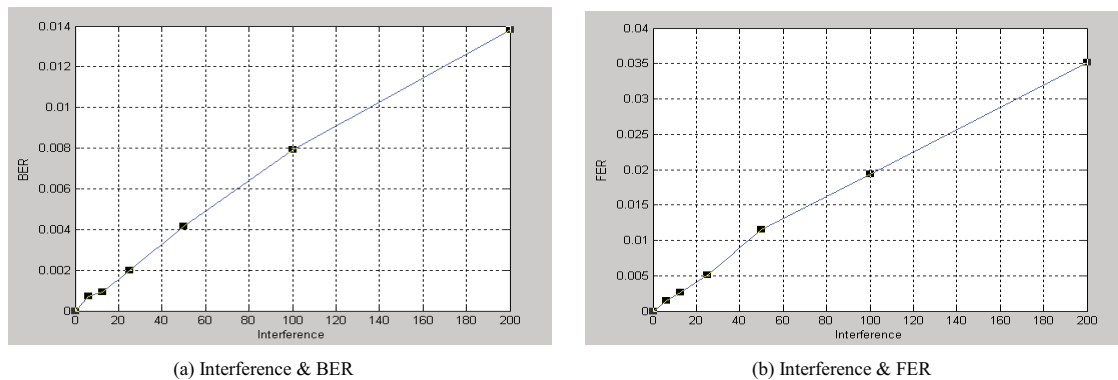


Figure 14. Relationship between interference & (BER, FER) for complex model

From the above results, it is obvious that the behavior of the system was successfully described using the suggested simulation methodology. It is also important to mention that this simulation method provides the ability to change the different system parameters to create new environment and hence, new simulation scenarios.

#### IV. CONCLUSIONS

In this paper, a new simulation methodology of wireless sensor networks (WSN) was presented. Simulink MATLAB was used as the tool to build the simulation environment. The strength of this simulation method falls in the ability to study the effect of different physical layer parameters (channel noise and interference, Signal to noise ratio...etc.) on the system behavior. The other advantage of this method is its flexibility in building the end nodes and sensors. This simulation methodology could be used to build different WSN types and opens the doors to use the MATLAB in this new field.

#### REFERENCES

- [1] W. Granzer, W. Kastner, G. Neugschwandtner, and F. Praus, "A modular architecture for building automation systems," in *Proc. 6th IEEE WFCS*, 2006, pp. 99–102.
- [2] A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1130–1151, 2005.
- [3] J. Rabaey, M. Ammer, J. da Silva, J.L., D. Patel, and S. Roundy, "Picoradio supports ad hoc ultra-low power wireless networking," *Computer*, vol. 33, no. 7, pp. 42–48, July 2000.
- [4] Gs Gupta, SC Mukhopadhyay, M. Sutherland, S. Demidenko, "Wireless Sensor Network for Selective Activity Monitoring in a home for the Elderly," *Proceedings of 2007 IEEE IMTC conference*. Poland, Warsaw 2007; 1(3): 1-6.
- [5] E. Callaway, P. Gorday, L. Hester, "Home Networking with IEEE 802.15.4: A Developing Standard for Low-Rate Wireless Personal Area Networks," *IEEE Commun Mag* 2002; pp. 69-77.
- [6] IF Akyildiz, Y. Sankarasubramaniam Cayirci E. "A Survey on Sensor Networks". *IEEE Commun Mag* 2002; pp. 102-114.
- [7] Fl Lewis, "Wireless Sensors Networks, Smart Environments: Technologies, Protocols, and Applications," ed. Cook DJ, Das SK, John Wiley, New York, 2004; pp. 1-18.
- [8] Tmotesky datasheet. <http://www.sentilla.com/pdf/eol/tmoteskydatasheet.pdf>. Retrieved: 2009-05-26.
- [9] MSB: Modula rsenso rboard datasheet. [www.scatterweb.com/downloads/MSB-datasheet-doc1.0-en.pdf](http://www.scatterweb.com/downloads/MSB-datasheet-doc1.0-en.pdf). Retrieved: 2009-05-26.
- [10] MICA, MICA2, MICAZ. [z2.inf.tudresden.de/dargie/wsn/slides/students/MICA.ppt](http://z2.inf.tudresden.de/dargie/wsn/slides/students/MICA.ppt). Retrieved: 2009-05-26.
- [11] MICA2 datasheet. [www.xbow.com/Products/Product\\_pdf\\_files/Wireless\\_pdf/MICA2\\_Datasheet.pdf](http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf). Retrieved: 2009-05-26.
- [12] MICAZ datasheet. [www.xbow.com/Products/Product\\_pdf\\_files/Wireless\\_pdf/MICAZ\\_Datasheet.pdf](http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAZ_Datasheet.pdf). Retrieved: 2009-05-26.
- [13] Contiki, an open-source operating system for wireless networked embedded systems. <http://www.sics.se/contiki/>. Retrieved: 2009-05-26.
- [14] TinyOS, an open-source operating system for wireless embedded sensor networks. <http://www.tinyos.net/>. Retrieved: 2009-05-26.
- [15] NesC, the sensor network programming language on TinyOS operating system. <http://nesc.sourceforge.net/>. Retrieved: 2009-05-26.
- [16] Contiki in the baltic sea. <http://www.sics.se/contiki/projects/contiki-in-the-baltic-sea.html>. Retrieved: 2009-05-26.
- [17] Bonnmotion. <http://iv.cs.uni-bon.de/wg/cs/applications/bonnmotion/>. Retrieved: 2009-05-27.
- [18] OMNeT++. <http://www.omnetpp.org/>. Retrieved: 2009-05-29.
- [19] Athanassios Boulis. Castalia, a simulator for wireless sensor networks and body area networks, version 2.0, user's manual. May 2009. Retrieved: 2009-06-27.
- [20] MiXiM. <http://mixim.sourceforge.net/>. Retrieved: 2009-05-29.
- [21] TOSSIM. <http://docs.tinyos.net/index.php/TOSSIM>. Retrieved: 2009-05-26.
- [22] David Curren. A survey of simulation in sensor networks. 2005.
- [23] AVR processor. <http://www.atmel.com/products/avr/>. Retrieved: 2009-05-26.
- [24] G. Chen, J. Branch, M. Pflug, L. Zhu, and B. Szymanski, "SENSE: A Wireless Sensor Network Simulator," *Advances in Pervasive Computing and Networking*, Springer, New York, pp. 249-267, 2004.
- [25] I. Downard, "Simulating sensor networks in NS-2," *NRL/FR/5522—04-10073*, Naval Research Laboratory, Washington, D.C., May 2004.
- [26] OPNET Web Site:<http://www.OPNET.com>
- [27] MATLAB Web Site: <http://www.mathworks.com/>