ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Special Issue 16, May 2016

PILOT/SIGNATURE PATTERN BASED ADAPTIVE MODULATION TRACKING

[1]S. SELVAKUMAR, [2]Dr.S.RAVI
Research scholar, ECE Department, SCSVMV University
Professor, ECE Department, Dr. M.G.R. Educational and Research Institute University
E-mail: [1]Sell84@gmail.com, [2]ravi_mls@yahoo.com

Abstract: In orthogonal frequency division multiplexing (OFDM) systems over fast-varying fading channels, channel estimation and tracking is generally carried out by transmitting known pilot symbols in given positions of the frequency-time grid. The traditional approach consists of two steps. First, the least-squares (LS) estimate is obtained over the pilot subcarriers. Then, this preliminary estimate is interpolated/smoothed over the entire frequency-time grid. In this paper, we propose to add an intermediate step, whose purpose is to increase the accuracy of the estimate over the pilot subcarriers. The presented techniques are based on the observation that the wireless radio channel can be parametrized as a combination of paths, each characterized by a delay and a complex amplitude. The amplitudes show fast temporal variations due to the mobility of terminals while the delays (and their associated delay-subspace) are almost constant over a large number of OFDM symbols. We propose to track the delay-subspace by a subspace tracking algorithm and the amplitudes by the least mean square algorithm (or modifications of the latter). The approach can be extended to multiple input multiple output OFDM or multicarrier code-division multiple-access systems. Analytical results and simulations prove the relevant benefits of the novel structure.

Introduction:

Each modulated signal is preceded by a unique 'N' bit pilot sequence (Manton, JH 2001). A switch in the transmitter shown in Figure 1 selects one of the modulated signals which have to be transmitted. AM signal waveform along with its pilot sequence is selected by the switch1. The FM signal is selected by the switch 2. The PSK signal is selected by the switch 3. The selected signal is sent through the communication channel

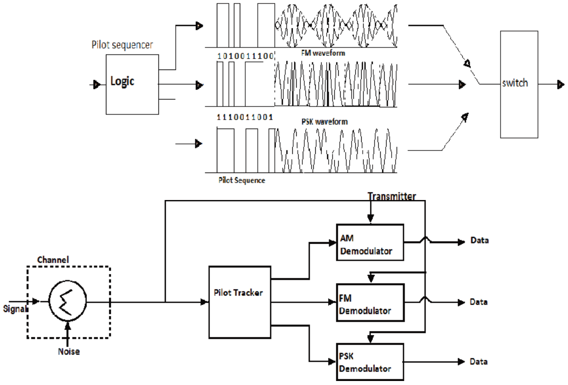


Figure 1 & 2 Pilot pattern based transmitter/ receiver

At the receiver shown in Figure 2, the received signal is given to the pilot tracker which automatically identifies the type of modulation by which the transmitted signal is modulated. The pilot tracker also uses the same type of mechanism used during the generation of Pilot Sequence at the transmitter side to generate the Pilot Sequence. The incoming pilot sequence and the generated pilot sequence are compared by the tracker and accordingly the modulation is identified. The correlation is done between the received Pilot Sequence and the generated Pilot Sequences at the pilot tracker. The pilot sequence which correctly matches with the recovered pilot sequence helps to find the type of modulation done for the received signal. Accordingly, the tracked signal is sent to its respective demodulators. The AM demodulator, demodulates the received signal and gives out the AM signal. The FM Demodulator gives out the FM signal and the data is obtained at the output of the PSK demodulator.

Adaptive Clustering

The MANET is a wireless network, where nodes communicate with each other using multi hop method in an infrastructure less environment and relies on a code division access scheme for data transfer. In this network architecture, nodes are organized into non-overlapping clusters. The clusters are controlled independently and as the nodes move, they are reconfigured dynamically (Li, CR & Gerla, M 1997)**.**

The advantage of this network architecture includes;

1. Due to node clustering, spatial reuse of the bandwidth is possible.
2. Bandwidth can be shared or reserved in each cluster.
3. The clustering method enables the architecture to be stable even during topological changes caused by node motion, node failure and node insertion/removal.

This proposed architecture also helps in providing an efficient and stable infrastructure for the integration of different types of traffic in a dynamic radio network.

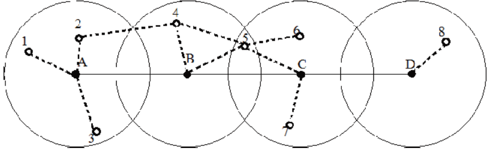


Figure 3 Conventional Cellular networks (Single-hop)

Figure 3 shows, the single hop cellular model wireless networks where, A, B, C and D are four fixed base stations which are connected by a wired backbone. Nodes 1 through 8 are all mobile nodes. A mobile node is a node which is only one hop away from a base station. Communication between two mobile nodes takes place through fixed base station and the wired backbone.

Multi hopping is defined as the ability of the radios to relay packets from one node to another without the use of base stations. It is mainly employed during disaster management in situations such as fire, earthquakes etc. Multi hopping through wireless repeaters reduces the battery power and increase network capacity.

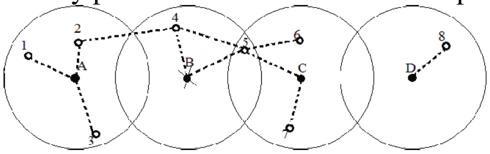


Figure 4 Multi-hop situation when base station B fails occurs

In a single hop network, if a base station fails, a mobile node may not be able to access the wired network. But, in multi hop network as shown in Figure 4, if a base station, say, B fails, node 4 will access the base stations A or C through node 2 or node 5 which in turn acts as wireless multi-hop repeaters.

Scheduler Based Implementation

In this work, the tracking algorithm (section 3.8 of this thesis) is implemented to suit realtime and includes a priority handler. Based on the priority of a node, the availability of the modulation scheme is varied.

For example:

1) Total number of slots= N
2) Total number of nodes= M
3) Different available modulation= K
   where the possible 'K' schemes are known to receiver.
4) Priority Table in Table 6.1 shows the priority in the transmission of 'M' nodes which are dynamic.

Table 1 Priority for each Node during various time interval (M= 4)

Node	Priority during time	Priority during	Priority during time
1	04	03	02
2	01	04	02
3	02	01	02
4	03	02	01

IMPLEMENTATION OF AUTOMATIC MODULATION IDENTIFICATION

The transmitter and receiver hardware architecture for transmitting various digitally modulated signals and receiving the particular modulated signal is discussed in the following topics.

Transmitter

Figure 5 shows the block diagram of the transmitter implemented in SDR. The Modulation selection switch selects a particular type of Digital Modulator to modulate the carrier signal generated by the sine/square wave carrier generator circuit using the information signal. The selection is based on the provided information about the dynamic channel after the mobility of nodes take place. The Digitally modulated signal before being transmitted is sent through the digital to analog converter, so that the signal can be sent as analog signal through the channel. Whenever there is a change in the node location and there by a change in the channel, the type of modulation concerned with that channel is chosen .

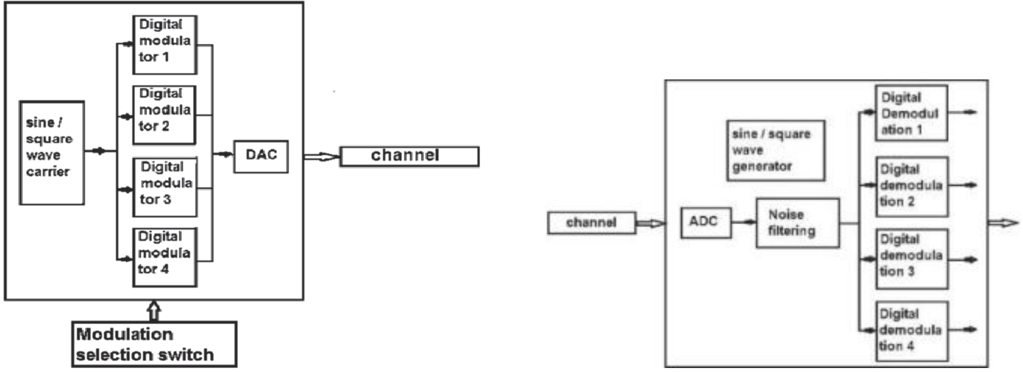


Figure 5 & 6 Block diagram of the SDR Transmitter / Receiver

6Receiver

At the receiver (Figure 6), the signal is received and sent to the analog to digital converter to convert the analog signal back to Digital Modulated signal. The noise present in the signal is filtered out using the Noise filtering circuit. The signal is then given to its respective Digital Demodulator, which demodulates the signal back to the original information signal. The respective demodulator is determined implementing the Automatic Modulation Identification technique.

Hardware Implementation

This FL 2440 ARM9 Embedded development board is used for designing and implementing the Automatic Modulation Identification technique.

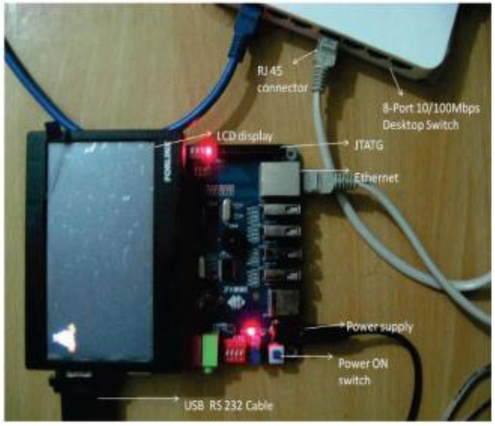


Figure 7 FL 2440 ARM9 Embedded development board

Data from one node (Personal Computer) is sent to the board in a serial manner using the USB RS 232 cable. The program helps in identifying a modulation type and the data is modulated using that type and sent to the next node which is connected to the receiving node. The next node helps in processing the header and finding the exact modulation technique. Figure 7 shows the FL2440 ARM9 Embedded development board.

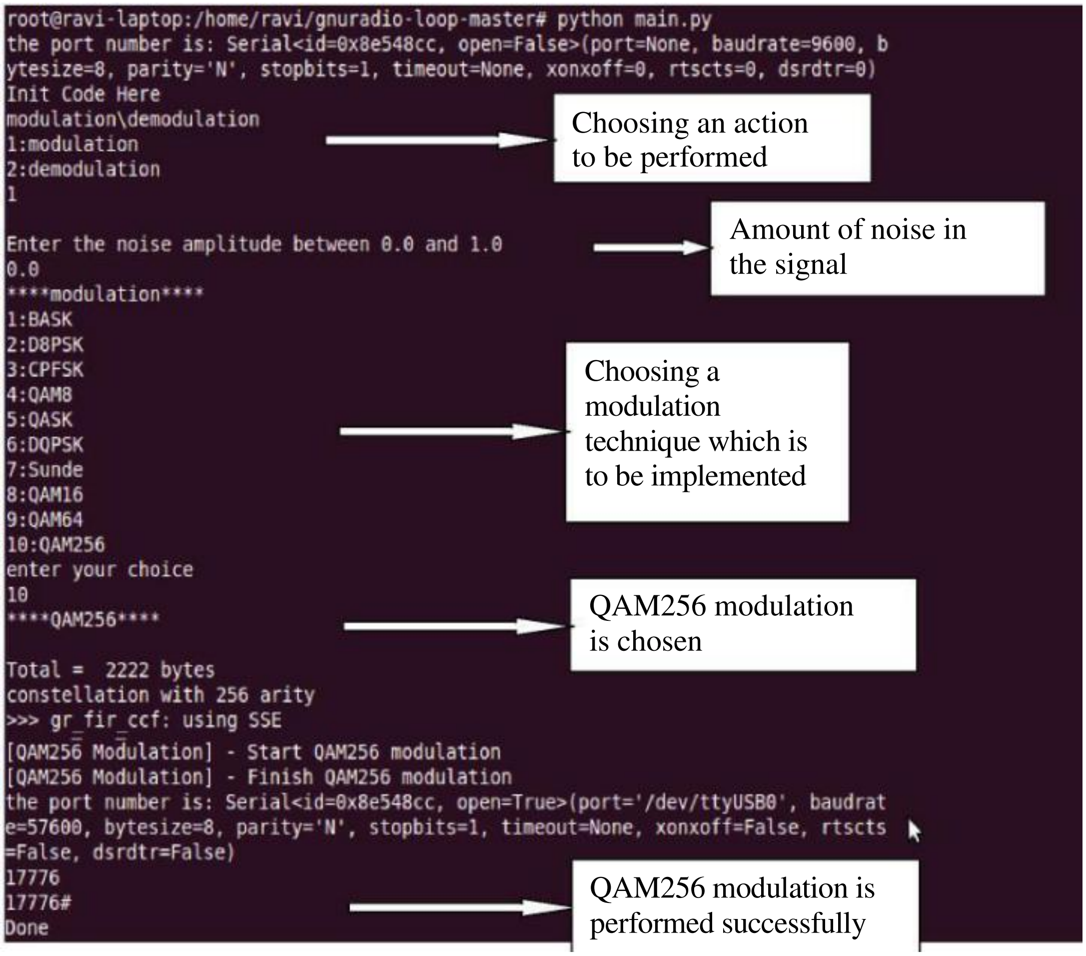
Snapshots showing the Hardware output in choosing the modulation/ demodulation type

The following screenshots shows the operation of the hardware in selecting a particular modulation/demodulation type and performing the chosen modulation successfully.

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

Snapshots showing the process of hardware operation in choosing the QAM256 modulation type

Figure 6.9 shows the hardware process involved in choosing QAM256 modulation. The steps involved are choosing whether the modulation or demodulation action has to be performed, the amount of noise in the signal, choosing a particular modulation technique which is to be implemented and the successful performance of the modulation technique.



```
root@ravi-laptop:/home/ravi/gnuradio-loop-master# python main.py
the port number is: Serial<id=0x8e548cc, open=False>(port=None, baudrate=9600, b
ytesize=8, parity='N', stopbits=1, timeout=None, xonxoff=0, rtscts=0, dsrdtr=0)
Init Code Here
modulation\demodulation
1:modulation
2:demodulation
1

Enter the noise amplitude between 0.0 and 1.0
0.0
****modulation****
1:BASK
2:D8PSK
3:CPFSK
4:QAM8
5:QASK
6:DQPSK
7:Sunde
8:QAM16
9:QAM64
10:QAM256
enter your choice
10
****QAM256****

Total =  2222 bytes
constellation with 256 arity
>>> gr_fir_ccf: using SSE
[QAM256 Modulation] - Start QAM256 modulation
[QAM256 Modulation] - Finish QAM256 modulation
the port number is: Serial<id=0x8e548cc, open=True>(port='/dev/ttyUSB0', baudrat
e=57600, bytesize=8, parity='N', stopbits=1, timeout=None, xonxoff=False, rtscts
=False, dsrdtr=False)
17776
17776#
Done
```

Choosing an action to be performed

Amount of noise in the signal

Choosing a modulation technique which is to be implemented

QAM256 modulation is chosen

QAM256 modulation is performed successfully

CLUSTERING MODEL IMPLEMENTATION

Setting Up Network Connections

The systems should be interconnected with Ethernet cable and the CRO terminals have to be connected to the audio input of the CPU to view the waveforms. The stepwise details to set up Network connections are listed below:

Step 1:

Execution Code set up:

i. Channel parameter files channel_param_BASK.txt, channel_param_CPFSK.txt, channel_param_DQPSK.txt, channel_param_QASK.txt and channel_param_QAM16.txt.

ii. Distance file distance.txt and IP address file ipaddress.txt

iii. Input files packet.txt or in.jpg

Step 2:

Userspace: python

Code Execution

Code Flow

It provides 3 options:

a. Configure Network
b. Send message
c. Exit

Configure Network

i. Select the first option in the main menu to configure the network, configuring the network enables the user to enter the own node or source node and then the user has to enter the total number of nodes in the network.

ii. All the channel parameters and IP addresses are loaded from text files to the respective arrays.

iii. The UDP port has to be opened for communication in asynchronous mode.

Send message

The second option from the main menu is then selected to send a message,

1. According to this option, the user has to enter the destination node to receive the message.
2. Next all the possible paths from source node to the destination node is found, and their corresponding weights depending on the channel parameters of each modulation technique is also noted.
3. Then print all the possible clusters with their corresponding weights. Also print the cluster that has the shortest path to the destination.
4. The best cluster is found depending on the weight.
5. Then modulate the input file and send the modulated file (ModulatedFile.bin) over the best cluster path.
6. The file is formed as packets of size 512 bytes and sent over the UDP.
7. Some details such as sending time, modulation technique and modulated file size are stored in the log file (temp1.txt) which will be present in the present working directory

Receiving message in asynchronous mode

1. The received file (RecModulatedFile.bin) will be stored in the present directory
2. Check if the received node is the destination node, if yes, then receive the message else forward the message to the next node depending on the best cluster path.
3. After the file is received, demodulate the file (DemodulatedFile.jpg or DemodulatedFile.txt) and store it in the present working directory.
4. If the file needs to be forwarded, the demodulated file (DemodulatedFile.jpg or

DemodulatedFile.txt) will be again modulated and the modulated file (ModulatedFile.bin) will be sent to the next node.

5. Some of the details such as sending time, modulation technique and modulated file size, Total Network Throughput and node throughput are stored in the log file (ShortestLogFile.txt) in the present working directory of the destination node.
6. The forwarding node will have the log file (tempRec1.txt) which will contain the sending time, modulation technique and modulated file size.
7. The log files created will be named according to the own node number.

temp<own node number >.txt and tempRec<own node number >.txt

Example: if ownnode number is 2, the log files created will be temp2.txt for the sending message or tempRec2.txt for receiving message.

After every modulation and demodulation, two log files Receiver_Log.txt and Transmitter_Log.txt will be created.

TOPOLOGY DESCRIPTION USING ADAPTIVE CHANNEL MATRIX

Each node within a group of nodes which are placed in the network layer can modulate the data and demodulate the same within the group. The communication technique involves analysing the channel conditions and accordingly choosing that modulation technique to the neighbouring node and combining the routing with shortest path algorithm. This process is continuously followed by all the nodes whenever they receive data. The metrics involved include the network layer metrics, wireless channel based metrics, modulation based aspects etc.

Optimising Network Layer Routing Algorithms

The network layer is optimized with the shortest path routing between the source node and the destination node. This reduces the variability in delay which is also called as Jitter.

Assume, Data rate = B (in Mbps)

Number of switching stages = N

And, if utilization of each slot is given as ‘λ’, then, for a transmission waiting time equal to xT ($0<x<1$), where ‘T’ is the individual stage transmission time, the jitter is calculated as, $BJ = AN[xT+T] + (1-A)NT$

$$BJ = ANxT + ANT + (1-A)NT$$

$$\text{Therefore, Jitter } J = [NT[Ax+1]] \, LB \qquad (6.1)$$

The effect of jitter is more important than trying to minimize it. Once jitter effect is calculated, suitable schemes can be created using buffers wherever it is necessary to enable continuous transfer of packets. In this work, the effect of variation in each parameter of Eqn. 6.1 is observed and plotted in Figure 6.12 for adaptive A, adaptive B and adaptive T. The graph in Figure 6.12 (a) shows the linear increase in jitter as utilization of each slot is increased. In Figure 6.12 (b) jitter reduces as the data rate increases and Figure 6.12 (c) shows that by choosing the shortest path algorithm in the network layer, the individual stage transmission time ‘T’ is reduced which thereby reduces the jitter.

AUTOMATIC MODULATION DETECTION BASED ON FEATURE EXTRACTION

Based on the observations, 20 parameters are extracted from each of the 10 modulated signals. The simulation is done using verilog coding and the algorithm extracts 20 features from a digitally modulated signal. These features can be collected by examining the statistical properties of both the signal and its normalized one. These features contain the necessary information to distinguish the signal between different modulations. The following snapshot as shown in figure 6.14 shows the detection of mod3 signal.

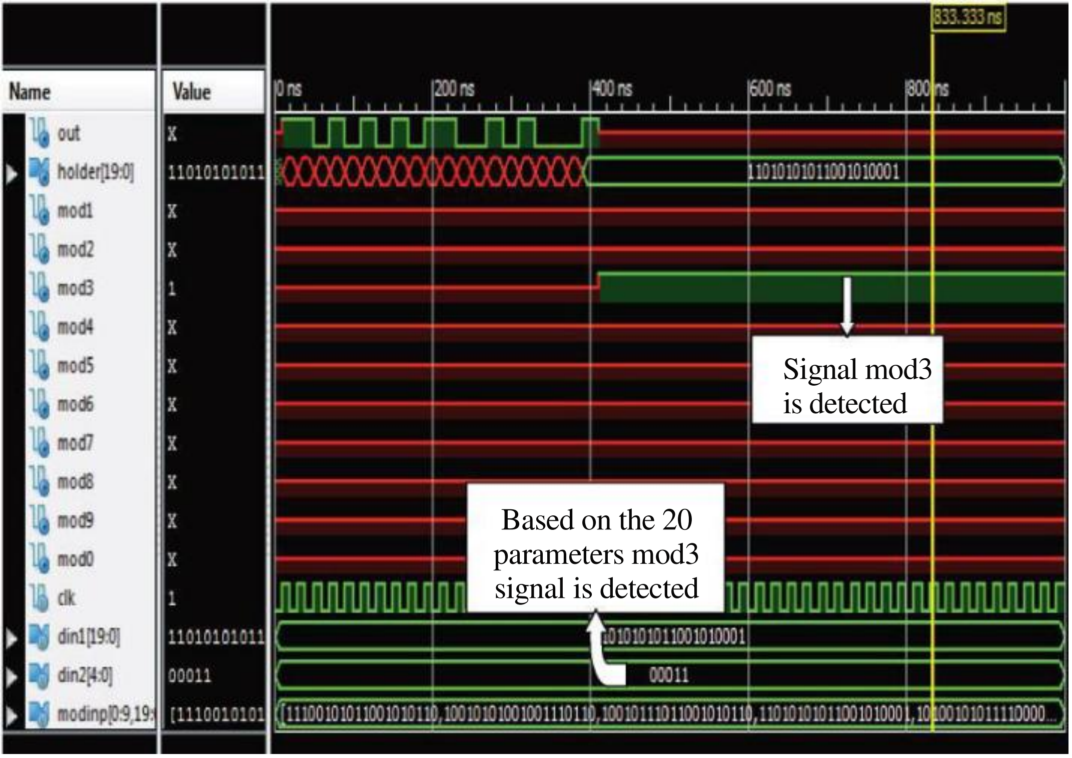


Figure 9 Snapshot showing the automatic detection of mod3 signal

From Figure 9 it can be seen that analysing the 20 parameters, the value obtained for the modulated signal is 11010101011001010001. Based on the number, the corresponding modulated signal mod3 is enabled. Similarly, Figure 6.30 shows the detection of mod5 signal based on the 20 parameters analysed for the given signal. The value obtained after analysing the modulated signal is 10110101011001010101. The corresponding modulated signal enabled is mod5 signal.

MODULATION CLASSIFICATION IN THE PRESENCE OF NOISE

The work deals with using a particular modulation technique which can be very much feasible for transmitting the signal in a low noise location. But, there is every possibility that noise can affect the transmitted modulated signal while transmitting through a communication channel. So, the channel has to be made a reliable information transmitter. This is achieved by implementing a channel coding technique to encode the signal at the transmitter and decode it at the receiver. The efficient and simple coding technique implemented in this work is Hamming encoding Technique which performs error detection and correction.

Error Detection and Correction

Hamming codes fall under the category of FEC which uses a “Block Parity” i.e, sending a block of parity bits to enable correction of single bit errors. Each parity bits are computed on different combination of bits from the data. In this work, the Hamming code introduces 3 redundancy bits to a 4-bit data stream. These redundancy bits are interspersed at bit positions 4, 5 and 7 with the original data bits to form a total of 7 bit encoded data stream.

In this program, the positions of the data bits are represented as1

P1	P2	P3	P4	P5	P6	P7
ü	ü	ü		ü		

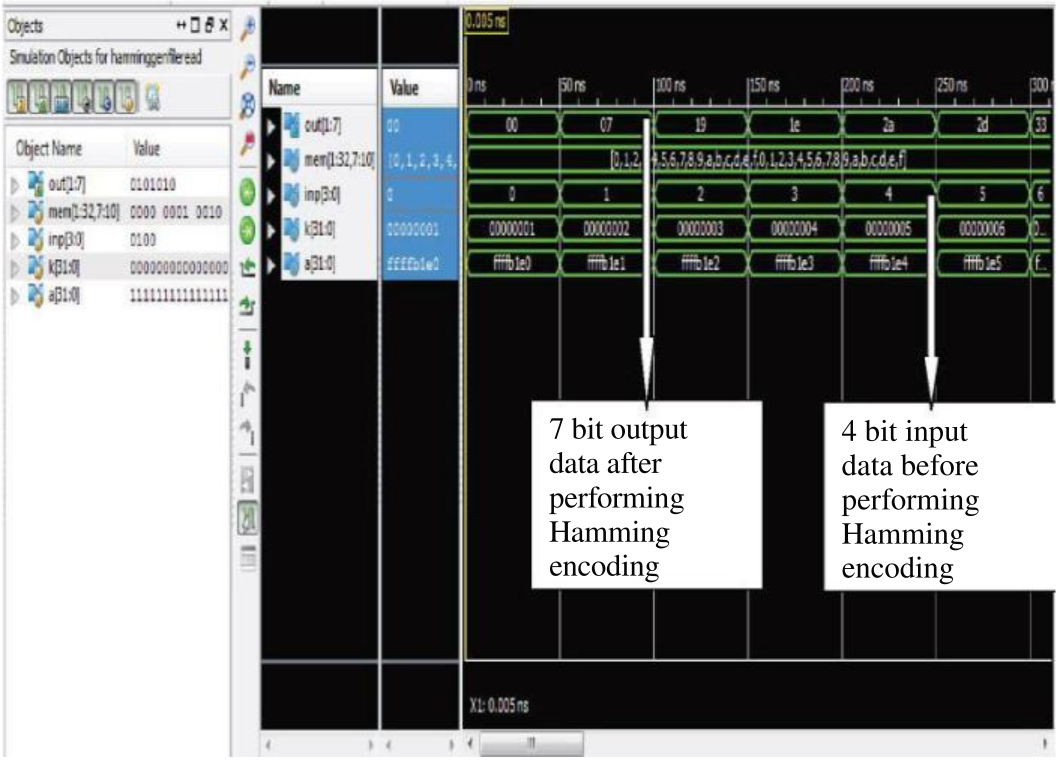
After error correction, if any, the data bits have to be reassembled by removing the redundancy bits.

Captured results for the generation of 7 bit Hamming code (7,4)

The captured verilog output shows the operation of the Hamming encoder. The Encoder converts the 4 bit data bits into its respective 7 bit encoded bit stream. The results are taken for the inputs starting from 0000 (0H corresponding hexadecimal value) to 1111 (FH corresponding hexadecimal value).

Figure 10 Captured result shows the Hamming Encoder inputs for values 0H to 5H

In Figure 6.31, the Hamming encoded 7 bit output data for the 4 bit input data for values starting from 00H to 05H is shown. Figure 6.32 shows, the Hamming encoded 7 bit output data for the 4 bit input data for values starting from 06H to 0BH. Similarly, the Hamming encoded 7 bit output data for the 4 bit input data for values starting from 0BH to



7 bit output data after performing Hamming encoding

4 bit input data before performing Hamming encoding

Consolidating the results obtained for all the 16 input values, the following Table 6.3 is prepared. The table shows the 7 bit Hamming coded values for its corresponding 3 bit input values to the Hamming coder.

Figure 10 Hamming decoder outputs

Figure 10 shows the Hamming decoder outputs 00, 07, 19, 1E, 2A and 2D for the inputs 01, 07, 19, 1E, 28 and 2D. The decoder detects the error in the 1st and 5th inputs and corrects them. Also in Figure 6.35, the inputs to the decoder are 34, 1B, 4C, 52, 55 and 61 and the outputs from the decoder are 34, 1B, 4C, 52, 55 and 61. The detector detects no error for the given inputs. Similarly in Figure 6.36, the inputs to the decoder and the outputs from the decoder are shown. The detector detects no error. All the inputs and outputs from the Hamming decoder are listed below in Table 6.4. The status of the error is also shown in the table.

Table 2 Contents of the input and output of the Hamming decoder

Sl. No.	7 bit input to the Hamming decoder	Hexadecimal value of the input	7 bit output from the Hamming decoder	Hexadecimal value of the output	Status of error
1	0000001	01	0000000	00	Error in position 1
2	0000111	07	0000111	07	No Error
3	0011001	19	0011001	19	No Error
4	0011110	1E	0011110	1E	No Error
5	0101000	28	0101010	2A	Error in position 2
6	0101101	2D	0101101	2D	No Error
7	0110011	33	0110011	33	No Error
8	0110100	34	0110100	34	No Error
9	1001011	4B	1001011	4B	No Error
10	1001100	4C	1001100	4C	No Error
11	1010010	52	1010010	52	No Error
12	1010101	55	1010101	55	No Error
13	1100001	61	1100001	61	No Error
14	1100110	66	1100110	66	No Error
15	1111000	78	1111000	78	No Error
16	1111111	7F	1111111	7F	No Error

CONCLUSION

The main essence of this thesis of performing the automatic modulation identification technique is explained in this chapter by tracking the Pilot sequence appended in the modulated signal. The work is implemented using hardware for 10 modulation types on FL 2440 ARM9 core. Moreover, simulated outputs are obtained to illustrate Automatic modulation detection based on feature extraction. Additionally, the scheduler based implementation is also demonstrated in this chapter to perform the node tracking analysis using the priority allocation for each node. Chapter 6 also deals with network layer routing algorithm to find the shortest path based on channel conditions and clustering of the mobile nodes to enable establishment and safeguarding of the path for continuous transmission of data from the source to the destination. In an adaptive SDR wireless network, the longevity of the nodes in the transmission path is controlled by the power consumed and the bandwidth provided by the link between the nodes to handle the data transfer.

REFERENCES

[1]. V. Brik, S. Banerjee, M. Gruteser, and S. Oh., “Wireless device identification with radiometric signatures”, in MobiCom ‘08: Proceedings of the 14th ACM International Conference on Mobile Computing and Networking, ACM, pp. 116-127, New York, USA, 2008.

[2]. Z. Li, W. Xu, R. Miller and W. Trappe, “Securing wireless systems via lower layer enforcements”, In WiSe '06: Proceedings of the 5th ACM Workshop on Wireless Security, pp. 33-42, ACM Press, 2006.

[3]. Maryam Soltan, Inkwon Hwang, Massoud Pedram, “Modulation-Aware Energy Balancing in Hierarchical Wireless Sensor Networks”, IEEE Transaction, 2008.

[4]. V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava, “Energy-Aware wireless microsensor networks”, IEEE Signal Processing Magazine, Vol. 19, pp. 40–50, March 2002.

[5] M. Soltan, M. Maleki, and M. Pedram, “Lifetime-aware hierarchical wireless sensor network architecture with mobile overlays”, Proceeding of IEEE Radio and Wireless Symposium, pp. 325-328, Jan. 2007.

[6] Z. Yang, Y. Yuan, J. He, and W. Chen, “Adaptive modulation scaling scheme for wireless sensor networks”, IEICE Transactions on Communications, pp. 882-889, 2005.

[7] C. Schurgers, O. Aberthorne, and M. B. Srivastava, “Modulation scaling for energy aware communication systems”, Proceeding, IEEE ISLPED, pp. 96–99, Aug. 2001.

[8] C. H. Liu and H. H. Asada, “A source coding and modulation method for power saving and interference reduction in DSCDMA sensor network systems”, in Proceeding American Control Conference, Vol. 4, pp. 3003–3008, May 2002.

[9] Deqiang Chen, and J. Nicholas Laneman, “Modulation and Demodulation for Cooperative Diversity in Wireless Systems”, IEEE Transactions On Wireless Communications, Vol. 5, No. 7, July 2006.

[10] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, “Cooperative diversity in wireless networks: efficient protocols and outage behavior”, IEEE Transaction Information Theory, Vol. 50, No. 12, pp. 3062–3080, Dec. 2004.

[11] Dong-Sun Kim, Sung-Joon Jang and Tae-Ho Hwang, “A Fully Integrated Sensor SoC with Digital Calibration Hardware and Wireless Transceiver at 2.4 GHz”, SENSORS, ISSN 1424-8220, Vol. 13, pp. 6775-6792, 2013.

[12] Zhang, C., Wei, C., Jiang, H., Wang, Z., “A Baseband Transceiver for Multi-Mode and Multi-Band SoC”, in proceedings of 2012 IEEE 55th International Midwest Symposium on Circuits and Systems, Boise, Idaho, 5–8, pp. 770–773, August 2012.

[13] Xu,W., Yue, Y., Zhou, T., “System Architecture Design of the Internet of Things based on ZigBee”, in proceedings of IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, Bangkok, Thailand, pp. 826–829, May 2012.

[14] Longbi Lin, Ness B. Shroff, and R. Srikant, “Asymptotically Optimal Energy-Aware Routing for Multihop Wireless Networks with Renewable Energy Sources”, IEEE Transactions, 2005.

[15] B. Danev and S. Capkun. Transient-based identification of wireless sensor nodes. In Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks

(IPSN) 2009, 2009.

[16] S. Selvakumar, Dr.S.Ravi, “Adaptive Modulation IN Reconfigurable Platform” Journal of Theoretical and Applied Information Technology. Vol 68, PP 108-114, Oct 2014.

[17] S. Selvakumar, Dr.S.Ravi, “DPSK and QAM Modulation Detection analyzed with BER Estimation” IEEE International Conference on Current Trends in Engineering and Technology, July 2014.