IET Communications

Research Article

IET Journals
The Institution of
Engineering and Technology

Implementation of modulation classifier over software defined radio

ISSN 1751-8628
Received on 16th September 2019
Revised 27th January 2020
Accepted on 5th March 2020
E-First on 22nd April 2020
doi: 10.1049/iet-com.2019.0922
www.ietdl.org

Gaurav Jajoo[1] ✉, *Yogesh Kumar*[1], *Ashok Kumar*[2], *Sandeep Kumar Yadav*[1]
[1]Department of Electrical Engineering, N.H.65, Karwar, Jodhpur, India
[2]Scientific Analysis Group (SAG), DRDO, Delhi, India
✉ *E-mail: jajoo.1@iitj.ac.in*

Abstract: Blind signal modulation recognition is an essential block for designing a cognitive radio. Different algorithms are developed in the literature, but few are given with detailed implementation. This study proposes a software-defined radio based implementation of blind signal modulation recogniser (BSMR) on field-programmable gate array (FPGA), which works without any prior knowledge of the received signal. The algorithm estimates carrier frequency offset, symbol rate, symbol timing offset, and corrects the signal for these offsets to extract constellation points. It uses clustering structure formed by constellation signature in I/Q plane to detect the modulation for different orders of ASK, PSK, and QAM. The proposed algorithm is deployed on FPGA, using LabVIEW, for a reliable and reconfigurable platform. The algorithm is optimised to use minimum hardware resources and facilitate future up-gradation. The system developed by implementing the algorithm on NI-FlexRIO-7975 FPGA module with NI-5791 adapter detects modulation type in real time without any training. Signals for testing are generated using NI-PXIe-5673 (RF transmitter), and BSMR identifies the modulation type in 81.451 ms under additive white Gaussian noise channel.

1 Introduction

Automatic modulation classification (AMC) or blind signal modulation recognition is the identification of the modulation format from the sensed signal with no prior knowledge of any transmitted signal parameters. AMC has numerous military and civil applications. Military application includes spectrum surveillance, electronic warfare, preparing jamming signal, electronic warfare. It is also used in link adaptation by which spectrum is utilised efficiently. Most of the classifiers assume perfect channel knowledge for recognition of modulation format. This assumption is invalid in case of military applications like surveillance, threat analysis, or in electronic warfare. Most of the current work is progressing towards the reduction of hardware, which gives the emergence of software-defined radio (SDR) and cognitive radio (CR). Several techniques are developed in past [1] but few are implemented and validated in real channel scenario [2–4]. AMC gained more attention in CR as it is widely used for a civilian application like spectrum management, link adaptation to overcome from the problem of spectrum under-utilisation. Real-time blind signal modulation recogniser (BSMR) requires a highly reliable, reconfigurable, independent platform, and SDR is an efficient platform for such system architecture [5]. This can be realised through the combination of software with a wide range of capabilities and suitable hardware platform, i.e. field-programmable gate array (FPGA). FPGA is also broadly used for many real-time applications such as system performance measurement, advanced control system design, and data acquisition [6–8].

BSMR recognises the modulation in two stages. First is preprocessing of the received RF signal, in which different parameters like carrier frequency, symbol rate, channel state information (CSI), and symbol timing offset are estimated and corrected for offsets. The second stage includes methods to classify among different modulation formats, which are broadly divided into two classes: likelihood-based (LB) and feature-based (FB).

LB classifiers achieve optimal performance in the sense of correct classification but assume perfect CSI. It is multiple hypotheses testing problem in which likelihood functions for all modulation schemes considered are calculated using known probability density function. They are compared with a threshold value which makes classification task complex and less usable in the blind scenario. Different likelihood ratio tests like average likelihood ratio test (ALRT), generalised likelihood ratio test (GLRT) [9] and combination of ALRT and GLRT, i.e. hybrid likelihood ratio test [10] are developed in literature to solve the problem of unknown parameters, to reduce complexity and biased nature towards nested modulation schemes. As higher order modulations are considered, likelihood function struggles with high computational complexity and not usable for real-time applications.

With the optimal results but higher computational complexity of LB methods, FB methods give near-optimal results, robust, easily implementable, and suitable for real-time applications [11]. Features from the received signal based on signal statistics, wavelet transform [12], cyclostationary [13, 14], multiple cumulants [15], correlation and cyclic cumulants [16] or combination of these are extracted, and further decision tree is employed to classify final modulation scheme. Now a days, different machine learning (ML) techniques like k-nearest neighbour [17], support vector machine [18], artificial neural networks [19], and deep neural network [18] are used as classifiers. These classifiers are computationally efficient and give reliable results. Most of the ML techniques used in literature requires to train the system, which makes them of less use if new modulation schemes are introduced for classification. Recently, unsupervised learning based on clustering and curve fitting methods are also proposed for AMC [20–24].

The present system classifies modulation format based on extracted constellation signature in real-time without any training compared to most of the ML techniques. To extract these points, the system depends on the existing method for CFO estimation and correction and symbol rate estimation. Once CFO and symbol rate are estimated, the developed process removes symbol timing offset from CFO corrected resampled signal for extraction of final constellation points. The number of clusters in the I/Q plane is recognised to estimate modulation order. ASK, PSK, and QAM are separated using linear regression error analysis and a feature based on calculated centroids. Aim of this research is to develop a generalised system, capable of identifying any order of ASK, PSK, and QAM.

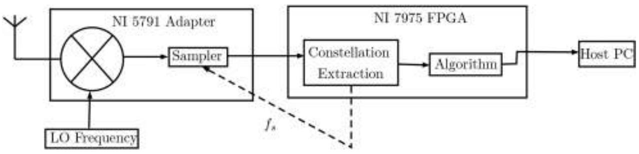

Fig. 1 *System model*

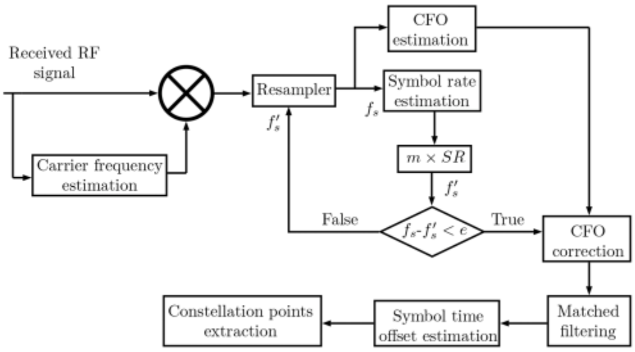

Fig. 2 *Preprocessing and extraction of constellation points*

The commercially available system for blind signal modulation detection is a black box system with no scope of further modification. Many papers with FPGA implementation based on wavelet [25], ANN [26], and conventional features [27, 28] are given in the literature, but only a few have been designed for real-time operation and provides little information about implementation. With the evolution of SDR, implementation of a radio receiver can be easily achieved through LabVIEW coding. This paper includes a detailed description of the implementation of the algorithm on LabVIEW FPGA, so that it can be replicated and further worked upon the improvement over results concerning latency, complexity, and real-time performance. Most of the existing methods consider the specific order of modulation and depends on the offline training of the system. This makes the task complex and system is not usable for other than the considered order of modulation. Unlike these methods, the present system can work for any order of modulation among ASK, PSK, and QAM with a slight modification. Hence, SDR based on FPGA hardware platform is chosen, as they are highly flexible, versatile, reconfigurable and can be modified by making required changes in the software. Different modules of the algorithm are designed using the LabVIEW FPGA module of National Instruments (NI). Virtual instruments (*VIs*) created with LabVIEW FPGA module can be targeted to FPGAs on RIO hardware, which is NI PXIe-7975R (Kintex-7 XC7K410T) FPGA based FlexRIO module in the system used.

Structure of the paper comprises six sections. Section 2 describes the FPGA and FlexRIO software communication architecture. Signal and system model is described in Section 3. Algorithm and its implementation on FPGA are explained in Sections 4 and 5, respectively. Section 6 shows the experimental results and related discussions. Section 7 concludes the paper.

2 LabVIEW FlexRIO software communication architecture

Hardware implementation of the algorithm over FPGA is done using LabVIEW *VIs*. Two *VIs* are used for implementing algorithm on labVIEW FPGA: (i) *Host VI* and (ii) *FPGA VI*. *Host VI* runs on computer and controls the FPGA. It is used to send information between host computer and FPGA target and provides user interface to the system. It interacts with *FPGA VI* and display the results.

Another *VI* is the *FPGA VI* (*target VI*). To transfer small and frequent data between *Host VI* and *FPGA VI*, front panel communication is used. To transfer large data, first in first out (FIFO) is used. FPGA module has three types of FIFO, of which target scoped FIFO is used to transfer the data within the *target VI*, and direct memory access FIFO is used to transfer data from host to target and vice versa. Another FPGA element used for storing large data is *block memory*, which can perform both read and write operation from random address. FPGA module of labVIEW compiles *target VI* to hardware and graphical code is translated to VHDL. Further Xilinx compiler synthesise this code into hardware realisation of labVIEW design. Finally, a bitfile is generated that contains programming instructions that are mapped onto FPGA target. Any change in the algorithm can be done by making the necessary changes on the *target VI* and generating a new bitfile. It can be reconfigured using the host.

3 Signal model and system model

Signals are assumed to be root raised cosine pulse shaped with random roll-off factor at the transmitter side. Signals received are sampled with very high frequency. Carrier frequency offset (CFO), phase offset, symbol timing offset, and additive white Gaussian noise (AWGN) channel are considered. Symbols from the downconverted signal after matched filtering and downsampling signal to symbol rate are represented by $r(n) = [r(1)r(2)r(3)\ldots r(N)]$, where

$$r[n] = A\mathrm{e}^{j(2\pi f_0 nT + \phi_0)}s(n) + w(n)$$

$s(n)$ represents equiprobable transmitted symbols drawn randomly from a considered set of modulation. As noise cannot be controlled in a wireless environment, AWGN noise is generated and added at the transmitter. The signal is transmitted through SMA cable. In AWGN channel with $w(n) \sim N(0, \sigma_w^2)$, if the transmitted symbols are of unit power, SNR can be defined as

$$SNR = \frac{P_{\text{signal}}}{P_{\text{Noise}}} = 10\log_{10}\frac{A^2}{\sigma_w^2}$$

Frequency offset and phase offset are represented by f_0 and ϕ_0, respectively. Symbol time is represented by T. Phase offset is assumed to be constant in slow fading channel and channel gain A remains constant for single signal realisation.

Our system consists of two modules. NI-5791 adapter module downconverts the signal with the frequency provided and digitise the signal at the high rate provided by the user. The algorithm is deployed on an attached FPGA module, i.e. NI-7975 FLEX RIO. It processes the data according to the designed algorithm and presents result on the host. The model of the system used is shown in Fig. 1.

4 Algorithm

4.1 Extraction of constellation points

As the algorithm depends on the extracted constellation signature, the system depends on prior methods for baud rate estimation, CFO estimation and correction. The symbol rate is estimated using squared envelope of complex baseband signal, and CFO is estimated using eighth-order non-linearity of baseband sequence. Symbol rate and CFO are estimated in parallel using [29], which reduces the overall modulation estimation time. Fig. 2 shows the block diagram of the preprocessing of signal and extraction of constellation points. After symbol rate estimation, the signal is resampled to m samples per symbol and compared with previously estimated sampling frequency. The resampled signal is passed for CFO correction only when the calculated difference between the present and previous sampling frequency is less than a set threshold. The CFO corrected signal is passed through the FIR filter for matched filtering.

Matched filtered signal has a timing offset, i.e. from those m samples/symbol, it is not known that which sample corresponds to constellation points. To estimate timing offset from a total of Z samples, the signal is bunched into Z/m number of sets where each bunch has m values. ith value corresponding to each bunch is added (where i ranges from 1 to m), and the outcome is an array of m samples from which the index corresponding to the maximum value is an estimate of symbol timing offset. If l is desired index,

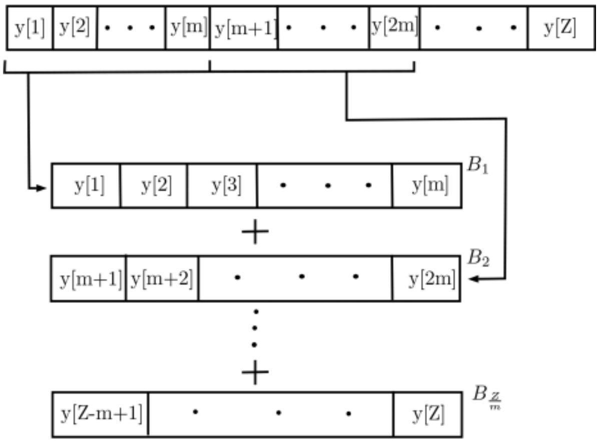

Fig. 3 *Symbol timing offset estimation visualisation*

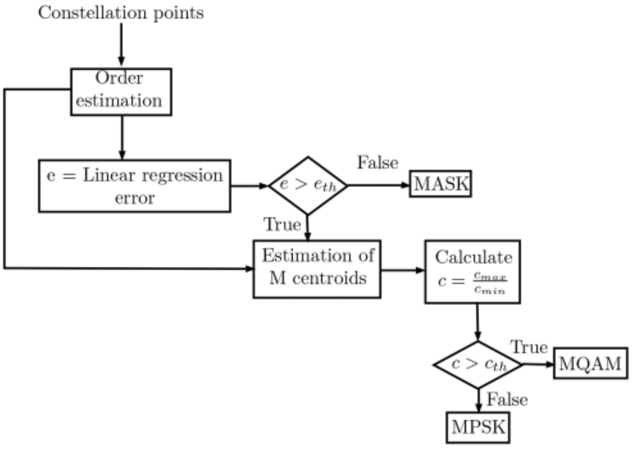
Constellation points
Order estimation
e = Linear regression error
$e > e_{th}$
False
MASK
True
Estimation of M centroids
Calculate $c = \frac{c_{max}}{c_{min}}$
$c > c_{th}$
True
MQAM
False
MPSK

Fig. 4 *Block diagram of algorithm for domain classification among ASK, PSK, and QAM*

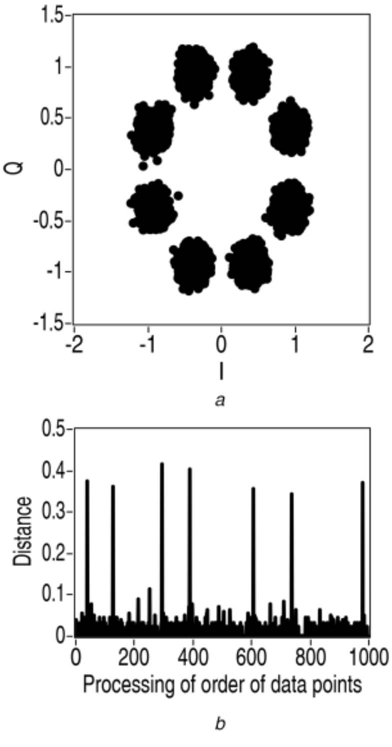

Fig. 5 *Estimation of modulation order*

(a) Extracted constellation points of 8PSK, ***(b)*** Distance versus processing order of data points for (a)

every $(l + rm)$th sample is taken, where $r = 0, 1, 2, \ldots, k$ such that $(l + km)$th data point corresponds to last acquired constellation point.

Fig. 3 shows the method for estimation of symbol timing offset. Phase offset is not addressed in this system as the proposed algorithm is independent to the rotation of constellation. The algorithm counts the number of clusters in I/Q plane, which does not change with rotation.

After preprocessing and acquiring signal constellation represented by $r'[n]$, the developed algorithm shown in Fig. 4 is applied to the data points for detecting the type of modulation

$$r'[n] = A\mathrm{e}^{j\phi_0}s(n) + w[n]$$

4.2 Estimation of order of modulation

Order of modulation is detected by estimating the number of clusters in I/Q plane. Steps described are followed for order estimation.

Input - N constellation points, seed point (first point is taken as seed point in the present case).
Output - Optimum number of clusters, i.e. order of modulation (M).
Default - Distance values (parameter corresponding to each data point) of all data points are initialised with a large value.

Step 1 For total N points, distances between seed point and $(N - 1)$ points are calculated. Seed point distance is saved as a default (large) value, and its order of processing is set to 1.
Step 2 The nearest data point is taken as *present point* for further calculation, and its order of processing and distance value (from the previous point) is saved.
Step 3 Distances between *present point* and all unprocessed points are calculated. If the distance of any unprocessed point from *present point* is less than the distance from any previous processed point, it is replaced by lesser distance else it is not changed.
Step 4 Go to step 2 until all points are processed.

Fig. 5 shows the 8PSK constellation points and its graph between distance and order of processing for extracted data points. Data points within the same cluster (i.e. same symbol) has less value of distances and when a *present point* switches from one cluster to another, its distance increases. As modulation order can only be in the form of 2^K, distances greater than the threshold and nearest to 2^K, (for $K = 1, 2, 3, 4, \ldots$) is chosen as order.

4.3 Modulation domain estimation

Three domains of modulation have been considered viz. ASK, PSK, and QAM. To detect ASK from the pool of ASK, PSK, and QAM, error through line regressed on data points is calculated. This error is less for ASK compared to PSK and QAM, and ASK is identified. If the modulation is not ASK, the following steps are taken for centroid calculation:

Input - L (number of centroids to estimate) seed values.
Output- L centroid values.

Step 1 Calculate distance between the first point and L centroid values. Assign this point to the group to which this point is nearest.
Step 2 Repeat step 1 for all points.
Step 3 Calculate the new centroid values by taking mean of data points assigned in the same group and update L values by calculated mean.
Step 4 Repeat steps 1 to 3 for 10 iterations.

The ratio of maximum and minimum absolute values of centroids is used to differentiate PSK and QAM. This ratio is lesser in PSK compared to QAM, as symbols have the same amplitude.

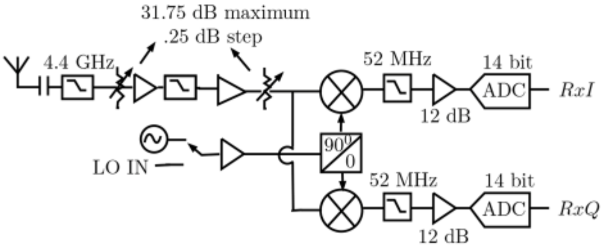

Fig. 6 *Model of 5791 for acquiring data*

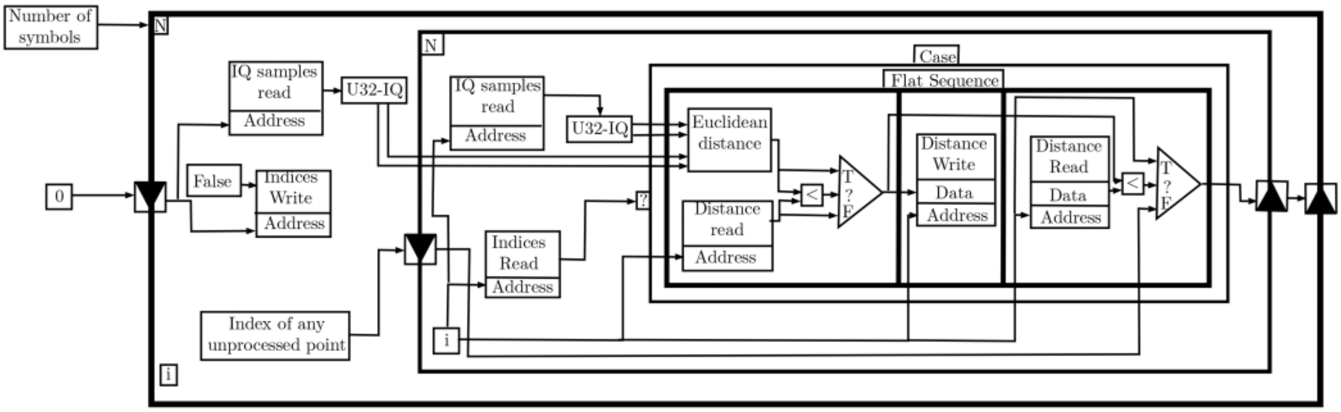

Fig. 7 *LabVIEW implementation for modulation order calculation*

5 FPGA implementation of the AMC algorithm

With the rapid development of SDR in the past few years, radio receivers can be implemented easily and efficiently using LabVIEW block diagram coding. To calculate FFT of the signal and matched filtering, two *Express VIs* specific to LabVIEW are used. All other *VIs* are self-designed. Hence, the algorithm can be easily replicated to any other tool of interest.

5.1 LabVIEW FlexRIO architecture

NI FlexRIO system is used to implement the developed algorithm. It provides flexible, customisable input and output for FPGA. FlexRIO system is composed of adapter module (NI 5791 in the present system) and FPGA (PXIe-7975R consists of Kintex-7 XC7K410T). NI 5791 adapter module defines the physical inputs and outputs of the system and provides continuous frequency coverage from 200 MHz to 4.4 GHz with 100 MHz bandwidth. Second is the NI FlexRIO FPGA Module, which is NI PXIe-7975R FPGA.

5.2 Hardware realisation of AMC algorithm

AMC system starts acquiring the signal once the *detect* signal is asserted. Carrier frequency is estimated through the power spectrum of the signal. Same frequency signal is generated through the adapter module for downconversion. The signal is sampled with user-specified frequency or with a default value of 10 MS/s.

5.2.1 Acquiring the data: Inphase and quadrature phase components are received through *RxI* and *RxQ* pin from 14-bit ADC in $I16$ format shown in Fig. 6. These values are digitised at the desired rate and transferred to another loop after packing in $U32$ format via target scoped FIFO in block memory named *IQSamples*. All the I/Q data retrievals for further processing will be done from *IQSamples* memory by unpacking $U32$ in $I16-I16$ formats and further converting to the fixed-point data type.

5.2.2 Symbol rate estimation, resampling and CFO estimation: The symbol rate is estimated using 8192 point FFT of the absolute square of the complex envelope. For n point FFT, latency is $2n$. Hence, 24576 $(2\times 8192+8192)$ points are acquired and given to FFT block. After discarding first 16384 data outputs, further 8192 valid points are continuously compared to find the index corresponding to the maximum value. The frequency corresponding to the desired index is the estimated symbol rate. After the first estimation, new data is continuously acquired with a rate of eight times the estimated symbol rate. A local variable is used to change the sampling rate in real time. The new symbol rate is estimated from the further acquired waveform. Eight times of new estimated symbol rate, i.e. a new sampling rate is compared with the previous sampling rate. The signal is forwarded to CFO correction block only when new estimation is desirably close to the previous estimation of symbol rate to achieve accuracy.

CFO is estimated simultaneously with the symbol rate. Same 24576 points used for symbol rate estimation from *IQSamples* are passed through eighth-order non-linearity using a high throughput complex multiply. These functions are similar to the numeric functions, but support automatic pipelining, and achieves higher throughput rate inside a single-cycle timed loop. The frequency corresponding to the location of the maximum value is used to estimate and correct frequency offset.

From a single loop running for 24576 iterations, CFO and symbol rate are estimated. CFO corrected signal is passed to the FIR filter for matched filtering. FIR filter coefficients are set to root raised cosine pulse shaping with eight samples per symbol. Filtered data is used for the estimation of modulation order and domain. Implementation for order and centroid estimation are given in further sections.

5.2.3 Modulation order calculation: Fig. 7 shows the block diagram of the LabVIEW implementation of the algorithm on FPGA. Three block-memories are used for order calculation. First is *IQ samples* which contains IQ points. Second is *Indices* which is initialised with *True* at all addresses and takes count of all unprocessed and processed points. Third is *Distance*, which is initialised with large value (in present case 8). Two *For* loops are used. Outer *For* loop run N times and inner *For* run N times for each iteration of outer *For* loop. The outer shift register is initialised with zero, and the inner shift register is given an index of the unprocessed point from *Indices* (in the first iteration, 0 is marked processed and fed to inner shift register). Whichever point is processed, *Indices* at that address is given *False* and case structure is used for considering only unprocessed points.

Case structure contains flat sequence, which runs sequentially from left to right. In the first block of flat sequence, for every iteration of outer *For* loop, the Euclidean distance between *present*

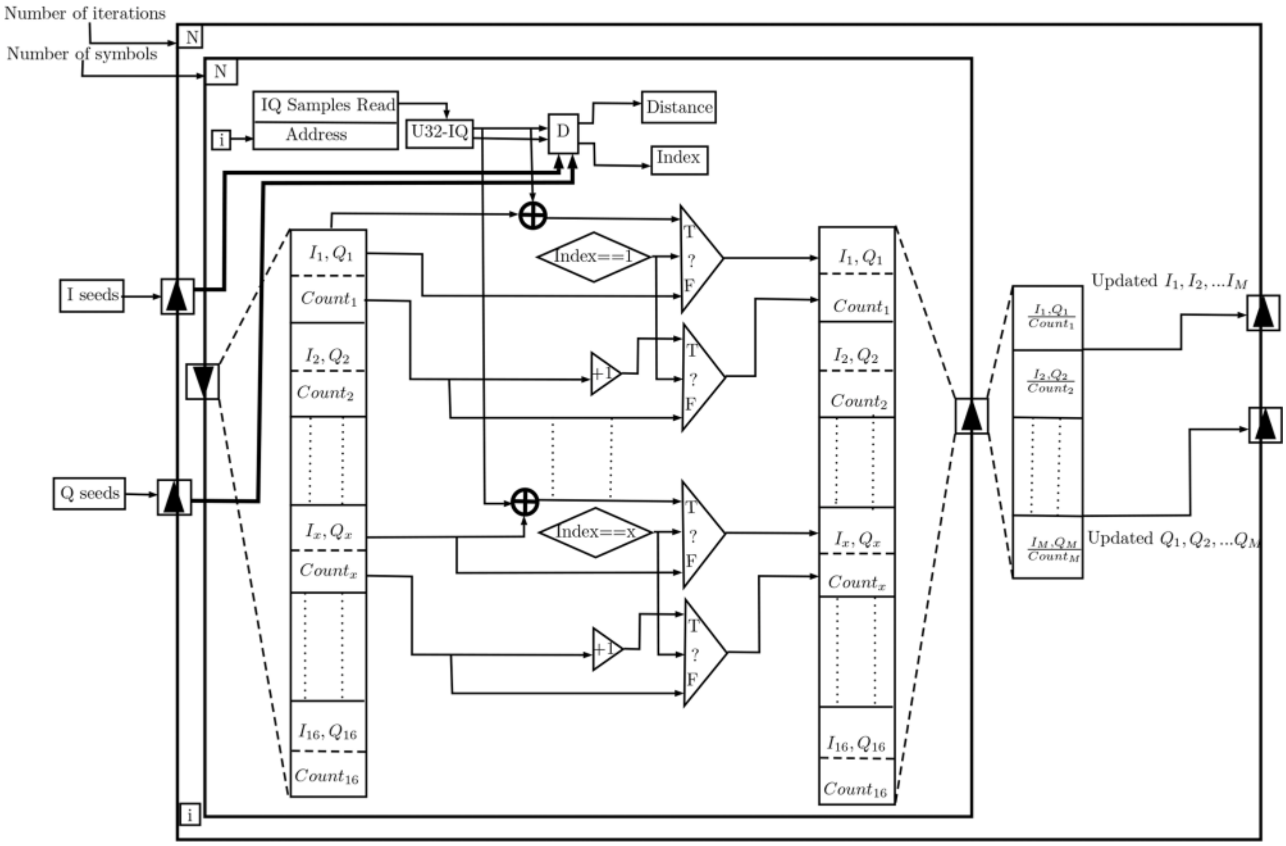

Fig. 8 *LabVIEW implementation for centroid calculation*

point and all unprocessed points (addresses wherever indices are true) is calculated and compared with its previous distance. This distance is replaced in the second block of flat sequence if found less. The third block always takes count of index value, which is nearest to the present point. The final output of inner *For* loop is the index value of the unprocessed point, which is nearest to *present point*. That particular point is read from *IQ samples* and False is written in *Indices* to mark it processed. This is done for all symbols. The final output is the *Distance* array, which is shown in Fig. 5*b*.

5.2.4 Centroid calculation: Implementation on FPGA for centroid calculation is shown in Fig. 8. This implementation is explained if the highest modulation order is 16. Same can be replicated for any other order modulation. Two *For* loops are defined of which outer *For* loop runs for ten iterations considering the optimal centroid estimation with less processing time and efficient resource utilisation over hardware. Inner *For* loop runs *N* (number of symbols) times for each iteration of outer *For* loop. Initial seed values for *I* and *Q* are taken as ideal PSK constellation points. One limitation of FPGA is that it does not allocate dynamic memory at run time. So to define the initial centroid points for *I* and *Q*, an array of 30 elements is used. This array is initialised with first 16 values (0–15) as ideal 16-PSK points, 15 to 23 elements with 8-PSK points, 24 to 27 values with QPSK, and last two with ideal points of BPSK. Depending on the estimated order, array subset is passed through *I seeds* and *Q seeds* to shift registers. An array of 32 elements with *I* and *Q* interleaved (initialised with 0) is defined for summation of points belonging to the same cluster. An array of 16 elements (initialised with zero) is defined to store the number of points falling in the same cluster. Actual implementation over the hardware has two arrays implemented separately, but these arrays are shown together for a clear understanding of the implementation.

Euclidean distances between each IQ point and seed values are calculated by *D* given in Fig. 8. Index of the nearest centroid is assigned to each data point based on the distance values. Present I and Q values are added to the previous values with the same index, and the respective count is incremented by 1. The process is repeated for all the data points. Out of 16 values, first *M* values are updated where *M* is modulation order. The ratio of maximum and minimum absolute centroid value is calculated. In the case of PSK, this value will be near to 1 while in QAM, it is greater than PSK. Ratio is compared with threshold ($c_{\text{th}} = 1.3$) and PSK is differentiated from QAM.

If the highest modulation order considered is changed to 64, the only modification required is to initialise the array of centroid values with points considering 32QAM and 64QAM, and passing the seeds as centroid subset accordingly. Also, the inner shift register will be initialised with a length of 128 values for interleaved *IQ* centroids and count with a length of 64.

6 Results and discussions

In this section, the performance of the proposed system is evaluated with respect to SNR and compared with some of the existing systems. RF signals for testing are generated using NI PXIe-5673 and received using NI 5791 shown in Figs. 9*a* and *b*, respectively. As the noise in the wireless environment cannot be controlled, to evaluate the classification accuracy, transmitter and receiver are connected through SMA cable. Signal with known SNR is transmitted through SMA cable, and accuracy of the algorithm is evaluated. Thresholds used at different parts of the algorithm are analysed. The algorithm is also validated by transmitting an RF signal through a wireless channel and classifying the signal with the receiver placed at a distance of 6.1 mt shown in Fig. 9*c*. In this arrangement, low noise amplifiers (ZX60-P103LN+) are used at receiver for amplifying the RF signal.

6.1 Experimental outcomes

To address the classification accuracy over various SNR values, different magnitudes of Gaussian noise were added in transmitting signals. Classification accuracy is calculated from 1000 signals, each containing 1000 symbols. 2ASK/BPSK and QPSK achieve an accuracy of more than 90% above 8.5 and 11 dB SNR, respectively, as shown in Fig. 10*a*. 8PSK and 8QAM require a

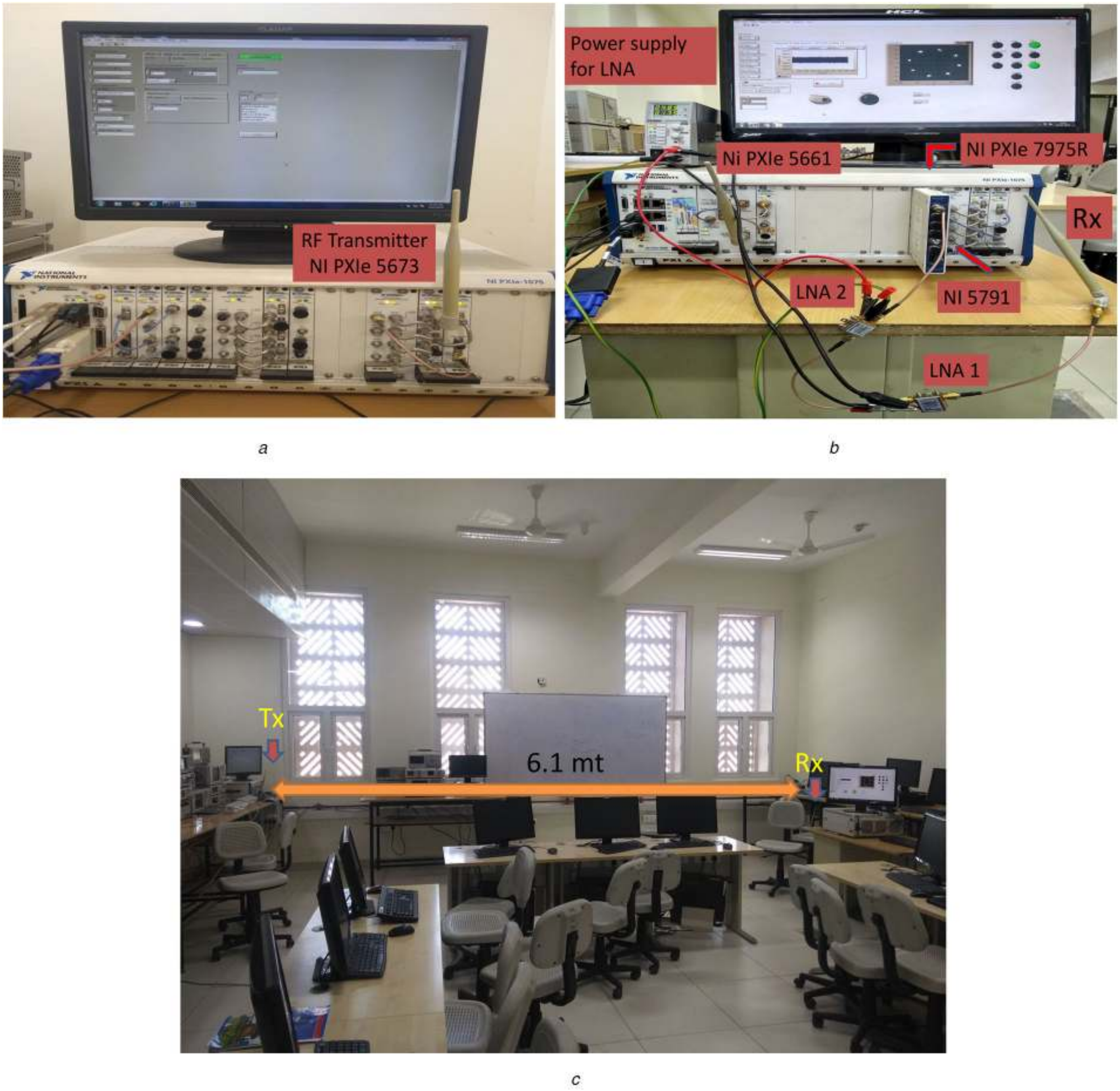

Fig. 9 *Transmitter and receiver for RF signal*
(a) Setup for RF signal generation, ***(b)*** Receiver setup for RF signal reception, ***(c)*** Setup of transmitter and receiver in the laboratory for validation of algorithm

signal of around 15.2 dB SNR and above to achieve 90% accuracy while 16QAM requires 16.8 dB for the same accuracy shown in Fig. 10*b*. Number of peaks between 12 and 24 is considered for order 16 as for peaks less than 12; order will fall into 8QAM or 8PSK while for greater than 24, the order will be near to 32 and will fall into modulation with order 32.

Many algorithms are developed which correctly identifies the basic modulations like PSK, FSK, ASK, QAM, and so on but cannot be modified further. Hence, this research tries to propose an algorithm to detect the higher modulation scheme, and system developed can be modified and updated for future requirements. Like for order 32 and 64, output array for comparison with number of peaks has been changed such that it contains values 32 and 64. Rest no change is required, which makes the developed algorithm flexible. Results for 32QAM, 64QAM, 16PSK, and 4ASK are shown in Fig. 10*c*.

Thresholds for linear regression error ($e_{\rm th}$) and centroid ratio ($c_{\rm th}$) is defined to achieve maximum accuracy for considered SNR. Values of regression error for all modulations are plotted in Fig. 11*a*. It can be observed from the figure that, regression error for all ASK (with order 2, 4, 8) has value less than $e_{\rm th}$, i.e. 0.3. This threshold differentiates ASK from PSK and QAM for all SNR. Also, as SNR increases, regression error for ASK tend towards 0, while for PSK and QAM, it is around 0.5. Similarly to separate PSK and QAM, the ratio of maximum and minimum absolute centroid value ($c_{\max}/c_{\min}$) has been used, which is shown in Fig. 11*b*. Fig. 12 shows the overall accuracy of QPSK, 8PSK, 8QAM, and 16QAM for threshold with different percentage of the maximum peak value. It can be observed that 60% of maximum peak covers the maximum area, which is set as $c_{\rm th}$. For higher SNR, as PSK constellation has a circular shape, $c_{\max}/c_{\min}$ attains a value of 1 while for QAM with order 8 and above, this value is above 1.5. All these analyses are done for 100 signals each with 1000 symbols.

Hardware execution provides higher performance than most processor-based software programs, but the performance decreases if the resource usage is not proper. Table 1 shows that 55.4% of total resources have been used for implementing the algorithm. Therefore, more algorithms can be developed and can be executed in parallel with the designed algorithm. Optimisation of the algorithm is done by partitioning the application code into smaller processes.

Real-time performance for different modules has been found and shown in Fig. 13. Once the detect button is pressed, the program goes into a case structure starting with *start* case in which signal is acquired. Fig. 13 shows that *algorithm* block throws the output in 56.638 ms and preprocessing block for extracting constellation points is taking 24.813 ms. Overall, the system is taking 81.451 ms for computing the modulation scheme after acquiring the signal.

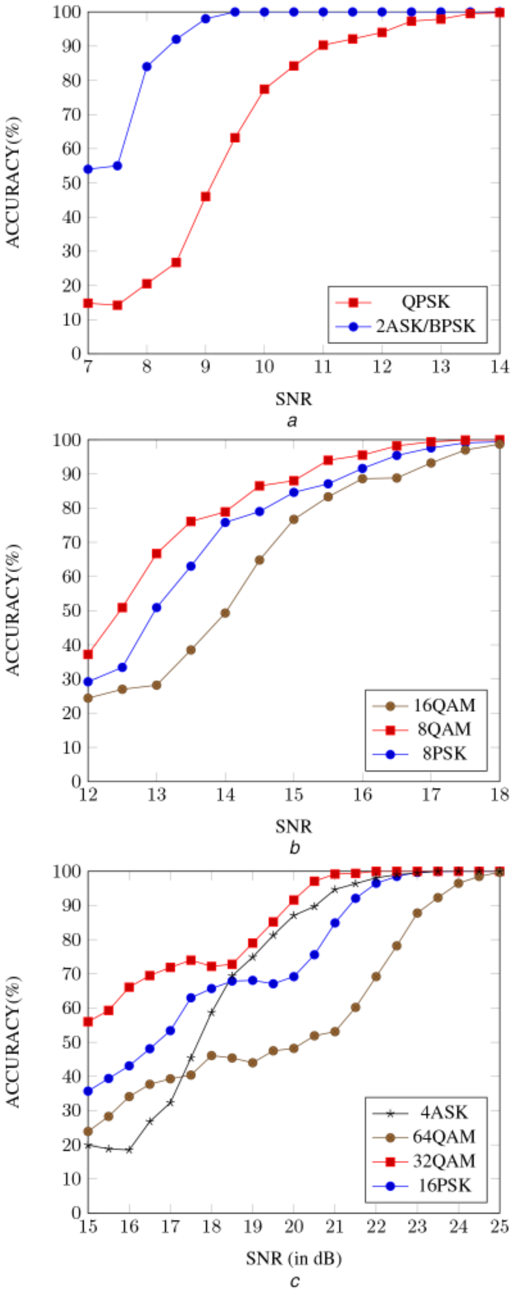

Fig. 10 *Accuracy versus SNR for different modulation schemes*
(a) Accuracy versus SNR for 2ASK/BPSK and QPSK modulation schemes, ***(b)*** Accuracy versus SNR for 8PSK, 8QAM and 16 QAM, ***(c)*** Accuracy versus SNR for 16PSK, 32QAM, 64QAM and 4ASK modulation schemes

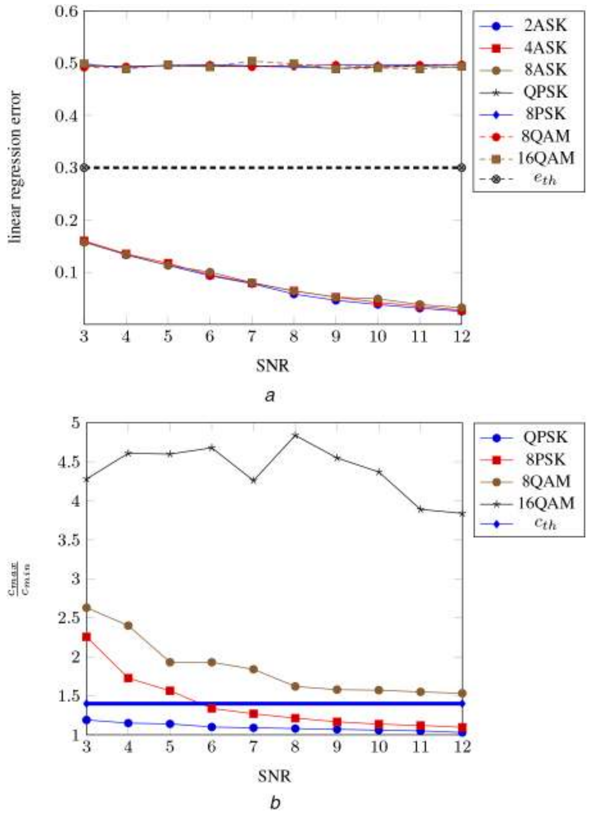

Fig. 11 *Threshold calculation*
(a) Linear regression error for different modulation schemes versus SNR ***(b)*** Value of ratio of maximum and minimum absolute centroid value versus SNR

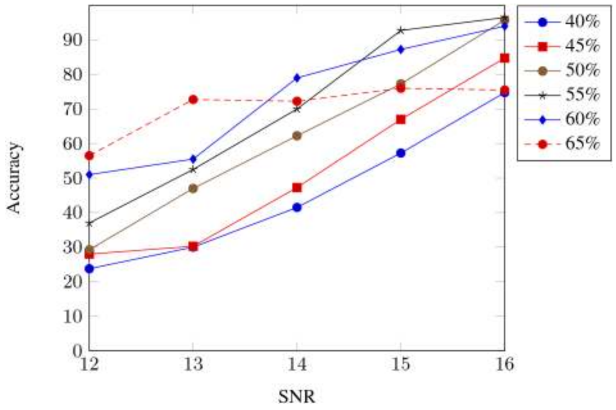

Fig. 12 *Overall accuracy of correct classification for threshold with different percentage of peak distance value. These distances are obtained while estimating modulation order. This graph shows that for threshold of 60% of maximum value, accuracy is overall higher*

Table 1 Hardware resource utilisation

Device utilisation	Used	Total	Percentage usage
total slices	35236	63550	55.4
slice utilisation	101918	508400	20
slice LUTs	85050	254200	33.5
block RAM	260	795	32.7
DSP48s	187	1540	12.1

6.2 Comparison with other classifiers

There are not a lot of real-time classifiers implemented on FPGA to directly compare the results. Most of the implementations either differ in the choice of the modulation scheme of interest or detailed implementation is not explained, so it is not possible to reproduce the results. So for comparison, results at a range of SNR covered in this paper are directly compared for the common modulation schemes. Fig. 14 shows the comparison between [30] and the present method for the BPSK modulation scheme. This choice of comparison is taken based on results obtained from practical signals with SNR. From the figure, it can be seen that the present method dominates [30] from an SNR of 7.5 dB and above. Nie *et al.* [31] report ‘Shortest distance based on C_{63}’ feature can achieve a maximum of 92% accuracy and ‘Shortest distance based on C_{63}/C_{21}^{3}’ achieves 100% accuracy above 25 dB for QPSK, 16PSK, and 64QAM. From the results obtained, our method outperforms all these modulation schemes. Keshk *et al.* [26] used DFT and MFCC features with SVM and ANN as classifiers for different orders of PSK. At 20 dB SNR, for 8PSK, the accuracy of [26] is always less than 100% (in OFDM over AWGN with signal features) while our method achieves accuracy of 100% with testing over 1000 signals. Similarly, accuracy for 16PSK with the same scenario achieves greater accuracy with VLR, RPROP classifiers.

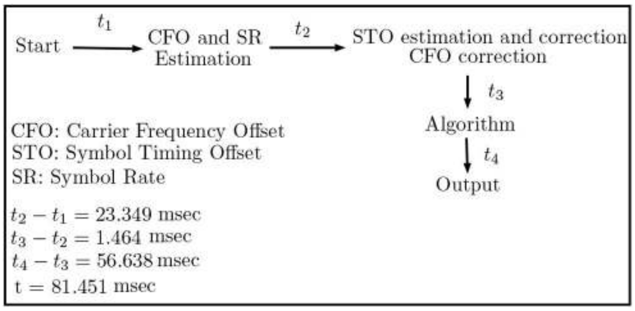
Start
t_1
CFO and SR Estimation
t_2
STO estimation and correction CFO correction
t_3
Algorithm
t_4
Output
CFO: Carrier Frequency Offset
STO: Symbol Timing Offset
SR: Symbol Rate
$t_2 - t_1 = 23.349$ msec
$t_3 - t_2 = 1.464$ msec
$t_4 - t_3 = 56.638$ msec
t = 81.451 msec

Fig. 13 *Time utilisation for different modules*

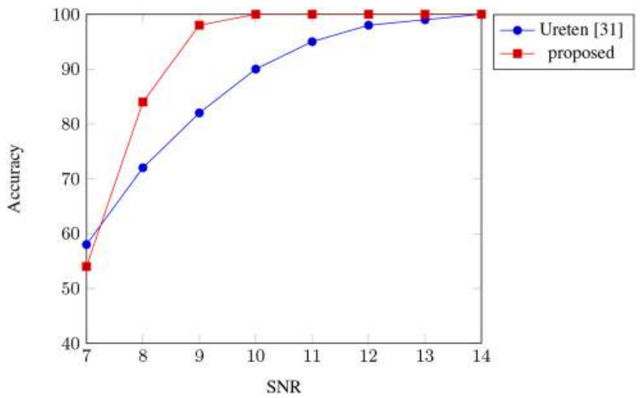

Fig. 14 *Comparison between [30] and proposed algorithm for BPSK modulation scheme*

Table 2 Comparison of proposed system accuracy with [32]

SNR	BPSK		QPSK		16QAM	
	[32]	Proposed	[32]	Proposed	[32]	Proposed
10 dB	85	100	85	77.4	75	—
15 dB	90	100	90	99.8	80	76.7
20 dB	94	100	94	99.7	90	99.6
25 dB	96	100	94	99.8	90	99.5

Table 2 compares the proposed method with [32] for BPSK, QPSK, and 16QAM modulation schemes.

Accuracy of classifiers based on Moment, Cumulant, GP-KNN, and EM-ML changes with carrier phase offset as given in [33] while the developed algorithm is independent to such phase offsets. By correcting the phase offset, information bit can also be extracted as the system is extracting the symbols and modulation scheme. The main limitation lies in the FFT size, which is 8192 (maximum size available), due to which resolution of spectrum in case of CFO and symbol rate estimation decreases.

7 Conclusion

In this paper, an algorithm is proposed and implemented on FPGA, which identifies the modulation format in the blind environment. It estimates different parameters of the blind signal and can be modified for any order of ASK, PSK, and QAM. The algorithm developed is independent to phase offsets, unlike most of the methods in the literature. It is optimised to use minimal resources on board and detects the modulation scheme in 81.451 ms. Detailed implementation of the system is explained. This work can be further extended to incorporate more modulations along with the present modulation schemes.

8 Acknowledgments

This project is funded by the Defense Research and Development Organization under project number S/DRDO/SKY/20150004.

9 References

[1] Dobre, O.A., Abdi, A., Bar-Ness, Y., *et al.*: ‘Survey of automatic modulation classification techniques: classical approaches and new trends’, *IET Commun.*, 2007, **1**, (2), pp. 137–156

[2] Eldemerdash, Y.A., Dobre, O.A., Üreten, O., *et al.*: ‘Identification of cellular networks for intelligent radio measurements’, *IEEE Trans. Instrum. Meas.*, 2017, **66**, (8), pp. 2204–2211

[3] Majhi, S., Gupta, R., Xiang, W., *et al.*: ‘Hierarchical hypothesis and feature-based blind modulation classification for linearly modulated signals’, *IEEE Trans. Veh. Technol.*, 2017, **66**, (12), pp. 11057–11069

[4] Majhi, S., Ho, T.S.: ‘Blind symbol-rate estimation and test bed implementation of linearly modulated signals’, *IEEE Trans. Veh. Technol.*, 2014, **64**, (3), pp. 954–963

[5] Tuttlebee, W.H.: ‘Software-defined radio: facets of a developing technology’, *IEEE Pers. Commun.*, 1999, **6**, (2), pp. 38–44

[6] Monmasson, E., Idkhajine, L., Cirstea, M.N., *et al.*: ‘Fpgas in industrial control applications’, *IEEE Trans. Ind. Inf.*, 2011, **7**, (2), pp. 224–243

[7] Marignetti, F., Minutillo, M., Perna, A., *et al.*: ‘Assessment of fuel cell performance under different air stoichiometries and fuel composition’, *IEEE Trans. Ind. Electron.*, 2011, **58**, (6), pp. 2420–2426

[8] Zheng, G., Zheng, W., Zhuang, G., *et al.*: ‘The design of global interlock system in j-text’, *IEEE Trans. Plasma Sci.*, 2014, **42**, (6), pp. 1775–1779

[9] Panagiotou, P., Anastasopoulos, A., Polydoros, A.: ‘Likelihood ratio tests for modulation classification’. MILCOM 2000 21st Century Military Communications Conf. Proc., Los Angeles, CA, USA, 2000, vol. 2, pp. 670–674

[10] Hameed, F., Dobre, O.A., Popescu, D.C.: ‘On the likelihood-based approach to modulation classification’, *IEEE Trans. Wirel. Commun.*, 2009, **8**, (12), pp. 5884–5892

[11] Dobre, O.A., Abdi, A., Bar-Ness, Y., *et al.*: ‘Blind modulation classification: a concept whose time has come’. 2005 IEEE/Sarnoff Symp. on Advances in Wired and Wireless Communication, Princeton, NJ, USA, 2005, pp. 223–228

[12] Hong, L., Ho, K.: ‘Identification of digital modulation types using the wavelet transform’. MILCOM 1999 IEEE Military Communications Conf. Proc. 1999, Atlantic City, NJ, USA, 1999, vol. 1, pp. 427–431

[13] Satija, U., Manikandan, M., Ramkumar, B.: ‘Performance study of cyclostationary based digital modulation classification schemes’. 2014 9th Int. Conf. on Industrial and Information Systems (ICIIS), Gwalior, India, 2014, pp. 1–5

[14] Rodriguez, P.M., Fernandez, Z., Torrego, R., *et al.*: ‘Low-complexity cyclostationary-based modulation classifying algorithm’, *AEU-Int. J. Electron. Commun.*, 2017, **74**, pp. 176–182

[15] Huang, S., Yao, Y., Wei, Z., *et al.*: ‘Automatic modulation classification of overlapped sources using multiple cumulants’, *IEEE Trans. Veh. Technol.*, 2017, **66**, (7), pp. 6089–6101

[16] Gupta, R., Majhi, S., Dobre, O.A.: ‘Design and implementation of a tree-based blind modulation classification algorithm for multiple-antenna systems’, *IEEE Trans. Instrum. Meas.*, 2019, **68**, pp. 3020–3031

[17] Aslam, M.W., Zhu, Z., Nandi, A.K.: ‘Automatic modulation classification using combination of genetic programming and knn’, *IEEE Trans. Wirel. Commun.*, 2012, **11**, (8), pp. 2742–2750

[18] Ali, A., Yangyu, F., Liu, S.: ‘Automatic modulation classification of digital modulation signals with stacked autoencoders’, *Digit. Signal Process.*, 2017, **71**, pp. 108–116

[19] Popoola, J.J., van Olst, R.: ‘The performance evaluation of a spectrum sensing implementation using an automatic modulation classification detection method with a universal software radio peripheral’, *Expert Syst. Appl.*, 2013, **40**, (6), pp. 2165–2173

[20] Jajoo, G., Yadav, Y.K., Yadav, S.: ‘Blind signal digital modulation classification through k-medoids clustering’. 2018 IEEE Int. Conf. on Advanced Networks and Telecommunications Systems (ANTS), Indore, India, 2018, pp. 1–5

[21] Jajoo, G., Kumar, Y., Yadav, S.K.: ‘Blind signal psk/qam recognition using clustering analysis of constellation signature in flat fading channel’, *IEEE Commun. Lett.*, 2019, **23**, (10), pp. 1853–1856

[22] Yadav, Y.K., Jajoo, G., Yadav, S.K.: ‘Modulation scheme detection of blind signal using constellation graphical representation’. 2017 Int. Conf. on Computer, Communications and Electronics (Comptelix), Jaipur, India, 2017, pp. 231–235

[23] Jajoo, G., Kumar, Y., Yadav, S.K., *et al.*: ‘Blind signal modulation recognition through clustering analysis of constellation signature’, *Expert Syst. Appl.*, 2017, **90**, pp. 13–22

[24] Tian, J., Pei, Y., Huang, Y.D., *et al.*: ‘Modulation-constrained clustering approach to blind modulation classification for mimo systems’, *IEEE Trans. Cogn. Commun. Netw.*, 2018, **4**, (4), pp. 894–907

[25] Digdarsini, D., Kumar, M., Khot, G., *et al.*: ‘Fpga implementation of automatic modulation recognition system for advanced satcom system’. 2014 Int. Conf. on Signal Processing and Integrated Networks (SPIN), Noida, India, 2014, pp. 464–469

[26] Keshk, M.E.H.M., El-Naby, M.A., Al-Makhlasawy, R.M., *et al.*: ‘Automatic modulation recognition in wireless multi-carrier wireless systems with cepstral features’, *Wirel. Pers. Commun.*, 2015, **81**, (3), pp. 1243–1288

[27] Iglesias, V., Grajal, J., Royer, P., *et al.*: ‘Real-time low-complexity automatic modulation classifier for pulsed radar signals’, *IEEE Trans. Aerosp. Electron. Syst.*, 2015, **51**, (1), pp. 108–126

[28] Cutno, P., Cheng, C.H.: ‘A software-defined radio based automatic modulation classifier’. Wireless Telecommunications Symp. (WTS) 2017, Chicago, IL, USA, 2017, pp. 1–6

[29] Sethi, A., Ray, B.: ‘Blind carrier/timing recovery and detection of modulation scheme’ (Google Patents, 2013. uS Patent 8,605,830

[30] Ureten, O., Yensen, T.N.: 'System and method for classifying signal modulations' (Google Patents, 2017. uS Patent 9,584,251

[31] Nie, Y., Shen, X., Huang, S., *et al.*: 'Automatic modulation classification based multiple cumulants and quasi-newton method for mimo system'. Wireless Communications and Networking Conf. (WCNC) 2017 IEEE, San Francisco, CA, USA, 2017, pp. 1–5

[32] Azza, M.A., Moussati, A.E., Moussaoui, O.: 'Implementation of an automatic modulation recognition system on a software defined radio platform'. 2018 Int. Symp. on Advanced Electrical and Communication Technologies (ISAECT), Rabat, Morocco, 2018, pp. 1–4

[33] Zhu, Z., Nandi, A.K.: '*Automatic modulation classification: principles, algorithms and applications*' (John Wiley & Sons, UK, 2014)