A new hybrid approach for symbol rate detection using simulated annealing and cyclic autocorrelation

James E, Whitney II and Richard A. Carr

***Abstract*—Cyclic Autocorrelation has been proposed as a technique for symbol rate detection in cognitive radios. However, cyclic autocorrelation can be inefficient due to the need to perform an autocorrelation of the received signal at different cyclic frequencies followed by an exhaustive search over the entire search space in order to find the cyclic frequency that produces the maximum autocorrelation. In this paper the simulated annealing algorithm is used as a preprocessor to the cyclic autocorrelation algorithm in order to estimate the symbol rate of M-Ary phase shift keyed (MPSK) signals. The Simulated annealing algorithm is used to traverse the search space in order to find the optimum cyclic frequency without performing and exhaustive search.**

***Keywords*—Cognitive Radios, Cyclic Autocorrelation, Simulated Annealing, Software-defined Radio, Symbol Rate detection.**

I. Introduction

COMMUNICATION systems are constantly evolving to increase functionality and efficient use of limited spectrum. New sophisticated communication protocols are being developed to accomplish these goal, however, the radios themselves also need to become smarter. Traditionally communication radios have been designed to resolve a signal of a specific modulation at a specific symbol rate. This means that if it becomes necessary for a radio to operate on a different signal that uses a different modulation scheme and operates at a different data rate, then a new radio would need to be designed. This can be costly in terms of both time and money.

Software-defined radios (SDRs) aim to solve this problem by replacing as much of the radios hardware components with equivalent software modules. The software modules can be programmed to handle multiple waveforms, with various data rates. SDR capabilities are also bolstered by the fact that they can be implemented on field programmable gate arrays (FPGAs) that are reconfigurable in the field, so SDRs can be updated as needed. This makes for a cost effective and robust radio that can be updated remotely.

While a SDR is able to handle various signals, it is still required that the appropriate parameters, i.e., the data rate, modulation type, be set before it is able to accurately resolve a given waveform. Some radios are equipped with algorithms and sensors that allow the radio to detect or estimate these parameters, thus allowing the radio to operate autonomously. These SDRs are referred to as cognitive radios. The applications for cognitive radios are many, ranging from satellite to satellite communications for deep space applications, to intelligent radios for military soldiers in the field. The cognition of a cognitive radio is facilitated by sensors and algorithms that are able sense the environment and determine the necessary parameters to allow the radio to configure itself accordingly. The symbol rate is one of the main parameters of a signal that needs to be resolved in order to accurately receive and demodulate a signal.

II. Current technologies

Many devices today such as smart phones and tablets contain multiple radios that are designed to resolve different types of signals. For example, the typical smart phone may have over ten different radios included in it. Those may include:

- Cellular radio for Global System for mobile (GSM) or code division multiple access (CDMA).
- Bluetooth Radio
- Wi-Fi a/b/g/n
- Radio for Global positioning systems (GPS)
- Radio frequency identification (RFID)
- Long-term evolution (LTE)
- Evolution data optimized (EVDO)
- Infrared radio
- Near field communication (NFC)

Many of these radios have to be configured on these devices that have limited real estate on the chips and circuit boards. Furthermore, having devices with numerous radios can be very inefficient in terms of power management.

This work was supported in part by the National Aeronautics and Space Administration (NASA).

J. E. Whitney, II, is with Morgan State University in the Department of Electrical and Computer Engineering, Baltimore MD 21251 USA (phone: 443-885-3067; e-mail: james.whitney@morgan.edu

R. A. Carr is graduate student at Morgan State University, Baltimore MD 21215 USA (e-mail: ricar1@morgan.edu).

A software-defined radio running on an FPGA or and digital signal processor (DSP) can replace some or all of the physical layers for these radios, thus leading to more efficient use of the real estate of the device. Having a single FPGA may be more power efficient as well. Table I illustrates some more advantages and disadvantages of software-defined and traditional radios.

Table I Comparison of the advantages and disadvantages of traditional and conventional radio.

	Traditional Radio	Software Defined Radio
Advantages	Established Techniques	Potential less power intensive. Smaller Footprint Reconfigurable Upgrades are software dependent Efficient use of spectrum
Disadvantages	Fixed Hardware Non reconfigurable Upgrades may require new hardware	Security Software bugs Requires human-in-the-loop

Table I shows that the main disadvantages of software-defined radios are security and software bugs. Since the radio is based in software, much like many other computer systems the integrity of the radio can be compromised if it is not properly secured. Software based systems are also susceptible to software bugs which can lead affects simple malfunctions to total failure.

III. Parameter Estimation for SDR

In order for a SDR to properly resolve a signal certain parameters such as the signals modulation type, symbol rate and signal-to-noise ratio (SNR) has to be known apriori, and so the radio operator has to manually enter these parameters into the radio.

Therefore, in order to have the radio operate without a human-in-the-loop, mechanisms had to be added on the front end of the radio. These mechanisms are able to detect or estimate these parameters. These types of radios are called cognitive radios. Fig. 1 and Fig. 2 are the respective block diagrams of a software-defined radio and a cognitive radio.

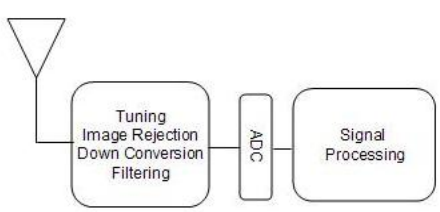


Fig.1 Block diagram of a Software-defined Radio.

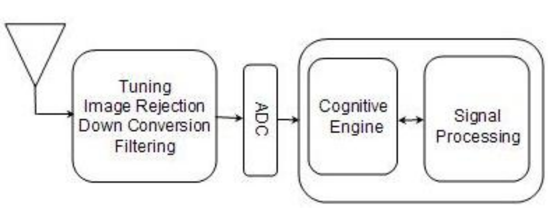


Fig.2 Block diagram of a cognitive radio.

Of all the parameters that the cognitive radio has to either estimate or detect the symbol rate is very important mainly because the symbol rate can be used in the calculation of the other parameters such as the modulation type and the SNR. The symbol rate of a signal can be detected one of two ways [1], the transmitter can send this information on a separate channel or the receiver can estimate the symbol rate, given that the receiver has full knowledge of all the possible symbol rates that the signal can have.

The first option may be impractical in a system where bandwidth is limited. The second option implies that the cognitive radio receiver must blindly estimate the incoming signals parameters with little or no apriori information about the signal.

There are a number of symbol rate estimation algorithms in existence that are capable of blindly estimating symbol rate. These include methods such as cyclic autocorrelation [2], rate estimation via wavelets [3], [4] Fourier transforms [5]. The wavelet technique described in [4] takes advantage of Haar mother wavelet and its ability to detect transient changes in the signal when a symbol change occurs. Whenever a symbol change occurs the wavelet method produces a spike, therefore when this method is applied to a signal with multiple symbol changes a series of peaks are produced and these peaks should display some periodicity. A Fourier transform is then applied to determine the frequency, i.e., the estimated symbol rate. This method implied that the signal must be at baseband. The inverse fast Fourier transform (IFFT) method described in [5] tries to determine the oversampling factor employed by the pulse-shaping filter employed by the transmitter. The algorithm starts by taking a periodogram of the received MPSK signal and to attain a signal G(F). The IFFT is then applied to G(F), resulting in a signal g(n). The oversampling factor is determined by counting the number of samples at which the first null of g(n) occurs. The symbol rate estimate is then determined by finding the quotient of the system sample rate and the estimated oversampling factor. This method requires very large FFTs in order to make the nulls more distinguishable.

For this work, the cyclic correlation algorithm was employed to estimate symbol rate. The cyclic correlation algorithms less complex compared to the wavelet and IFFT methods and lends itself to practical implementation.

IV. The Cyclic Autocorrelation Algorithm

The cyclic correlation algorithm takes advantage of the cyclostationary nature of linearly modulated signals [2] which include M-Ary phase shift keyed (MPSK) signals. The autocorrelation of M-Ary signals repeat with a period that is equal to the symbol rate of the signal [6]. The governing equation is shown in (1).

$$R(\alpha,\tau)=\frac{1}{N}\sum_{n}^{N-1}r(n)\cdot\overline{r(n+\tau)}\cdot e^{-j2\pi\alpha n} \qquad (1)$$

where,

α – is referred to as the cyclic frequency
τ – is the lag
N – is the total number of samples
r – is the received signal
$\bar{r}$ – is the complex conjugate of the received signal

The algorithm performs an autocorrelation at different cyclic frequencies, in search of the optimum cyclic frequency, where the autocorrelation value is maximized. The signal is shifted in the time domain based on τ and the kernel $e^{-j2\pi\alpha n}$ shifts the autocorrelation in the frequency domain [7]. The estimated symbol rate of the given signal is calculated by applying (2)

$$\hat{T}_s = F_s \cdot \alpha_{opt} \qquad (2)$$

where,
$\hat{T}_s$ – is the estimated symbol rate
F_s – is the sample rate
α_{opt} – is the optimum cyclic frequency

The algorithm does an exhaustive search over all possible cyclic frequencies in order to find the optimum frequency and depending on the resolution of the frequencies the process could become very inefficient.

V. Simulated Annealing

The simulated annealing (SA) algorithm is an optimization technique that is based on the annealing process in metallurgy [8]. The algorithm is able to find the global maximum of an objective function by randomly permuting one or more variables of the objective function. The simulated annealing algorithm has the ability to avoid becoming stuck at local extrema and so given enough time the algorithm will find the global extreme of the objective function.

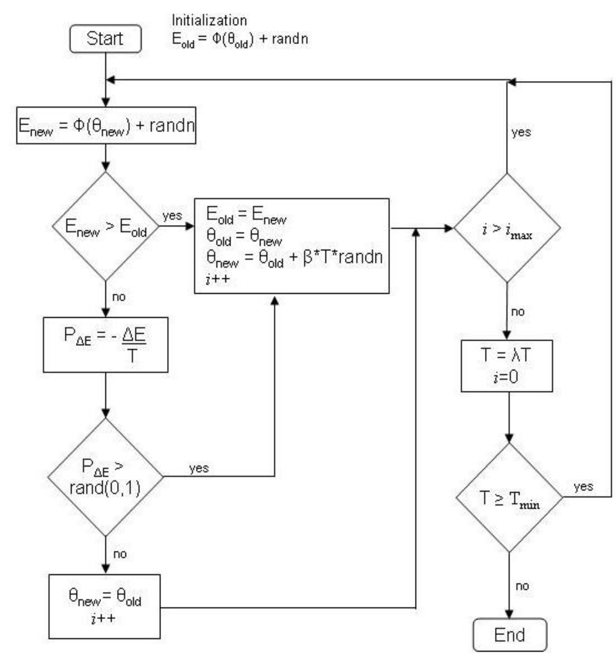
In metallurgy when a metal is heated to a high temperature the atoms within the metal become energized and the randomly and manically move around within the metal. If this temperature is held constant for long enough some equilibrium will be reached. The temperature is then lowered and once again the atoms will randomly move around until some equilibrium is reached for this lower temperature. Since the temperature is lower, the energy levels of the atoms are also lowered and so they do not move around as manically. The temperature is once a gain lowered and the process is it repeated with the temperature being slowly decreased until it reaches some minimum, i.e., freezing temperature. At the minimum temperature the atoms within the metal are at a minimum energy state and have assumed a crystalline structure. The result of the crystalline structure is a more ductile metal. This entire process is referred to as annealing.

Simulated Annealing is analogous in that the process starts at a high temperature. The temperature in simulated annealing is referred to as the control parameter. When the control parameter is high, the algorithm will randomly select one or a combination of free parameters randomly and with wide variation. The free parameters are used to evaluate the objective. The control parameter is slowly decreased to some minimum value. As the control parameter decreases the random selection of the free parameter will not vary as widely. By the time the algorithm arrives at the minimum temperature the it should converge upon a set of parameters that cause the objective function to result in a global extrema. The simulated annealing algorithm is able to avoid getting stuck in a local minimum/maximum by accepting an incorrect result occasionally [9]. The algorithm will accept an incorrect answer based on the Boltzmann probability (3).

$$E = \begin{cases} P_{\Delta E} = -\left(E_{new} - E_{old}\right)/k_b T > U(0,1), & accept \\ otherwise, & reject\ \ incorrect\ \ result \end{cases} \qquad (3)$$

Where, $P_{\Delta E}$ is Boltzmann's probability, and k_b is Boltzmann's constant. The probability is evaluated by differentiating between the previous energy state and the newest energy state divided by the control parameter. If $P_{\Delta E}$ is less than a uniformly random number between 0 and 1 the incorrect solution is rejected. In simulated annealing k_b is usually set to one, therefore the probability is influenced greatly by the control parameter T.

So for example if the algorithm was searching for a global minimum and it was trapped at a local minimum, the algorithm may randomly choose a set of parameters that lead to a higher energy estimate, and that estimate may be excepted as the next state. This essentially allows the algorithm to escape from the local minimum. Fig. 3 shows a flowchart of the general simulated annealing algorithm.



Where,

- θ – is a set of free parameter that are needed by the objective function.
- $E(\theta)$ – is the result of evaluating the objective function

with a given.

- T – the control parameter, i.e., the temperature.
- Φ – the objective function.
- ΔE – is the difference between the new and old evaluations of the objective function.
- $P_{\Delta E}$ – referred to as Boltzmann's probability
- β – scaling parameter
- λ – a number between 0 and 1

VI. Simulated Annealing with Cyclic Correlation

Using the cyclic correlation algorithm for symbol rate detections, requires performing an autocorrelation of the recived signal at different cyclic frequencies and then performing a search to identify which cyclic frequency results in the maximum correlation coefficient. The exhaustive search for the optimum cyclic frequency can be inefficient especially if the resolution of the cyclic frequencies and the correlation lags are small. The cyclic correlation algorithm requires to input parameters, namely the cyclic frequency α, and the lags τ. The simulated annealing algorithm is used to randomly select combinations of these two variables, from a search space and submit them to the cyclic correlation algorithm. This means that the cyclic correlation objective function will not have to be executed for all combinations of α, and τ within the search space. Also due to the randomness of the parameter selection process the resolution of the two parameters are allowed to vary. By using SA an near optimum solution is guaranteed, i.e., the correct set of parameters will be selected from the search space. The result is algorithms that can autonomously detected the symbol rate of a received signal given that the symbol rate lies within the selected search space, and no other information.

Fig. 4 depicts a flowchart of the modified algorithm. The main difference between Fig. 3 and Fig. 4 is that the modified algorithm has to take into account the restrictions associated with the cyclic correlation algorithm.

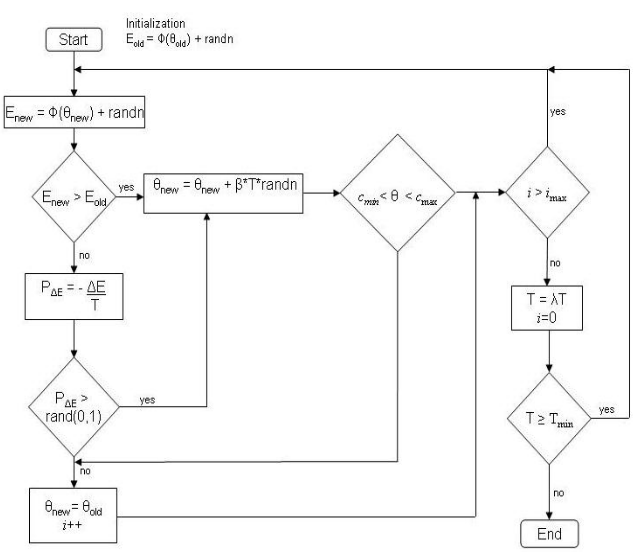


Fig.4 Flow chart of the simulated annealing with cyclic

The two-dimensional search space for the algorithm is defined a all possible combinations of α and τ. In order to determine the range of α's, consider (1) where α is defined by (4)

$$\alpha = R/F_s \tag{4}$$

Where R is a candidate symbol rate and Fs is the sampling rate used to sample the received signal. based on the relationship in (4) it can be seen that the lower limit of α > 0 since, there can be no negative frequencies, nor can the symbol rate be a value of zero. The upper limit is bound by the sampling theorem which states that Fs > 2R, so the upper limit of α is 0.5. Therefore the cyclic frequency is bounded as shown in (5).

$$0 < \alpha < 0.5 \tag{5}$$

The range of lag is determined by the fact that the cyclic correlation algorithm needs to autocorrelate at least one symbol. Therefore, it is necessary to calculate the number of samples per symbol, based on the highest and lowest possible symbol rates of the received signal. The number of samples per symbol is defined by (6).

$$samples\ per\ symbol = F_s/R \tag{6}$$

Therefore (7) defines the number of necessary lags.

$$Fs/R_{\max} \le \tau \le Fs/R_{\min} \tag{7}$$

These bounds are represented by c_{min} and c_{max} in the flowchart in Fig. 4. Essentially, whenever a θ_{new} is evaluated it has to be checked against the bounds before it is accepted for the next iteration of the algorithm. Otherwise the new value is discarded and the old values are retained on the next iteration.

VII. Simulation Results

An experiment was used designed to test the simulated annealing with cyclic correlation algorithm. The algorithm was tested with BPSK signals of differing symbol rates ranging from 300 kBd – 1500 MBd. The simulated annealing algorithm was allowed to simultaneously select combinations of α and τ.

Table II Shows the result of the hybrid algorithm versus the true data

Symbol Rate Bits/s	True Cyclic Frequency	Estimated Cyclic Frequency	Estimated Symbol rate
300000	0.1	0.10	300348.26
345000	0.115	0.12	345064.54
405000	0.135	0.13	404228.15
525000	0.175	0.17	524556.86
630000	0.21	0.21	630151.12
780000	0.26	0.26	780153.38
855000	0.285	0.28	854489.49
945000	0.315	0.31	944970.13
1060000	0.355	0.35	1059586.12
1200000	0.4	0.40	1200143.69
1320000	0.44	0.44	1319528.89
1485000	0.495	0.49	1484680.24

Table II shows the results of the test by comparing the true parameters for the cyclic frequency, the lag and the estimated symbol rate. It can be shown from Table II was that the algorithm was able to successfully determine the cyclic frequency for all of the respective symbol rates to within one percent of error. These results show that the algorithm is capable of blindly determining the symbol rate of MPSK symbol.

VIII. Conclusion

The cyclic correlation algorithm with simulated annealing was able to estimate the symbol rate to within 0.1% of the true rate. The simulated annealing optimization algorithm was able to randomly vary the two input parameters, i.e., lag and cyclic frequency, of the cyclic correlation algorithm. This allows for less executions of the cyclic correlation algorithm, while still producing an accurate result.

Acknowledgment

The authors would like to thank the collaborative team from NASA Goddard Space Flight Center, including Dr. Harry Shaw and Mr. Eric Harris. We would also like to thank the COMSARE and CETACT teams from Morgan State University for their assistance throughout the duration of this project.

References

[1] Wymeersch, H. Moeneclaey, M., "ML-based rate detection for multi-rate receivers", *International Conference on Communication*, Vol. 3, 2005 pp 2092-2096.

[2] Mazet, L. Loubaton. Cyclic Correlation based symbol rate estimation. Thirty Third Asilomar Conference on Signal, Systems and Computers, Vol. 2, 1999, pp. 1008-1012.

[3] Yin, W., Wang K., A new method to symbol rate estimation of MPSK Signals. *Congress on Image and Signal processing*, Vol. 5, 2008, pp 394-398.

[4] A. Kubankaova, D. Kuanek, "Methods of digital modulation recognition and their testing," in Proc. 14th WSEAS Int. Conf. on Communications, Corfu Island, 2010, pp. 234-237.

[5] Flohberger, M., Kogler W., Symbol rate estimation with Inverse Fourier Transforms, *International Workshop on Satellite and Space Communications*, 2006, pp 110-113.

[6] Ou, Yang Yi-Ming Wang, Dan Ye, "A new spectrum sensing method for cognitve radio." In Proc. 10th WSEAS Int. Conf. on multimedia systems and signal processing, , 2010, pp. 35-38.

[7] A. Mazalek, et al. "A simple classifier of PSK and FSK signals using characteristic cyclic spectrum," in Proc. 13th WSEAS Int. Conf. on Mathematical and Computational methods in science and engineering, Sicily, 2011, pp. 435-439.

[8] Bertimas, D; Tsitsiklis, J., "Simulated Annealing," *Statistical Science*, Vol. 8, no.1, pp. 10-15, 1993.

[9] O. Valenzuela, M. Pasadas, "Using simulated annealing for knot placement for cubic spline approximation," in Proc. Int. Conf. on Mathematical models for Engineering Science, Port De La Cruz, 2010, pp. 148-154.

James E. Whitney earned his PhD. degree in electrical engineering from Marquette University in Milwaukee, Wisconsin, in 1995. He earned his M.S. degree in electrical engineering in 1992 from The Catholic University of America, Washington, D.C., and his B.S degree in electrical engineering from the University of Maryland, College Park, Maryland, in 1985.

He is currently at Morgan State University, Baltimore, Maryland where he is an Associate Professor in the Electrical and Computer Engineering Department and is also Director of the signals and sensors (SIGSENS) laboratory. He has worked for, or collaborated with scientists and engineers at the National Aeronautics and Space Administration (NASA), Naval Undersea Warfare Center (NUWC), Newport Rhode Island, Johns Hopkins University and Applied Physics Laboratory (JHU-APL), and Norsk Rikskringkasting (NRK) in Oslo, Norway. He has served, and continues to serve as a consultant to several universities and commercial clients.

Dr. Whitney is a member the Institute of Electrical and Electronic Engineers (IEEE), Audio Engineering Society (AES), and American Society of Engineering Educators (ASEE). His current and past research is in the areas of signal processing including: high-performance embedded computing via DSP and FPGA's, autonomous algorithms, acoustics, optimization, and detection/estimation.

Richard A. Carr was born in Spanish Town, Jamaica in 1984. This author graduated from Woodlawn high School in 2002 and earned a Bachelor's of Science, and Masters of Engineering in electrical engineering, from Morgan State University in Baltimore Maryland respectively in 2006 and 2009.

He is currently pursuing a Doctor of engineering degree at Morgan State University on a part-time basis. He is also employed part-time at the Johns Hopkins University Applied Physics Lab (JHUAPL) in Laurel Maryland as a Software Engineer. His research interests include digital signal processing and its practical applications using field programmable gate arrays (FPGAs), microprocessors and digital signal processors (DSPs).