Deep Learning for An Anti-Jamming CPM Receiver

Y. Qiu, C. Brown and L. Li
Communications Research Centre Canada
Innovation, Science and Economic Development, Government of Canada
Ottawa, Canada
{evanyifeng.qiu, colin.brown, li.li2}@canada.ca

***Abstract*— A novel continuous phase modulation (CPM) receiver model is proposed in this paper that employs deep learning (DL) techniques to improve the signal recovery and synchronization performance under heavy jamming conditions for frequency hopping (FH) based waveforms. A hybrid deep neural network is used to implement the DL in the receiver model. The simulation results show that the proposed receiver with DL boosting is robust under tone jamming which is a worst case scenario for a CPM receiver. The model achieves 3 – 5dB improvement under single-tone jamming, in terms of bit-error-rate (BER), and 2dB improvement under multi-tone jamming, compared with a receiver without DL.**

Keywords— Receiver, deep learning, continuous phase modulation, synchronization, frequency hopping, anti-jamming, acquisition

I. INTRODUCTION

Continuous phase modulation (CPM) attains high power and bandwidth efficiency due in part to the constant envelope nature of the modulation, which allows efficient use of the transmitter power amplifiers. CPM is thus particularly suitable for tactical communications operating in the VHF and low UHF narrow-band channels, where communications range is a key requirement.

Widespread deployment of CPM, in modern radios, has typically been limited by the complexity of implementing the maximum likelihood (ML) method on demodulation and receiver synchronization. In general, applying ML demodulation techniques to CPM involves considerable computational burden [1]. Fortunately, techniques exist to reduce the detector complexity without compromising the performance [2].

For receiver synchronization, a number of methods exist to estimate the carrier frequency, phase and symbol timing parameters either individually [3] or jointly [4-5]. A further level of complexity for the receiver design comes from the tactical environments where the radios are subject to jamming. In this case, taking into account that frequency hopping (FH) is utilized to limit the performance degradation, the receiver must first employ an FH acquisition function, which often involves an initial coarse acquisition of the received pseudo-noise encoded signal, and then a refined acquisition by a tracking loop to bring the system in-lock on the hopping sequence, before demodulation of the CPM waveform [6][7].

The current approaches for CPM synchronization [4-5][10] achieve satisfactory receiver performance in the additive white Gaussian noise (AWGN) channel without jamming. The receiver performance is, however, degraded by jamming signals, in particular under tone jamming conditions, as verified by our experiments presented in Section IV. To ameliorate the problem, a new approach is required to suppress or mitigate the impact of jamming and to sustain the receiver performance.

Deep learning (DL) has shown promising results in a wide range of applications, including, for example, computer vision, natural language processing, and speech recognition. Recently, the application of DL has been extended to communication systems, from signal processing in physical layer [11-12] to application optimization for mobile networking [13]. Many reports have shown the strength of DL in performance improvement in complex communication scenarios that are difficult to describe with tractable mathematical models [11]. In this work, the DL capability is employed in the proposed new anti-jamming CPM receiver model.

The contribution of this work is two-fold:

- A CPM receiver model is established and evaluated which consists of a FH sequence acquisition function and the synchronization function. Compared with previous work [4-7, 10] that investigated either receiver FH sequence acquisition or synchronization, this model provides more complete end-to-end receiver evaluation results.
- The receiver model is further enhanced by a novel DL assisted anti-jamming capability. The new model demonstrates robustness and improved performance under severe tone jamming signals.

The rest of this paper is organized as follows. In section II, the CPM receiver model is discussed, highlighting the various receiver stages needed to receive and process a frequency hopped CPM waveform. This is followed in section III with the description of the proposed DL-assisted receiver model for enhanced performance. Section IV presents the simulation results of the new receiver to a narrow-band frequency hopping system under the AWGN channel and different noise jamming conditions. Finally, section V provides concluding remarks and discusses future work.

This work is supported by Defence Research and Development Canada and Department of National Defence Canada.

II. RECEIVER MODEL

A. The CPM Signal Model

The general CPM signal model as described in [4] has a complex baseband representation as

$$s(t) = \sqrt{\frac{E_s}{T_s}} e^{j\psi(t,\alpha)} \tag{1}$$

where E_s is the energy during one symbol period T_s. The phase function $\psi(t,\alpha)$ is given by

$$\psi(t,\alpha) = 2\pi h \sum_{i=0}^{n} \alpha_i q(t - iT_s) \tag{2}$$

where $\alpha_i \in \alpha$ is the *M-ary* data symbols from a set of $\{\pm 1, \pm 3, \dots, \pm(M-1)\}$ with $M = 2^m$ and m is a positive integer. The variable h is the rational modulation index. The phase response $q(t)$ is a monotonic and continuous function represented as

$$q(t) = \begin{cases} 0, & t \le 0 \\ \int g(t)dt, & 0 < t \le LT_s \\ \frac{1}{2}, & t > LT_s \end{cases} \tag{3}$$

where $g(t)$ is the frequency response pulse which is non-zero over the time interval $[0, LT_s]$. On bases of different values of the correlation length (or pulse length) L, CPM signals can be classified as either a full response signal ($L = 1$) or a partial response signal ($L > 1$). Some commonly used formats of $g(t)$ are listed in Table I, where the REC, RC, and GMSK denote the rectangular pulse, raised cosine pulse, and Gaussian-MSK pulse, respectively. B is the 3dB bandwidth of the Gaussian pulse shape, and $Q(x)$ has the form

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{\frac{-t^2}{2}} dt \ . \tag{4}$$

Assuming the CPM signal is transmitted over an AWGN channel, the complex baseband representation of the received signal is

$$r(t) = e^{j(2\pi\nu t + \theta)} s(t - \tau) + w(t) \tag{5}$$

where ν, θ and τ are carrier frequency offset, carrier phase offset and symbol timing offset, respectively. Both $\theta \in [-\pi, \pi]$ and $\tau \in \left[-\frac{T_s}{2}, \frac{T_s}{2}\right]$ are independently distributed and have a uniform distribution. $w(t)$ is a zero-mean complex baseband white Gaussian noise with an equivalent power spectral density (PSD) of $2N_0$, and its real and imaginary parts are both independent Gaussian processes with a PSD of N_0. Note that the transmitted data symbols α are implicit in the definition of $s(t)$.

TABLE I. VARIOUS PULSE SHAPES FOR CPM

$g(t)$	Mathematical Expression	
REC	$\frac{1}{2LT_s}$,	$0 \le t \le LT_s$
RC	$\frac{1}{2LT_s}\left(1 - cos\frac{2\pi t}{LT_s}\right)$,	$0 \le t \le LT_s$
GMSK	$\frac{1}{2LT_s}\left\{Q\left[\frac{2\pi B}{\sqrt{ln2}}(t - \frac{(L+1)T_s}{2})\right] - Q\left[\frac{2\pi B}{\sqrt{ln2}}(t - \frac{(L-1)T_s}{2})\right]\right\}$,	$0 \le t \le LT_s$

B. CPM with Frequency Hopping (FH)

Considering frequency hopping and algorithms for joint synchronization, a CPM receiver model is established as shown in Fig.1. The FH-based carrier modulated CPM sequence $r_c(t)$ is processed by the band-pass filter (BPF) and low-pass filter (LPF) to obtain its baseband version $r(t)$. The baseband sequence $r(t)$ is then passed through the frequency hopping sequence acquisition (FHSA) to acquire a valid hopping sequence and estimate the sequence starting point. This process is followed by joint synchronization and decoding. The joint synchronization scheme ccomprises two stages with an architecture similar to that in [10], i.e., a burst-mode feedforward joint synchronization (BFFJS) in the first stage and a data-aided joint synchronization (DAJS) in the second stage. The model estimates the synchronization parameters ν, θ and τ jointly, and uses them to recover the transmitted FH-CPM sequence and decode the output sequence of data symbols $\tilde{\alpha}$ corresponding to the transmitted sequence α.

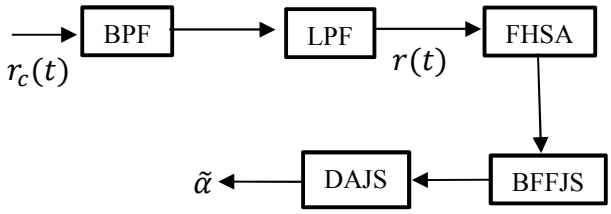

Fig 1 Joint FH acquisition and synchronization

Data transmission in a frequency hopping system is normally in burst-mode, i.e., the transmission of disjoint packets through and on each frequency hop. The received packet burst has a known duration and structure which consists of three parts. The first part is a synchronization preamble or training sequence. It consists of L_0 known and optimized data symbols which are used to estimate synchronization parameters. The second part in the burst is denoted as the unique word (UW), which is used to validate the burst sequence and determine the location of data symbols within a burst. The UW is assumed to be a pseudo-random sequence containing L_{UW} symbols. The third part of the burst is the payload, which carries $L_{payload}$ information symbols. The two-stage joint synchronization scheme, as shown in Fig.1, determines synchronization parameters ν, θ and τ jointly by observing the preamble portion of the burst and the demodulated burst data sequence.

C. Overview of the Receiver Model

Algorithms for FHSA and the two joint synchronization stages, namely BFFJS and DAJS, are not elaborated upon in this paper. Instead a brief overview is given in this subsection to establish the background for introducing a DL model in the next section.

The FHSA takes a two-stage serial search approach based on the acquisition schemes described in [6] and [7]. The first stage takes advantage of an energy detection method to perform the coarse initial acquisition to detect the presence of a FH sequence and identify the starting point of the sequence to

be within $[t_s - \frac{T_h}{2}, t_s + \frac{T_h}{2}]$, where t_s is the real start and T_h the per hop interval (i.e., hopping period). If successful, the second stage uses a correlation between the preamble combined with the UW of received sequence and a predefined codebook which is known to the receiver. This is a refined process with matched filtering energy detection which locates the starting point of the FH sequence to be within $[t_s - \frac{T_s}{2}, t_s + \frac{T_s}{2}]$. The received sequence is then passed to the joint synchronization to obtain the carrier frequency, carrier phase, symbol timing offset, i.e., ν, θ and τ, and to decode the sequence.

The BFFJS algorithm employed is the same as described in [5]. The algorithm uses the packet preamble which is an optimum training sequence devised using the Cramér–Rao bound (CRB) criterion [8]. The training sequence minimizes the CRBs for ν, θ and τ simultaneously and can be applied to the entire CPM family. The algorithm is a type of joint maximum likelihood (ML) estimation where the preamble symbol sequence is known to the receiver. The likelihood function for the estimation of a set of parameters from a waveform in AWGN is given in [9]. With the complex and constant envelope of the CPM signal, the joint log-likelihood function (LLF) including the three synchronization parameters was derived in [5]. The estimates of ν, θ and τ, denoted as $\tilde{\nu}$, $\tilde{\theta}$, $\tilde{\tau}$ respectively, are obtained by maximizing the LLF [5].

As a feedforward processing sequence, the BFFJS estimates the frequency offset first, followed by the symbol timing delay, and then the phase offset. This yields the estimation parameters efficiently and shortens the estimation time. However, the errors in the estimated frequency offset ν inevitably contaminates τ and θ, as the algorithm has no feedback mechanism to track and compensate for estimation errors. Hence, a data-aided joint synchronization (DAJS) algorithm is employed to continuously track, estimate and compensate carrier phase offset θ and symbol timing offset τ with a feedback mechanism.

The DAJS algorithm leverages the maximum likelihood decision-directed (DD) method presented in [4]. As described in [1], the CPM modulator may be viewed as a trellis encoder. Then a ML detector applying the Viterbi algorithm may search for the optimal path in the modulator trellis, assuming perfect knowledge of θ and τ. The variable values $\tilde{\theta}$, $\tilde{\tau}$ and $\tilde{\alpha}$ that maximize the log-likelihood function (LLF) are the ML estimates for θ, τ and α. That is, the algorithm computes the maximum value in the $(\tilde{\theta}, \tilde{\tau})$ plane using an estimate $\tilde{\alpha}$ of the transmitted data sequence generated from the Viterbi detector (i.e., data-aided from Viterbi detector). The $\tilde{\theta}$ and $\tilde{\tau}$ obtained from the BFFJS are used to start the Viterbi decoder. As in the equations, the $(k+1)$-th estimation of τ and θ employs the k-th $(k \leq (N-1))$ or the $(k-D)$-th estimation, where D is a delay (e.g., $D = 1$), as an error signal to make corrections [4]. This generates a feedback mechanism to update the Viterbi decoder in each step for a better $\tilde{\alpha}_{k+1}$, i.e., the $(k+1)$-th $\tilde{\alpha}$, to further optimize τ and θ in maximizing the LLF.

This receiver model has satisfactory performance when evaluated in the AWGN channel for different scenarios, as shown in Section IV. Performance however deteriorates considerably when subjected to jamming signals. In view of this, DL techniques were examined for the receiver model to evaluate their ability in mitigating noise jamming, assuming an AWGN channel. As a first step to evaluate the benefits of a DL boosted receiver, the AWGN channel is assumed that considers jamming without fading. Thus the receiver model does not include a channel equalizer to combat channel fading. In Section V, the implications and a next step that includes the channel equalizer under fading conditions are discussed.

III. Deep Learning Assisted Receiver Model

Employing artificial neural networks, deep learning is more and more applied in supporting wireless communications, especially in the area of signal processing and time series analysis [11]. A neural network (NN) consists of multiple layers becoming "deep" when the number of layers is greater than 3. A deep NN performs deep learning by employing a large number of parameters and mapping functions to capture the relationship between input data features and output responses. In the area of wireless signal processing, the NN model can be used to predict and classify the signal i.e., data.

A. The DL Boosted CPM Model

Expanding on the CPM receiver model presented in Section II, a hybrid DL NN model was designed and applied as a signal predictor in the receiver. This DL predictor aims to improve the performance of the CPM receiver model under jamming noise. As shown in Fig.2, the DL is applied to mitigate and suppress tone jamming noises, which are the spectral tones emitted by the jammer to interfere with the frequency hops. The jamming tones may severely distort the FH waveform with little energy expenditure by the jammer.

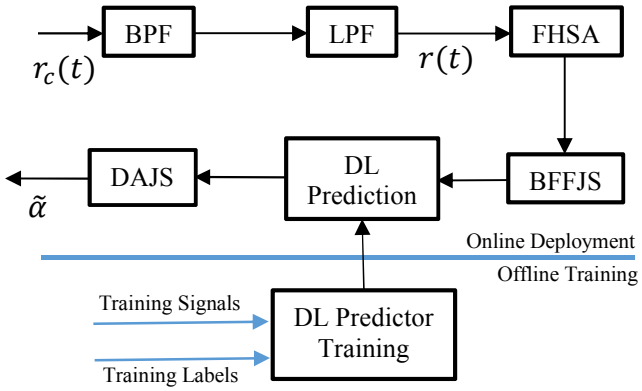

Fig.2 DL boosted CPM receiver model

In Fig.2, the DL predictor is applied after the first stage of synchronization, i.e., the BFFJS, to correct the waveform of the symbols by removing the distortion resulting from the jamming tones. In the input to the DL predictor, synchronization offsets are partially removed by the BFFJS to enable a better recognition for predicting the correct waveform of symbols. The corrected symbols out of the DL predictor are then sent to the DAJS where the performance of the Viterbi decoder can be improved.

B. The Hybrid DL Predictor

The DL predictor is built on a multilayer perceptron (MLP) incorporating a convolutional neural network (CNN) and a long short-term memory (LSTM) [4] to form a hybrid neural network model, as described in Table II. Rectified linear unit (ReLU) is used as the activation function, while mean-absolute-error (MAE) is the loss function and Nesterov Adam (Nadam) the optimizer.

TABLE II. LAYOUT OF THE MLP-CNN-LSTM MODEL

Layers	Parameter Numbers	Output Dimensions
Input (1× 32)	N/A	1 × 32
Convolution 1D (64 filters, size 1× 32) + ReLU	65,600	1 × 64
LSTM + ReLU	33,024	1 × 64
Max Pooling (size 32, stride 1)	0	1 × 64
Dense	2,080	1 × 32
Loss Function = MAE		
Optimizer = Nadam		
Learning Rate = 0.0001		

The MLP is generally robust to noise and offers non-linearity modeling power. The CNN is especially effective in extracting features from raw data, namely the transform or distortion invariance of the data and to recognize the original presentation of the data, e.g., recognizing an object in an image. Here the CNN is applied to the time series signal by taking the sequence of observations as a one-dimensional mapping to extract the most salient elements. The LSTM as a type of recurrent neural network (RNN) is used to add the explicit processing of the order between observations, which is not offered by MLP or CNN. The purpose is to capture the long-term correlation and obviate the need for a pre-specified time window in modeling the complex multivariate sequence of the received waveform. The DL predictor is trained offline. After the training, the predictor is deployed in the receiver and performs symbol prediction in real-time, much faster than any conventional optimization algorithms.

Ten sets of different synchronization parameters, i.e., ν, θ and τ, were used for training the MLP-CNN-LSTM neural network model. The results thus show how the trained model may handle the synchronization parameters not covered in the training set. Using more synchronization parameters in training would improve the performance of the DL predictor, at the expense of increased training time, computational resource and cost. In the current training configurations, 200k symbols are generated as training data, with an up-sampling rate of 16 amounting to 3.2 million samples.

Although this paper is focused on tone jamming cases because of the severity of the performance degradation they engender, different DL NN models were experimented with, in terms of full band and follower band noise jamming channels. The results will be briefly discussed in the next section.

IV. RECEIVER PERFORMANCE EVALUATION

A. Simulation Setup

The DL predictor is implemented using Python with deep learning libraries of Keras and Tensorflow [14]. After training, the predictor is applied in the receiver model as shown in Fig.2 to test across different scenarios and to cover different modulation variations. The transmitted symbols are modulated to be uncoded binary sequences ($M = 2$) by MSK, 2RC and GMSK schemes, with a modulation index $h = 0.5$. The time-bandwidth product of GMSK is 0.3, with a pulse length $L = 4$. The length of the transmitted CPM preamble is $L_0 = 64$ in default setup. $K_f = 64$ is set for zero padding on FFT operations as in [4] to increase the resolution and accuracy of frequency offset estimation in the BFFJS algorithm depicted in Section II. In the simulations, the carrier frequency, phase and symbol timing offsets are assumed to be in the range of $[-2.5\%, 2.5\%]$, $[-\pi, \pi]$ and $[-0.5T_s, 0.5T_s]$, respectively. The Viterbi algorithm in the DAJS uses a traceback delay $D = 1$. The scenario includes the following additional parameters: symbol rate $T_s = 20$ ks/s, channel bandwidth $B_s = 80$ KHz, hopping bandwidth $B = 0.8$ MHz, hopping rate $R_p = 100$ hops/s, and frame duration $T_f = 10$ ms.

Jamming noise was added to the AWGN channel, including full band, follower, single tone and multi-tone jamming noise. Single tone and multi-tone jamming signals were applied through the entire hopping sequence. The single tone jamming signal attacks at the center of each hopping sequence band, while two side tones of the multi-tone jamming are at about 25% of the hopping sequence bandwidth away from its band center. The residual SNR of the jamming tone was at 20 dB. The simulations were run on a Microsoft Azure Virtual Machine (VM) configured with 32 CPU cores and 128 GB memory. 1000 simulation iterations were executed for each scenario, and the performance results were summarized as the average of the iterations. Training of the DL predictor is offline for up to several days to cover different scenarios. Then the obtained DL predictor performs in real-time in the receiver, much faster than conventional algorithms.

B. Simulation Results

Under the AWGN channel, the receiver model in Fig.1 that has no learning capability, which is referred to as "BRM" hereafter, performs well for all three modulation schemes including MSK, 2RC and GMSK. The DL boosted receiver model, referred to as "DLRM" hereafter, delivers a performance close to the ideal case, in terms of bit-error-rate (BER). In the "ideal" case, all synchronization parameters are known by the receiver in demodulation. Fig.3 shows the result for the 2RC scheme. Improvements of the synchronization for the 2RC scheme, illustrated in Fig.4 and Fig.5, come from estimating the phase and timing offsets. In all the figures, the legend "No Deep Learning" indicates the performance results from the base receiver model (BRM) in Fig.1, whereas the legend "Deep Learning" the results from the DL boosted receiver model (DLRM) in Fig.2.

Results of the MSK scheme are similar to those of the 2RC scheme. For GMSK modulation, the BRM performance is further behind the ideal case, when compared with the MSK and 2RC schemes. The DLRM for GMSK pushes the results close to the ideal level and achieves about 1dB improvement, in terms of signal-to-noise ratio (SNR), when reaching the BER of 10^{-4}. Due to space constraints, plots for MSK and GMSK results are not illustrated here. Overall, the BRM

performs well in the AWGN channel and the DLRM further brings the performance close to the level of the ideal case.

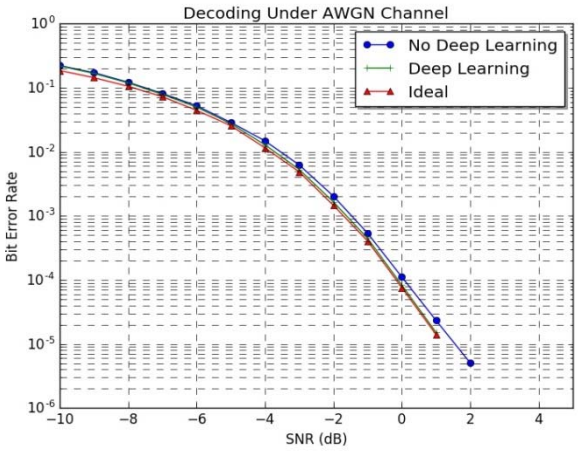

Fig.3 BER for 2RC scheme under AWGN channel

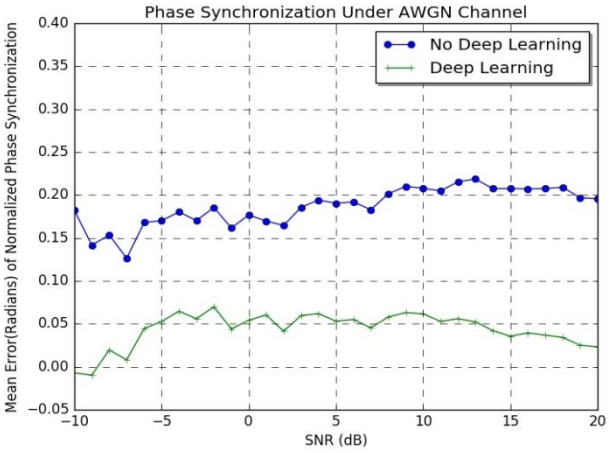

Fig.4 Mean Error of Phase synchronization for 2RC Scheme

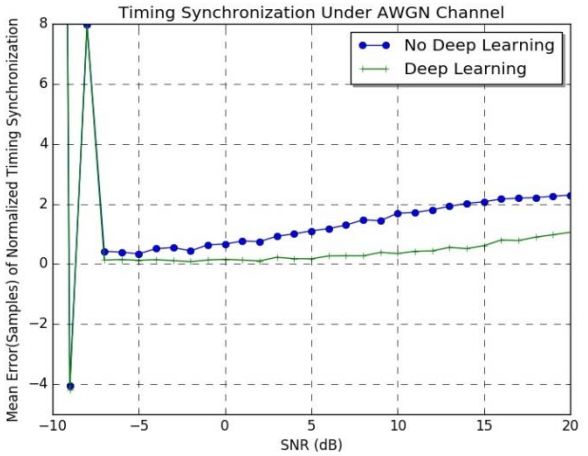

Fig.5 Mean Error of symbol timing synchronization for 2RC Scheme

Figs.6–8 illustrate the performance improvement of DLRM in the single tone jamming scenario for MSK, 2RC and GMSK, respectively. The signal-to-jamming-plus-noise ratio (SJNR) is thus used. The jamming noise severely degrades the performance of BRM. DLRM, on the other hand, exhibits 3 – 5dB improvement with the aid of the DL predictor. In the three schemes, MSK shows the biggest improvement followed by GMSK.

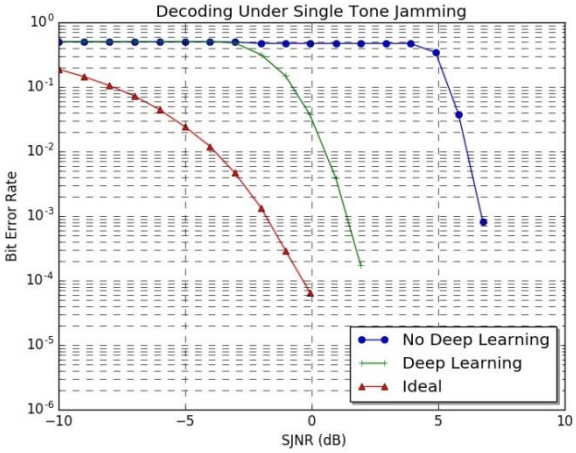

Fig.6 BER for MSK scheme under single tone jamming

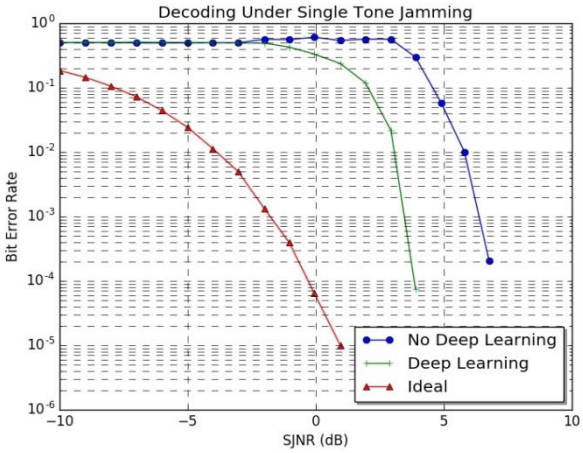

Fig.7 BER for 2RC scheme under single tone jamming

The BER performance for the GMSK scheme under multi-tone jamming conditions is illustrated in Fig.9. The results for MSK and 2RC schemes in multi-tone jamming scenarios are similar to that for GMSK, and thus are not shown here. The performance improvements under multi-tone jamming are less pronounced than those in the single-tone jamming case. This may be due to the more complex spectral patterns for the DL model to recognize during the symbol waveform prediction. However, at least 2dB of improvement can still be obtained from the DLRM.

Overall, the DLRM as presented in Fig.2 mitigates the tone jamming quite effectively due to the feature learning capability of the DL that recognizes the jamming suppressed waveforms. This enables the receiver to recover better the synchronization parameters for improved decoding performance.

Use of a DL predictor was also investigated in full band and follower noise jamming scenarios. In these scenarios, full band noise jamming is found to result in less than 1 – 2dB

degradation from the ideal case, when using the BRM without the DL predictor. However, strong follower jamming with no delay or 10% delay reduces the performance by 3 – 4dB for the BRM. To handle full band and follower jamming, the DL predictor was adapted using the neural network model with more layers of CNN and LSTM. Meanwhile, the DLRM also used a different architecture than that presented in Fig.2, in an attempt to remove overall noise from the sequence even before the BFFJS algorithm. However, the performance improvement obtained for demodulation and decoding in these scenarios was insignificant, with gains often under 1 dB. The DLRM improved the phase and timing synchronization but delivered little improvement to the frequency synchronization, and thus limited overall gain in demodulation and decoding.

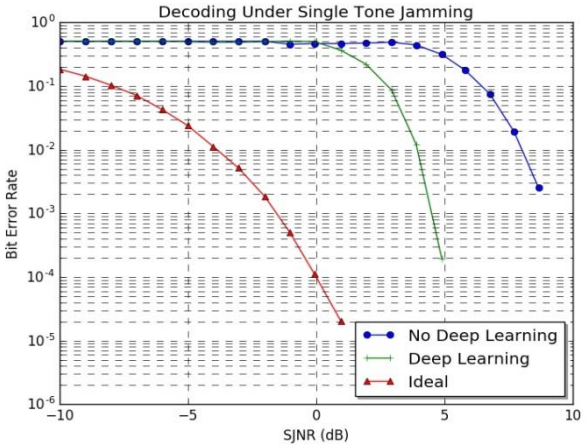

Fig.8 BER for GMSK scheme under single tone jamming

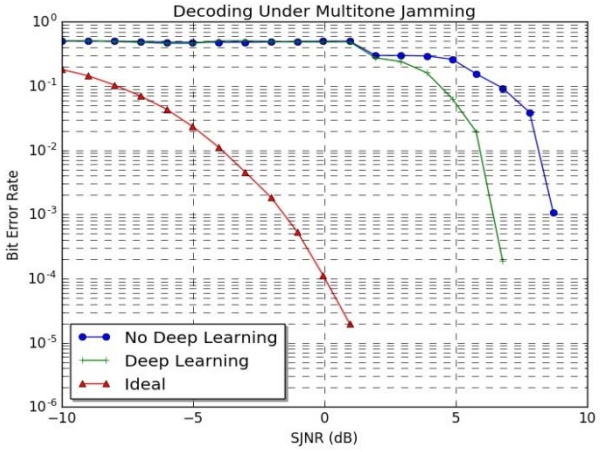

Fig.9 BER for GMSK scheme under multitone jamming

V. Conclusion Remarks

In this paper, an enhanced FH-CPM receiver is proposed that combats jamming in tactical communications applying deep learning techniques. In addition to the state-of-the-art FH sequence acquisition and joint synchronization schemes, the receiver model is expanded to utilize a DL predictor built on a hybrid neural network to improve the performance. The DL predictor employs multiple layers of CNN, LSTM and MLP to recover the FH waveform after the first stage of coarse synchronization, i.e., the BFFJS, and before the second synchronization stage, i.e., the Viterbi decoder based DAJF. The DL predictor is trained offline and then employed in the real-time receiver. The simulation results have demonstrated significant performance enhancement to the BER under tone jamming conditions. It is however important to consider the fading channels for a FH-CPM receiver, which will be the focus of future work that includes both frequency selective fading and flat fading conditions. The schemes for further improving the receiver performance under full band and follower jamming are also part of the continued study.

References

[1] J. B. Anderson, T. Aulin and C.-E. Sundberg, *Digital Phase Modulation*, New York: Plenum Press, 1986.

[2] P. A. Laurent, "Exact and Approximate Construction of Digital Phase Modulation by Superposition of Amplitude Modulated Pulses (AMP)," *IEEE Transactions on Communications*, vol. 34, no. 2, pp. 150 – 160, February 1986.

[3] C. Brown, P. J. Vigneron, "Maximum likelihood timing estimation for CPM using Legendre polynomials, " *IEEE Wireless Communications Letter*, vol. 4, no. 5, pp. 509 – 512, October 2015.

[4] M. Morelli, U. Mengali and G. Vitetta, "Joint Phase and Timing Recovery with CPM signals," *IEEE Transactions On Communications*, vol. 45, no. 7, pp 867 – 876, July 1997.

[5] E. Hosseini and E. Perrins, "Timing, Carrier, and Frame Synchronization of Burst-Mode CPM," *IEEE Transactions on Communications*, vol. 61, no.12, pp. 5125 – 5138, December 2013.

[6] C. A. Putman, S. S. Rappaport and D. L. Schilling, "Comparison of strategies for serial acquisition of frequency-hopped spread-spectrum signals, " *IEE Proceedings F - Communications, Radar and Signal Processing*, vol. 133, no. 2, pp. 129 – 137, April 1986.

[7] L.E. Miller, J.S. Lee, R.H. French and D.J. Torrieri, "Analysis of an antijam FH acquisition scheme," *IEEE Transactions On Communications*, vol. 40, no. 1, pp. 160 – 170, January 1992.

[8] E. Hosseini and E. Perrins, "The Cramér–Rao bound for training sequence design for burst-mode CPM," *IEEE Transactions on Communications*, vol. 61, no. 6, pp. 2396 – 2407, June 2013.

[9] J. G. Proakis, *Digital Communications (4th Edition)*, McGraw-Hill, Aug. 2000.

[10] W. Zhao, A. Li, J. Si, and J. Bai, "Performance analysis of a joint estmator for timing, frequency, and phase with continuous-phase modulation," *IET Communications,* vol. 10, no. 3, pp 263 – 271, 2016.

[11] T. O'Shea and J Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions On Communications & Networks*, vol. 3, no. 4, pp. 563 – 575, December 2017.

[12] H. Ye and G.Y. Li, "Initial Results on Deep Learning for Joint Channel Equalization and Decoding," *IEEE 86th Vehicle Technology Conference (VTC – Fall),* pp. 1 – 5, September 2017.

[13] C. Zhang, P. Patras and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," *IEEE Communications Surveys & Tutorials*, January 2019.

[14] Chollet, François, "Xception: Deep Learning with Depthwise Separable Convolutions", 2017 IEEE CVPR, July. 2017