NeuroWav: Toward Real-Time Waveform Design for VANETs using Neural Networks

Jayson Boubin
The Ohio State University
KeyW Corporation
Columbus/Dayton, USA
boubin.2@osu.edu

Aaron M. Jones
Air Force Research Laboratory
United States Air Force
WPAFB, USA
aaron.jones.41@us.af.mil

Trevor Bihl
Air Force Research Laboratory
United States Air Force
WPAFB, USA
trevor.bihl.2@us.af.mil

***Abstract*—Vehicular Ad-Hoc networks depend on clear communication between vehicles using radio frequency in order to operate effectively. Interference from existing technologies using the RF spectrum, e.g. IoT devices, UAV, mobile systems, calls into question the feasibility of future VANET systems without an ability to cut through the noise. One approach to overcome interference is to use waveform design to provide this capability. Regrettably, most traditional algorithms are too computationally complex to perform efficiently in real-time. In this paper, we present early work on NeuroWav: a neural network based approach to waveform design to combat the effects of interference at low latency. NeuroWav is low size, weight, and power, executes 10X faster than the fastest extant waveform design algorithms, and provides performance results comparable with a high fidelity waveform design algorithm. Simulation results are provided that corroborate the theoretical expectations.**

I. Introduction

The future of autonomous driving is dependent on resilient and reliable network technologies [6], [8]. Vehicular Ad-Hoc networks (VANETs) are particularly sensitive to interference [3], [12]. Saturation of the radio frequency (RF) spectrum by IoT systems, aerial systems, mobile devices, and other technologies [1], [4], [5], [16] leaves limited spectrum for new systems [18]. It is most important to these systems to improve signal-to-interference-plus-noise ratio (SINR) while considering additional practical concerns, such as modulus, range sidelobes, algorithm convergence rate, etc.

Researchers have created waveform design techniques that increase SINR in saturated RF [2], [19], [20]. Waveform design methods, e.g. cognitive radio, exist to sense the environment and change the waveform based on sensed interference to increase the SINR in saturated RF. One technique, spectral notching, places notches within a waveform allowing signal to avoid spectral interference while remaining at a constant modulus, and without affecting signal reliability [13], [17].

Neural networks have much potential for waveform design problems. Neural networks could be used to classify input waveforms to general cases which correspond to precomputed output waveforms. A sufficiently small neural network may be capable of classifying an input waveform to its corresponding notched output at considerably lower latency than extant waveform design algorithms. While neural networks are known for their high latency, many neural network implementations are capable of real-time inference [14], [15].

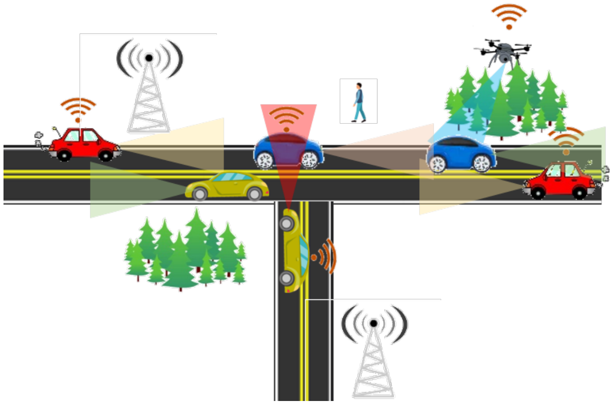

Fig. 1: VANETs in congested/dynamic RF environments.

The goal of this paper is to present an early solution to cognitive waveform design using neural networks. We present our neural network approach to waveform design, NeuroWav, which is capable of classifying input waveforms at high accuracy in real-time on conventional compute hardware (i.e. under 10ms).

Section 2 of this paper presents the spectral notching problem and design of NeuroWav. Section 3 discusses the implementation of NeuroWav. Section 4 covers early results of NeuroWav using generated waveform data. Section 5 discusses future work and conclusions.

II. Problem Formulation

Two algorithms used to generate spectral notched waveforms are the Error Reduction Algorithm (ERA) [7] and the Re-Iterative Uniform Weight Optimization algorithm (RUWO) [9]. ERA uses an iterative approach to design waveforms with spectral notches from inputs and has a fast convergence rate (10's ms with CPU). RUWO uses either a deterministic model of interference or empirical interference data to construct spectral notches, providing better spectral nulling quality at a much higher convergence time. Despite the effectiveness of these techniques, algorithms like ERA and RUWO are too high latency to merit use in real-time, low size, weigh, and power (SWaP) systems. The objective of this initial work is to achieve or improve upon the convergence rate of the ERA, with similar waveform characteristics/performance.

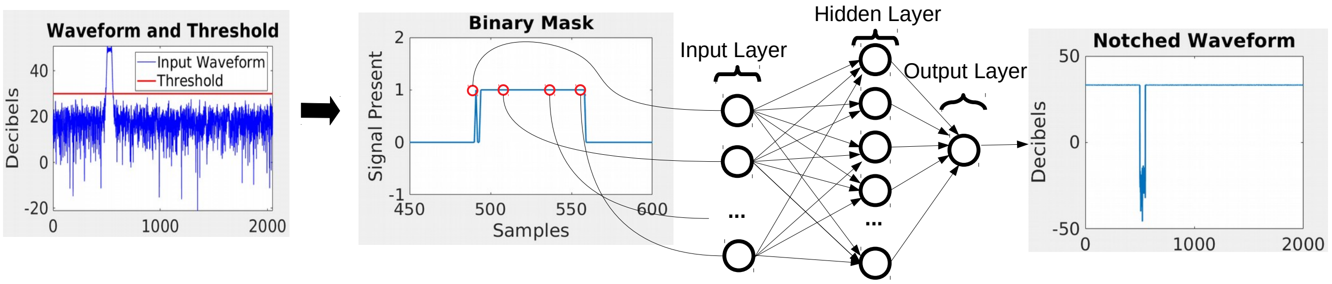

Fig. 2: Data Generation and NeuroWav process. a) Example data is generated and displayed in the Fourier domain. b) Waveform is transformed into a binary mask. c) Select points on the binary mask are provided as input to the neural network. d) Neural network predicts the class of our waveform.

A. *Transmit and Interference Signal Model*

Given the continuous time, pulsed waveform, s, whose envelope is constant non-zero over $t \in [0, T]$ for duration T

$$\mathrm{s}(t) = \alpha(t)\exp\{j(2\pi f_c t + \phi(t))\} \tag{1}$$

where α is pulse envelope, f_c carrier frequency and ϕ the phase, we desire to design the waveform to avoid interference while maintaining $||\alpha||_2^2 = 1$. We critically sample the baseband waveform to express in complex-valued vector form, $\underline{\mathrm{s}} \in \mathbb{C}^{M\times 1}$, as

$$\underline{\mathrm{s}} = \underline{\alpha} \odot \exp\{j\underline{\phi}\}. \tag{2}$$

The interference (scenario depicted in figure 1) is modeled simply as a noise corrupted linear frequency modulated (LFM) signal for ease of use (a more realistic interference models from prior work could be applied [11]). The digitized interference and noise can be described by defining $\underline{n} = [0, 1, 2...(M-1)]$ as the vector of samples M within the signal, and f_s the sampling frequency, $\underline{t}_\mathrm{s} = \underline{n}/f_\mathrm{s}$, the time sample vector, thus the interference $\underline{\Lambda}$ is defined in discrete form as:

$$\underline{\Lambda}(\underline{t}_\mathrm{s}, f_0, B, T) = \exp\{j2\pi(f_0\underline{t}_\mathrm{s} + \frac{B}{2T_i}\underline{t}_\mathrm{s}{}^2)\} + \underline{\mathrm{n}} \tag{3}$$

where f_0 is the initial frequency, B the bandwidth, and T_i the duration of the LFM sweep, equal to the simulation extant, see related work [10] for an example.

For a digitally sampled interference environment with Fourier transform, $\mathbf{I} = \mathbb{F}(\underline{\Lambda})$, we threshold $|\mathbf{I}| > \gamma$, where γ is a tunable parameter for desired SINR. Following the basic thresholding, a binary mapping of each Fourier bin is expressed where 0 represents a usable bin and 1 represents interference signal present and thus unsuitable for transmission. This process generates the binary mask depicted in figure 2.

B. *Example Data Generation*

Our dataset includes 93 waveforms with unique notch locations generated from the RUWO algorithm (the same waveforms were generated from the ERA algorithm for comparison purposes). Each waveform is then modified with independent identically distributed complex Gaussian noise $\underline{n} = \mathbb{CN}(0, \underline{\sigma}_l^2)$ where l determines the variance from set $\sigma^2 = [0 : 2 : 20]$ and 100 draws from each provide a total of 93000 waveforms for training and classification. The varying noise values are analyzed for robustness to the SINR.

Additional pre-processing is necessary. Briefly, before data is input into the network, relevant features must be extracted. Feature extraction includes converting our observed interference signal to a binary mask, where values of 1 represent samples where signal strength exceeds a given threshold. This process is completely deterministic for a finite number of Fourier bins, and one can make additional reasonable assumptions such as minimum signal width and maximum number of signals in practical scenarios. This further limits the possible cases to study.

Our networks training set consists of sampled feature points from our binary mask waveforms. For each neuron in our input layer, feature points from our binary mask are selected as input to the network at even intervals, spanning the mask. As seen in figure 2, feature points extracted from the binary mask include the first and last points in the mask, along with evenly spaced samples in between. This sampling-based process, given enough feature points, allows NeuroWav to gain sufficient information about our waveform to classify it without having to take as input the entire waveform. This process allows NeuroWav to remain low SWaP and low latency while maintaining high accuracy.

C. *Pitfalls and Assumptions*

NeuroWav has assumptions aside from those previously stated. The threshold used to create the binary mask is crucial. Too high or low a threshold will return a highly noisy mask that is difficult to classify. For our dataset, our threshold is fixed at 30 dBm. Additional processing time may be required to dynamically determine thresholds for problem domains that necessitate it.

Our problem contains 93 output classes. For a problem with a higher order of output classes, a larger network may be necessary, which may increase latency. Despite this, NeuroWav's early implementation leaves room for speedup improvements in both implementation environment and low power hardware acceleration discussed in Section 5.

III. Neural Network Approach

Our early NeuroWav approach uses a neural network to match input waveforms with appropriate notched outputs generated by RUWO.

Figure 2 describes our neural network architecture. We use a feed forward artificial neural network with a variable number of input nodes which take in waveform features, as well as a variable number of hidden units. Both the number of inputs and hidden units in our network are hyper-parameters. Inputs are gathered from our thresholded waveform dataset as integers. For each node in the input layer, we input the integer position of a sample in our input waveform binary mask. Samples are selected at even intervals along the binary mask as shown in Figure 2. Each input label corresponds to a waveform type. They are provided to the network as integers; one for each of the 93 possible classes.

Our neural network was implemented in Matlab version 2019a using the deep learning toolbox. Our network is implemented as a feed forward multi-layer perceptron with an input layer of variable nodes, 1 hidden layer of variable nodes, and one single neuron output layer. Our hidden layer uses the sigmoid activation function. The training procedure relies on the Levenberg-Marquardt optimizer for error backpropagation. Our network trains on 80% of our 93000 waveforms for 1000 epochs. 10% of our waveforms are reserved for validation, and the final 10% are reserved for testing.

Our network has a series of hyper-parameters, most important being the number of both input nodes and hidden units. We also have to consider the environmental factors of SINR and latency in analyzing the performance of our network compared to other waveform design techniques. In section 4, we analyze the interactions between these hyper-parameters and environmental factors in detail as they compare to our networks performance against ERA and RUWO.

IV. Results

In this section, we display some of the more relevant analysis for typical VANET operations. In summary, we observe improved performance in convergence rate while achieving the prescribed performance of the traditional ERA and RUWO approaches.

A. Convergence Rate

In figure 3, we present the convergence rates distributions in seconds of the 3 approaches: NeuroWav trained using 15 sample points and 50 hidden units, ERA with a convergence parameter value of 10^{-3} and RUWO. Latencies were captured using a Lenovo Thinkpad™ T470 laptop running Ubuntu 18.04 with an I7 7500u processor and 24GB of RAM.

RUWO has the slowest convergence rate, with a mean of 18 seconds, which is far too high for real-time applications. ERA with a small epsilon averages 90ms, but produces inferior results to RUWO. NeuroWav produces precomputed RUWO results in 7ms on average. Notably, this is 1 order of magnitude faster than ERA at the same task, and 3-4 magnitudes faster than RUWO.

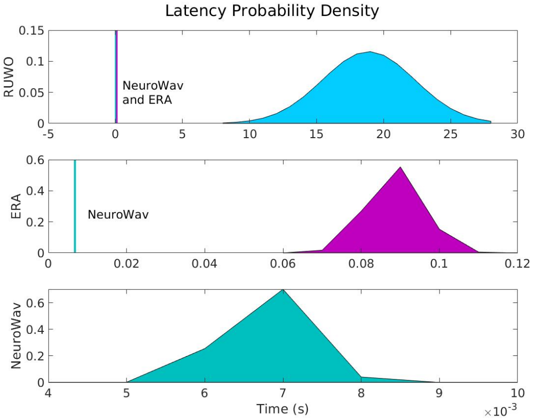

Fig. 3: Distribution of Convergence Time

B. Spectral Null Depth

As first discussed in prior work [10] for a pulsed application, a waveform cannot entirely eliminate the energy contained within the stopband region(s). However, the depth of the nulled region may be sufficient if the energy within the stopband does not significantly interfere with the other users. Here, the null quality metric Δ_{nd} is defined as as the ratio of the average power contained within the nulled region to the passband region as

$$\Delta_{\text{nd}} = \frac{\text{n}(\Omega_{\text{pb}})}{\text{n}(\Omega_{\text{sb}})} \frac{\sum\limits_{i\in\Omega_{\text{sb}}} |\underline{s}_{\text{f},i}|^2}{\sum\limits_{k\in\Omega_{\text{pb}}} |\underline{s}_{\text{f},k}|^2} \tag{4}$$

where Ω_{pb} and Ω_{sb} are the sets containing the Fourier bin indices $0, \ldots, M-1$ corresponding to the passband and stopband regions of the spectrum. The $\text{n}(\bullet)$ operation returns the size of the set. In table I we report the mean statistic for each of the three signal design algorithms null quality. We note that NeuroWav and RUWO have the same null quality despite NeuroWav's decreased latency.

TABLE I: Average Spectral Null Depth

Algorithm	**Null Quality**
ERA	18.5 dB
RUWO	27.8 dB
NeuroWav	27.8 dB

C. SINR vs Accuracy

The SINR of our input signals is an important metric to consider when determining accuracy and the feasibility of NeuroWav operating in real world systems. SINR is defined as follows:

$$\text{SINR} = \frac{||\underline{s}||_2^2}{\underline{s}^H \mathbf{K} \underline{s}} \tag{5}$$

where $\mathbf{K}$ is the interference and noise covariance matrix.

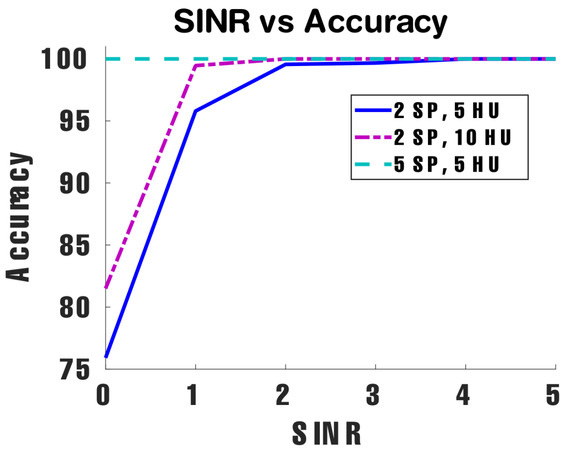

Fig. 4: SINR in dB versus classification accuracy.

VANETs operate in highly dynamic conditions and must be sensitive to low SINR signals to operate correctly. Perfectly clear signals can by no means be assumed. In figure 4, it can be seen that NeuroWav is adept at classifying inputs even at very low SINR.

We present results for NeuroWav trained at 3 different hyper-parameter configurations to demonstrate our SINR resilience. Our samples differ in the number of sample points (SP) they intake, and in the number of hidden units (HU) each network has. Our simplest setting (2SP, 5HU) takes the least time to execute, but has the lowest SINR/Accuracy ratio. At 0dB SINR, our network achieves an accuracy of 75.81%. As SINR increases, our network converges to 100% accuracy, reaching 100% at 4dB SINR. Adding hidden units (2SP, 10HU) improves accuracy but slightly increases training and inference time. By increasing sample points (5SP, 5HU), we are able to increase accuracy to 100% even at 0dB SINR. While increasing sample points increases training and inference time, NeuroWav still meets real-time constraints with upwards of 50 hidden units and 15 sample points as demonstrated in Figure 3.

V. Conclusions & Future Work

NeuroWav is an early approach to real-time waveform design that has serious implications for VANETs. VANETs capable of real-time waveform design will be able to cut through the RF noise of city environments, facilitating quick connections to cloud systems and other vehicles. To implement large scale vehicular autonomy policies, this capability is a must. NeuroWav uses a novel feature extraction approach to classify input waveforms with deep spectral notches in under 10ms on conventional hardware. Using this technique, VANETs and other RF systems can take better advantage of the RF band to cut through our saturated airspace.

NeuroWav is an early approach to this problem. Future work will focus on pushing NeuroWav towards designing waveforms from scratch using recurrent and spiking networks in real-time. To meet real-time constraints as well as remain low SWaP, research into low power neuromorphic hardware accelerators will be crucial.

Acknowledgment

This work was supported by the United States Air Force Sensors Directorate. However, views and opinions expressed are those of the authors and do not necessarily reflect official policy or position of any agency of the U.S. government.

References

[1] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.

[2] Mark R Bell. Information theory and radar waveform design. In *Proceedings. IEEE International Symposium on Information Theory*, pages 428–428. IEEE, 1993.

[3] Jeremy J Blum, Azim Eskandarian, and Lance J Hoffman. Challenges of intervehicle ad hoc networks. *IEEE transactions on intelligent transportation systems*, 5(4):347–351, 2004.

[4] Jayson Boubin, Naveen Babu, Christopher Stewart, John Chumley, and Shiqi Zhang. Managing edge resources for fully autonomous aerial systems. In *ACM Symposium on Edge Computing*, 2019.

[5] Jayson Boubin, John Chumley, Christopher Stewart, and Sami Khanal. Autonomic computing challenges in fully autonomous precision agriculture. In *2019 IEEE International Conference on Autonomic Computing (ICAC)*. IEEE, 2019.

[6] Pietro E Carnelli, Mahesh Sooriyabandara, and R Eddie Wilson. Large-scale vanet simulations and performance analysis using real taxi trace and city map data. In *2018 IEEE Vehicular Networking Conference (VNC)*, pages 1–8. IEEE, 2018.

[7] James R Fienup. Phase retrieval algorithms: a comparison. *Applied optics*, 21(15):2758–2769, 1982.

[8] Jerome Harri, Fethi Filali, and Christian Bonnet. Mobility models for vehicular ad hoc networks: a survey and taxonomy. *IEEE Communications Surveys & Tutorials*, 11(4):19–41, 2009.

[9] Thomas Higgins, Tegan Webster, and Aaron K Shackelford. Mitigating interference via spatial and spectral nulling. *IET Radar, Sonar & Navigation*, 8(2):84–93, 2014.

[10] A. M. Jones and P. McCormick. Spectral notching via error reduction algorithm with relaxed papr constraint. In *2019 IEEE Radar Conference (RadarConf)*. IEEE, 2019.

[11] Aaron M Jones, Brian D Rigling, and Muralidhar Rangaswamy. Generation of synthetic uhf rfi in urban north american environments. In *2016 IEEE Radar Conference (RadarConf)*, pages 1–6. IEEE, 2016.

[12] Mehdi Khabazian, Sonia Aissa, and Mustafa Mehmet-Ali. Performance modeling of message dissemination in vehicular ad hoc networks with priority. *IEEE Journal on Selected Areas in Communications*, 29(1):61–71, 2010.

[13] B. Ravenscroft, S. D. Blunt, C. Allen, A. Marione, and K. Sherbondy. Analysis of spectral notching in fm noise radar using measured interference. In *International Conference on Radar Systems (Radar 2017)*, pages 1–6, Oct 2017.

[14] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[16] Giuseppe Virone, Andrea M Lingua, Marco Piras, Alberto Cina, Federico Perini, Jader Monari, Fabio Paonessa, Oscar A Peverini, Giuseppe Addamo, and Riccardo Tascone. Antenna pattern verification system based on a micro unmanned aerial vehicle (uav). *IEEE Antennas and Wireless Propagation Letters*, 13:169–172, 2014.

[17] LS Wang, JP McGeehan, Chris Williams, and Angela Doufexi. Application of cooperative sensing in radar–communications coexistence. *IET communications*, 2(6):856–868, 2008.

[18] Michael Wicks. Spectrum crowding and cognitive radar. In *2010 2nd International Workshop on Cognitive Information Processing*, pages 452–457. IEEE, 2010.

[19] Michael Wicks and Eric Mokole. *Principles of waveform diversity and design*. The Institution of Engineering and Technology, 2011.

[20] Michael C Wicks. A brief history of waveform diversity. In *2009 IEEE Radar Conference*, pages 1–6. IEEE, 2009.