

DSIC: Deep Learning based Self-Interference Cancellation for In-Band Full Duplex Wireless

Hanqing Guo, Shaoen Wu

Department of Computer Science Dept. of Electrical and Computer Engineering Dept. of Business Intelligence & Analytics
Ball State University University of Massachusetts Dartmouth Stevens Institute of Technology
Muncie, IN USA Dartmouth, MA Hoboken, NJ
{hguo, swu}@bsu.edu hwang1@umassd.edu Mahmoud.Daneshmand@stevens.edu

Honggang Wang

Mahmoud Daneshmand

Abstract—In-band full duplex (IBFD) wireless is of utmost interest to future wireless communication and networking due to great potentials of spectrum efficiency. IBFD wireless, however, is throttled by its key challenge, namely self-interference. Therefore, effective self-interference cancellation is the key to enable IBFD wireless. This paper proposes a real-time non-linear self-interference cancellation solution: *Deep learning based Self-Interference Cancellation (DSIC)* to enable IBFD wireless. In this solution, a self-interference channel is modeled by a deep neural network (DNN). Synchronized self-interference channel data is first collected to train the DNN of the self-interference channel. Afterwards, the trained DNN is used to cancel the self-interference at a wireless node. This solution has been implemented on a USRP SDR testbed and evaluated in real world in multiple scenarios with various modulations in transmitting information including numbers, texts as well as images. It results in the performance of 17dB in digital cancellation, which is very close to the self-interference power and nearly cancels the self-interference at a SDR node in the testbed. The solution yields an average of 8.5% bit error rate (BER) over many scenarios and different modulation schemes.

Although some solutions have been proposed in the literature, their performance is not satisfactory.

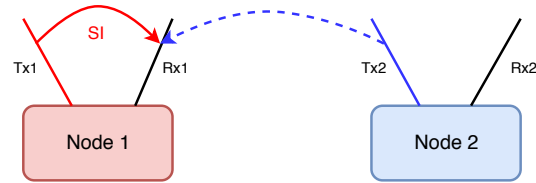


Fig. 1: IBFD Self-Interference Scenario

In this paper, we target self-interference cancellation (SIC), the toughest issue in IBFD wireless. We propose a non-linear SIC solution based on deep learning. We have implemented this solution on a USRP SDR testbed and evaluated its performance in the real world with this prototype. In this work, we have answered three questions: (1) how to collect synchronized wireless channel information to train the deep learning model? (2) how to model a wireless channel with a deep neural network? (3) how to implement an open source SDR IBFD wireless framework to work in real world?

In the rest of this paper, Section II reviews the background of IBFD wireless and literature solutions related to our work. Next, Section III presents the SI problem with a simulation, then formulates the theoretical foundation for digital SIC solutions, and describes our deep learning SIC solution (DSIC) in details. Then, Section IV presents an open source SDR framework implementing IBFD wireless on a testbed, followed by Section V summarizing the performance of our solution in the tests. This work is finally concluded in Section VI.

I. INTRODUCTION

Traditional wireless communication, widely used today, supports bi-directional transmissions in the way of time-division duplex (TDD) or frequency-division duplex (FDD). In-band full duplex (IBFD) wireless refers to the bi-directional transmission on the same frequency band at the same time, which intuitively boosts the spectrum efficiency and channel utilization. As the demand of wireless transmission is sharply increasing, IBFD wireless is very promising in the future.

Although IBFD wireless has been studied for years in theory and analysis, it has been progressing very slowly for practical use because of a very challenging issue called self-interference to be addressed in engineering. Refer to the IBFD wireless architecture shown in figure 1 where the wireless node Node1 is receiving data [101, 102, 103, ..., 120] via its antenna RX1 from the TX2 of Node2, while it is also transmitting data via its antenna TX1. The consequence is that the power emitted by the TX1 of Node1 will interfere the TX2's signal received at the RX1 of Node1. Such interference is called self-interference (SI), which completely overshadows the received signal and leads to the communication failure from Node2 to Node1. To enable IBFD wireless, SI has to be cancelled so that the overshadowed received signal can be recovered.

II. RELATED WORK

Since the first systematic study of IBFD wireless on a narrow band [1], several solutions have been proposed for IBFD wireless on larger bandwidths [2]–[5]. These IBFD wireless SIC solutions can be classified into 3 different categories [6]: (1) antenna cancellation, (2) analog cancellation and (3) digital cancellation.

Digital cancellation can be further classified into linear digital cancellation and non-linear digital cancellation. Linear digital cancellation aims to cancel the deformed digital signals passing over the environment channel, while non-linear digital cancellation targets at cancelling non-linear cubic and higher order elements generated by radio circuits [5], [7]. A team

at Stanford University designed an FIR filter to emulate the components of the line of sight signals, then used at modeled SI channel to emulate and generate the SI signals at a full duplex node [5]. They also included a non-linear component in SIC by using a general model in the form of Taylor Series Expansion. Another work [8] proposed a pre-calibration-based cancellation technique that linearizes a transmitter and cancels SI with linear-only cancellation at a receiver, which basically, converted a non-linear problem to a linear problem and then solved it. Recently, a significant milestone in SIC is present from Columbia University [9], their digital SIC algorithm with a highest non-linearity order of 7 is implemented in LabVIEW and further suppress the residual SI signal after RF SIC, their result shows it has averagely 43 dB digital SIC across 20 MHz.

As far as the performance of IBFD wireless solutions is concerned, some works have reported that the channel capacity could be doubled on a single-hop link, while in mesh network the actual benefit of IBFD wireless may be effected by spatial reuse and asynchronous contention [10]. Other works [5], [11] shows 1.87 times throughput improvement and 1.42 times capacity over a half-duplex system at a transmission power of 20 dBm. A recent work [12] shows that at a certain situation, IBFD wireless can double the downlink and uplink throughputs of static TDD. Another recent work reported a performance improvement by 30% to 66% compared to the 5G Dynamic TDD [12]. Intelligence was also considered for beamforming [13].

III. DEEP LEARNING BASED DIGITAL SELF-INTERFERENCE CANCELLATION

A. Theoretical Foundation of Digital Self-Cancellation

Refer to Figure 1 that shows a basic architecture of IBFD wireless. Suppose that, at a wireless node, the self-transmitted signal $x[n]$ passes through the SI channel and is transformed into $y_{si}[n]$ at the receiver, and denote $y[n]$ the received signal at the receiver that is a mixture of the SI signal $y_{si}[n]$ and the useful signal $m[n]$ from another node. The following formula (1) can be used to recover the useful signal $m[n]$. Since the system knows of its self-transmitted signal $x[n]$, and the received mixture signal $y[n]$, hence then, the key challenge to recover the useful signal $m[n]$ is to compute the SI signal $y_{si}[n]$ based on the known signal $x[n]$.

$$m[n] = y[n] - y_{si}[n] \quad (1)$$

Denote $h[n]$ the impulse response of SI channel, then the SI signal $y_{si}[n]$ can be calculated as:

$$y_{si}[n] = \sum_{k=0}^K h[k]x[n-k] \quad (2)$$

Based on Formulas (1) and (2), the useful message $m[n]$ can be obtained if we know $h[n]$. In practice, however, we can't obtain any precise SI channel impulse response $h[n]$ beforehand. Therefore, what we can do is to estimate the SI channel response $\hat{h}[n]$ as accurate as possible, which is used

to replace the real impulse response $h[n]$ in estimating the SI signal $\hat{y}_{si}[n]$ as below where K refers to the number of taps:

$$\hat{y}_{si}[n] = \sum_{k=0}^K \hat{h}[k]x[n-k] \quad (3)$$

$$\hat{m}[n] = y[n] - \hat{y}_{si}[n] \quad (4)$$

to get $\hat{m}[n]$ as recovered useful signal.

To demonstrate the feasibility of the digital cancellation concept, a simulation was performed on a GNU Radio SDR platform, with a framework implemented as in Figure 3. There are two digital vector sources: useful signal $m[n]$ being $[1, 2, \dots, 10]$, and the SI signal vector $[101, 102, \dots, 110]$, as shown on the top plots in Figure 2. Both of them use the QPSK modulation, but pass through different simulated channel models of $h[k]$, which are determined by the settings of taps. The taps of the useful signal $m[n]$ *wireless channel* are $[1]$, while the taps in the *SI channel* are $[1, 1]$, and the *estimated SI channel* taps are $[1.1, 0.9]$. After passing through channel models, all of those three streams go into Decimating FIR filters, which will be described in Section IV. Then the first time sink receives $m[n]$, the second time sink receives $y_{si}[n]$, the third time sink receives $y[n]$, and the last time sink gets the recovered signal $\hat{m}[n]$ by using formula (4).

The simulation result shown on the bottom plots in Figure (2) indicates that, even though the estimated SI channel response $\hat{h}[k]$ is not completely the same as $h[k]$, the recovered signal $\hat{m}[n]$ (on the right bottom plot) is actually very similar to the useful signal $m[n]$ (one the left top plot), in that the real and image parts of those signals have very similar shapes, and the relation between the real and image numbers of signal I-Q phases is almost kept. This result proves that if $\hat{h}[k]$ are estimated accurately, then $m[n]$ can be recovered from $y[n]$, even if $y[n]$ is severely polluted by the SI signal $y_{si}[n]$.

B. Deep Learning Digital Cancellation

As discussed above, the core of digital SI cancellation is to estimate the SI channel response $\hat{h}[k]$. Rather than modeling and estimating the SI channel with traditional wireless channel models as most of literature solutions do, we propose to model the SI channel response with a deep neural network (DNN).

1) *SI Channel Modeling with DNN*: A DNN model is introduced to estimate the SI channel and its structure is shown in Figure (4).

a) *Input and Output*: In this DNN model, the input is denoted as $x[n], x[n-1], \dots, x[n-K]$, and the output is referred as $y_e[n]$. Instead of modeling the SI channel as a one-to-one mapping, we model it as a many-to-one DNN, because the many-to-one model has a similar structure to an FIR filter. Considering Formula (3), to estimate $\hat{y}_{si}$, we need to compute the sum of $\hat{h}[k]x[n-k]$, where k varies from 0 to K , which means that every output of the SI channel not only relates to the current input $x[n]$, but also previous K samples of x .

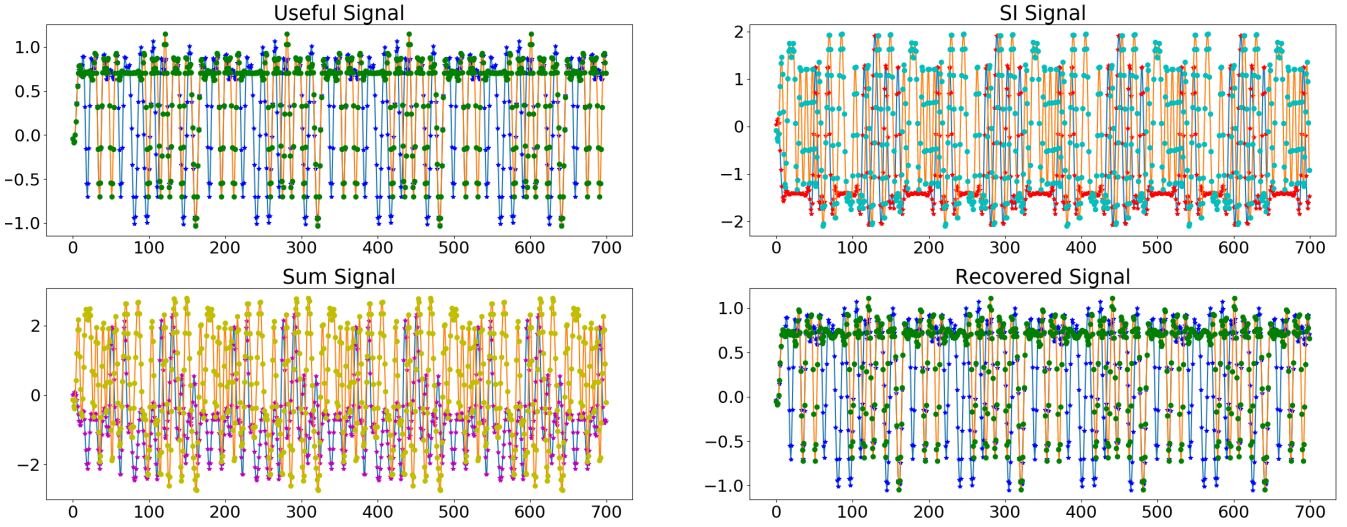


Fig. 2: Digital Cancellation Validation Results

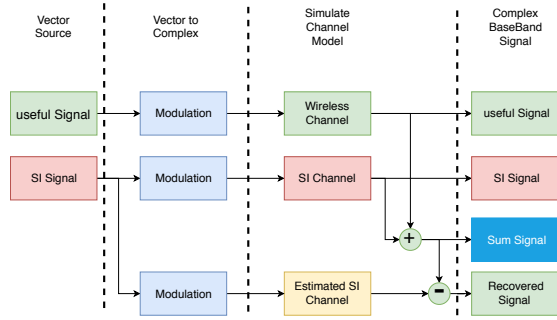


Fig. 3: Digital Cancellation Validation Framework

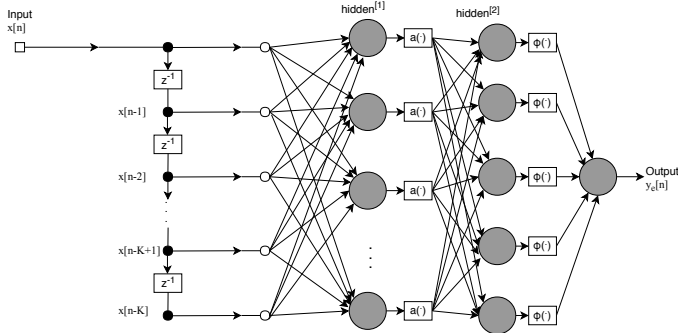


Fig. 4: DNN Model for SI Channel

b) *DNN Structure*: The simplified and parameterized structure of the designed DNN is shown in Figure (5). From the figures of (4) and (5), this DNN has 2 hidden layers, each of which is followed by an activation function $a(\cdot)$ or $\Phi(\cdot)$. The activation function enables the channel modeling a non-linear structure. We choose the Relu function for both $a(\cdot)$ and $\Phi(\cdot)$. It should be noted that there is no activation function after the last layer, because the DNN is used to compute mapping values, whose range could be any possible rational number, and it doesn't need something like Softmax or any other classification function. The first linear block takes K input signals and generates an output of 1×20 dimensions,

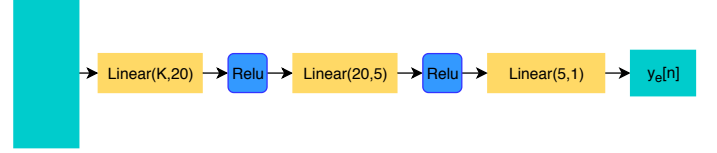


Fig. 5: Simplified Network

where K can be seen as the size of FIR filter taps and means how many samples are related to output sample. Then the second linear block takes 1×20 input signals and generates an output of 1×5 dimensions, and the last linear block takes the 1×20 signals and output a value of 1×1 dimension. In each linear block, the weight matrix and tge bias matrix are respectively denoted as $W^{[l]}$ and $b^{[l]}$, where l is the layer number in the DNN. Denote $o^{[l]}$ the output of linear block, the output of an activation function is referred as $a^{[l]}$. As a result,

$$W^{[l]} = \begin{bmatrix} w_{1,1}^{[l]} & w_{1,2}^{[l]} & w_{1,3}^{[l]} & \dots & w_{1,20}^{[l]} \\ w_{2,1}^{[l]} & w_{2,2}^{[l]} & w_{2,3}^{[l]} & \dots & w_{2,20}^{[l]} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{K,1}^{[l]} & w_{K,2}^{[l]} & w_{K,3}^{[l]} & \dots & w_{K,20}^{[l]} \end{bmatrix}$$

According to Figure 5, the whole process to work on input x can be summarized by Formula (5), where x is a matrix containing from $x[n-K]$ to $x[n]$ and its dimension is $(1 \times K)$.

$$\begin{aligned} o^{[1]} &= xW^{[1]} + b^{[1]} & a^{[1]} &= a(o^{[1]}) \\ o^{[2]} &= a^{[1]}W^{[2]} + b^{[2]} & a^{[2]} &= a(o^{[2]}) \\ o^{[3]} &= a^{[2]}W^{[3]} + b^{[3]} & y_e[n] &= o^{[3]} \end{aligned} \quad (5)$$

c) *Loss Function*: We choose the Mean Square Error (MSE) as the loss function, because the difference between an estimate $y_e[n]$ and the real $y[n]$ could be an either positive or negative number. The optimization goal of this DNN model is to generate the $y_e[n]$ as similar as possible to $y[n]$. As a result, the optimization during the DNN training is formulated as:

$$loss = \sum_{m=0}^M ||y_m - y_{e,m}||^2 \quad (6)$$

where M is the mini-batch size of training data and $y_{e,m}$ is an output of the DNN when feeds $m_t h$ training data, and y_m is a collected real SI sample coming through the SI channel. Each parameter of the DNN network such as $W^{[l]}$ and $b^{[l]}$ will be updated through back-propagation in every epoch by deviating this loss function.

2) *DNN Training*: To train this DNN, we need to construct a training dataset to be in the dimension of $((N - K + 1) * K)$ due to the input x dimension is $(1 * K)$, where N is total number of samples in training dataset. To achieve this, a recurrent scheme is employed to generate K samples for each input. For example, suppose the input samples are $[s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9]$, and K equal to 3, the dataset is constructed as below :

$$dataset = \begin{bmatrix} s_1 & s_2 & s_3 \\ s_2 & s_3 & s_4 \\ s_3 & s_4 & s_5 \\ \vdots & \vdots & \vdots \\ s_7 & s_8 & s_9 \end{bmatrix}$$

Then a training mini-batch is generated by randomly selecting a mini-batch size number of rows from the dataset. The benefit of using mini-batch to feed data is the faster and more stable convergence of loss function compared to using one sample at one time, because the loss derivation takes more than one samples so that any exceptional sample will only have little impact on the direction of convergence.

C. Deep Learning Digital Cancellation Framework

With the SI channel modeled by a DNN discussed above, our proposed deep learning digital cancellation framework, as illustrated in 6, includes: (1) SI channel probing, (2) SI channel data collection, (3) DNN SI channel model training, and (4) DNN model data loading for digital cancellation, which are presented in details in the following.

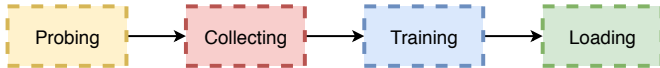


Fig. 6: Basic Steps of Our Design

1) *SI Channel Probing*: The first step of our design is to probe the SI channel. In this period, an IBFD wireless node sends designated probing signals or vectors x from its transmission chain while its receiver chain senses what is received. The designated probing signals will pass the SI channel and transform to unknown received SI signals denoted by y_{si} .

2) *SI Channel Data Collection*: Following the SI channel probing, the framework collects and records the I and Q components of both the probing complex signals x and the received SI signals y_{si} . Both of them are baseband complex signals, and their I and Q components comprise constellation points on a modulation constellation map.

3) *DNN SI Channel Model Training*: With the data of x and y_{si} collected, they are then used to train a DNN SI channel model in a supervised learning way. The structure of the DNN SI channel model is explained in details in the next section III-B1.

4) *Loading for Digital Cancellation*: After the DNN SI channel model is sufficiently trained, the trained model is loaded into the receiver chain. In IBFD wireless communication later, this trained DNN channel model estimates the SI signal $\hat{y}_{si}$ according to the Formula (5). Meanwhile, the receive chain recovers the received useful signal m based on Formula (3) with the SI signal $\hat{y}_{si}$ estimated by the trained DNN SI channel model.

IV. PROTOTYPE IMPLEMENTATION

To perform the assessment in real world, we have implemented our solution into a SDR prototype testbed of GNURadio and two USRP X310 nodes. GNURadio is the software that can customize the wireless radio and works with UHD, the USRP node driver. USRP X310 nodes are generic SDR hardware. To implement our solution to work in the real world, several challenges are confronted.

A. Modulation in Gnuradio

The first challenge is in modulation. To achieve a higher data transmission rate, modulation functions should be implemented. The Modulation block in GNURadio takes byte stream as input, and output $I - Q$ complex samples. If we want to apply our design on those $I - Q$ complex samples, we have to design a way transforming message data from byte stream to those complex samples.



Fig. 7: Inner Block in Modulation

1) *Constellation Mapping*: In GNURadio, the first block in Modulation is the Constellation Mapping block, which takes k bits, where $k = \log_2 M$, and M is number of constellation points, and generates output pulse of In-phase (I) and Quadrature (Q) signals. Suppose the input is a char 57, which takes one byte in a bit sequence of 00111001. Take QPSK as example, in this case, $M = 4$ and $k = 2$, if the constellation map is designed as Figure (8a). Every time only two bits of the bit sequence 00111001 are modulated, i.e. 00 to $1 + 0j$, 11 to $0 - 1j$, 10 to $0 + 1j$ and 01 to $-1 + 0j$. As a result, the modulated complex sequence is $[1 + 0j, 0 - 1j, 0 + 1j, -1 + 0j]$ for this bit sequence as shown in Figure (8b).

2) *Up-sampling and Pulse shaping*: After symbols are generated as shown in Figure (8b), they are processed with up-sampling. The up-sampling ratio depends on a sample-per-symbol parameter. For example, an up-sampling ratio of 4 means every symbol is represented by 4 samples. Noted that a sample in digital information is a sampling point in a continuous signal. Since square pulse has large ISI, a pulse shaping filter follows up-sampling, which is a root-raised-cosine filter to reshape the pulse to transmit in the channel. This filter results in a better frequency domain response, and it achieves 0 ISI in theory when two root-raised-cosine filters are combined

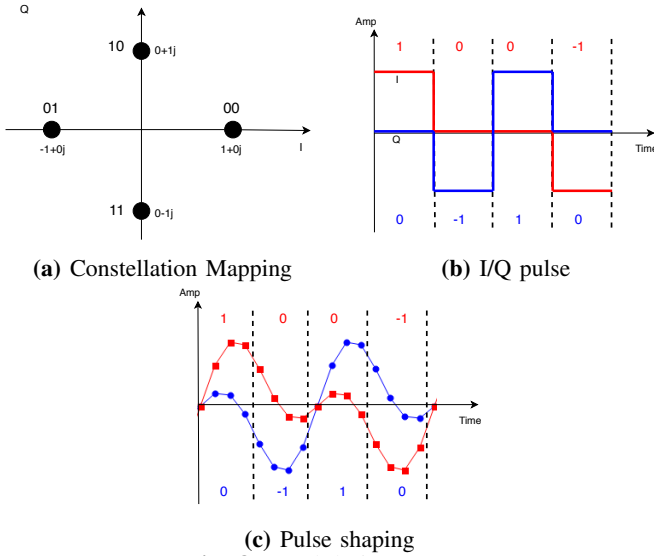


Fig. 8: Modulation Process

together as one raised cosine filter as in Equation (7). Figure 8(c) shows what I/Q samples look like after up-sampling and pulse shaping. Those samples are then the exact inputs to the SI channel, and are what we need to collect.

$$H_{rc}(f) = H_{rrc}(f) \cdot H_{rrc}(f) \quad (7)$$

B. Synchronized Data Collection

The second challenge to the prototype is how to synchronize the collected channel data. It is very difficult to collect synchronized input signals to SI channel and their corresponding output signal from the SI channel because there is a random delay in the signal processing chain of USRP hardware. Collecting the input signals to the SI channel data starts as soon as the program starts. However, the output signal from the SI channel data to be collected by the receiving chain of the USRP node is randomly delayed. This delay can be host and USRP communication delay or driver communication delay.

As for the USRP random delay problem, researchers have presented a method to address this issue by calculating the barker code correlation [14]. A barker code is added at the head of every source signal sequence. At a receiver, once if the receive chain has all barker codes in its buffer, it can extract a synchronized signal. We have implemented a barker code based frame synchronization block in GNURadio, to identify a synchronized signal in the frame synchronization block by inserting a probe function in main stream of GNURadio signal processing chain.

C. Loading Module

Another implementation challenge is about loading the DNN model for digital cancellation, because there is no deep learning module in GNURadio. After we collect synchronized SI channel input and output data and train the DNN channel model, we save all of the DNN parameters such as $W^{[l]}$ and $b^{[l]}$ into a dictionary data structure. The DNN model is implemented in Pytorch, but a loading module block is created in GNURadio to load the trained model parameter dictionary, and the known input signal is forwarded to this model to

generate an estimated $y_e[n]$. Algorithm 1 below shows the implementation of loading and SI cancellation.

Algorithm 1 SI Cancellation

Input: input vector source $x[n]$, baseband complex signal $y(t)$

Output: output vector source $\hat{m}[n]$

Description: This algorithm input $x[n]$ and $y(t)$, where $y(t)$ is sum complex signals contains SI signal $y_{si}(t)$ and useful signal $\hat{m}(t)$, then cancel $\hat{y}_{si}(t)$ from $y(t)$ to recover $\hat{m}[n]$

```

1:  $N = \log_2 M$ 
2: for  $k < \text{length}(x[n])$  do
3:    $x(t) = \text{Modulate}(x[k], N)$ 
4:    $k = k + 1$ 
5:  $\hat{y}_{si}(t) = \text{load\_model}([x(t), x(t-1), \dots, x(t-\text{taps} + 1)], \text{path})$ 
6:  $\hat{m}(t) = y(t) - \hat{y}_{si}(t)$ 
7:  $\hat{m}[n] = \text{Demodulate}(\hat{m}(t), N)$ 
8: return  $\hat{m}[n]$ 

```

V. PERFORMANCE

We have conducted extensive performance evaluation of our solution with the implemented prototype in real world.

A. DNN Loss Convergence

First, we have tested various DNNs with different K to see the impact of the input size on the DNN parameter convergence. Figure 9 shows the convergence dynamics of the MSE loss with $K=2, 20, 50$ and 100 respectively. Because each byte contains 8 bits, every 2 bits are modulated into a QPSK symbol, and each symbol is represented by 4 samples, 16 samples consequently are needed to represent one byte. Therefore, any sample size smaller than 16 does not contribute to the learning. This is why the sample size 2 gives nothing for the network to learn as illustrated on the figure. When the sample size is larger than 16, the DNN learns something as shown in Figure 9b. With the taps size increased to 100, the loss convergences very quick and stays at a very low level, which means the DNN model can generate a predicted SI $y_{si}[n]$ very close to the real SI $y_{real}[n]$.

B. SI Estimation and Cancellation

We have then tested the performance of the DNN channel model in estimating the SI after the training. Figure 10 shows the difference between the real SI signals and the estimated SI signals. The difference is smaller than 0.02, which means this solution can cancel at least $10 \log_{10} \frac{0.3}{0.02} = 7\text{dB}$, and the average performance is $10 \log_{10} \frac{0.3}{0.005} = 17.8\text{dB}$.

C. Bit Error Rate with SI Cancellation

Further experiments have been conducted to assess the impact of various modulations on the bit error rate in four environment settings: Room1, Room2, Outdoor, and Hallway. We transmit a vector stream $[1, 2, 3, \dots, 10]$ in transmission chain with three types of modulations (QPSK, 16PSK, and 64PSK) at the USRP node1, then cancel the SI in GNURadio with the trained model, next demodulate the complex signals after the

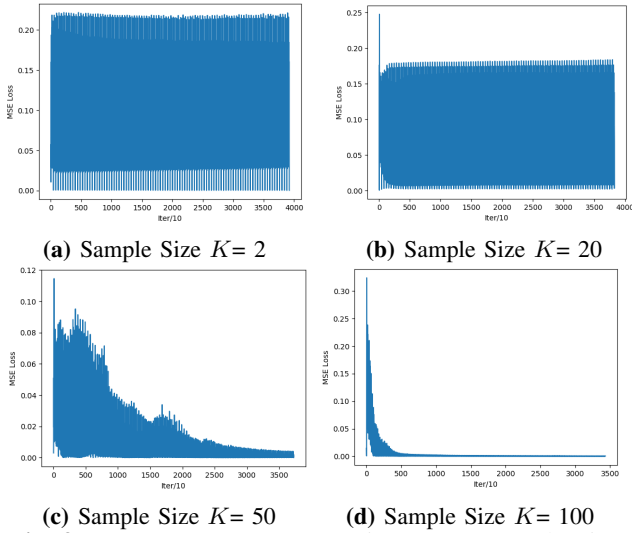


Fig. 9: Loss Convergence at Various Input Sample Sizes

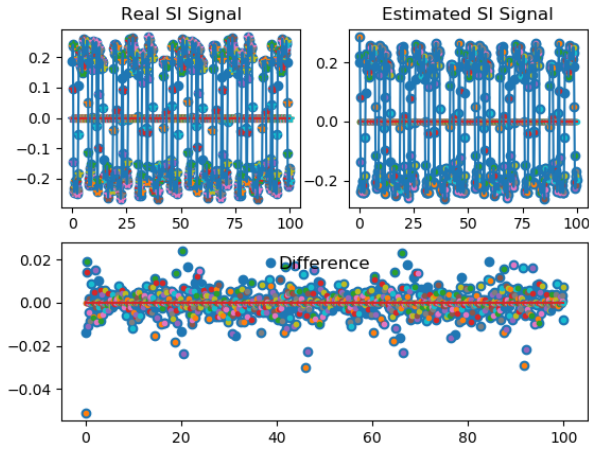


Fig. 10: SI Estimation with 100 Taps

SI cancellation to get the vector stream [101, 102, 103, ..., 120] transmitted by the USRP node2. After that, the reconstructed vector stream after the SI cancellation is finally compared with the original vector stream sent by the USRP2 node2 to calculate the bit error rates. These experiments have been repeated for many times in each of the four scenarios, and the average bit error rates are calculated as shown in Table I.

Scenario/BER(%)	QPSK	16PSK	64PSK
Room1	8.5	25	37.4
Room2	11.2	28	41.6
Outdoor	8.0	21	37.9
Hallway	13.3	27.5	45.1

TABLE I: BER vs Modulations

From the data in the table, the outdoor environment yields the best performance with our solution. The worst result is in the hallway, which indicates multi-path fading has a large impact on the performance. Another observation is that the performance degrades as the modulation level increases, which matches to the expectations that lower modulation level is more robust to transmission errors.

VI. CONCLUSION

In this work, we propose a novel non-linear digital cancellation method based on deep learning to achieve IBFD wireless.

This solution first probes the channel to collect SI channel data, which are used to train the SI channel DNN. The trained parameters are used to estimate the SI for cancellation in the realtime communication. We have implemented this deep learning SI cancellation solution into a GNURadio prototype testbed with USRP nodes. The extensive performance evaluation shows that (1) the deep learning approach is feasible to achieve very high accuracy in digital cancellation, (2) the proposed deep learning model has a fast and stable convergence in the training, (3) the deep learning solution can work across various modulations in different environments. As other signal processing modules such as error coding are added, it is reasonable to expect better performance.

VII. ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation under Grant No. #1726017.

REFERENCES

- [1] S. Chen, M. Bench, and J. McGeehan. Division-free duplex for wireless applications. *IEEE Electronics Letters*, (34):147–148, 1998.
- [2] Katsuyuki Haneda, Eino Kahra, Shurjeel Wyne, Clemens Icheln, and Pertti Vainikainen. Measurement of loop-back interference channels for outdoor-to-indoor full-duplex radio relays. In *Proceedings of the Fourth European Conference on Antennas and Propagation (EuCAP)*, pages 1–5. IEEE, 2010.
- [3] D. W. Bliss, P. Parker, and A. R. Margetts. Simultaneous transmission and reception for improved wireless network performance. In *IEEE/SP 14th Workshop on Statistical Signal Processing*, 2007.
- [4] M. Duarte and A. Sabharwal. Full-duplex wireless communications using off-the-shelf radios: Feasibility and first results. In *Forty-Fourth Asilomar Conference on Signals, Systems, and Components*, 2010.
- [5] Bharadia, Dinesh, McMilin, Emily, Katti, and Sachin. Full duplex radios. *SIGCOMM Comput. Commun. Rev.*, 43(4):375–386, Aug 2013.
- [6] Dongkyu Kim, Haesoon Lee, and Daesik Hong. A survey of in-band full-duplex transmission: From the perspective of phy and mac layers. *IEEE Communications Surveys & Tutorials*, 17(4):2017–2046, 2015.
- [7] Ramez Askar, Thomas Kaiser, Benjamin Schubert, Thomas Haustein, and Wilhelm Keusgen. Active self-interference cancellation mechanism for full-duplex wireless transceivers. In *Cognitive Radio Oriented Wireless Networks and Communications*. IEEE, June 2014.
- [8] Min Soo Sim, MinKeun Chung, Dongkyu Kim, Jaehoon Chung, Dong Ku Kim, and Chan-Byoung Chae. Nonlinear self-interference cancellation for full-duplex radios: From link-level and system-level performance perspectives. *IEEE Communications Magazine*, 55(9):158–167, 2017.
- [9] Tingjun Chen, Mahmood Baraani Dastjerdi, Jin Zhou, Harish Krishnaswamy, and Gil Zussman. Wideband full-duplex wireless via frequency-domain equalization: Design and experimentation. *arXiv preprint arXiv:1812.01126*, 2018.
- [10] Xiufeng Xie and Xinyu Zhang. Does full-duplex double the capacity of wireless networks? In *INFOCOM, 2014 Proceedings IEEE*, pages 253–261. IEEE, 2014.
- [11] Sanghoon Kim and Wayne E Stark. On the performance of full duplex wireless networks. In *Information Sciences and Systems (CISS), 2013 47th Annual Conference on*, pages 1–6. IEEE, 2013.
- [12] Osama Al-Saadeh and Ki Won Sung. A performance comparison of in-band full duplex and dynamic tdd for 5g indoor wireless networks. *EURASIP Journal on Wireless Communications and Networking*, 2017(1):50, 2017.
- [13] Shanzhi Chen, Shaohui Sun, Qiubin Gao, and Xin Su. Adaptive beamforming in tdd-based mobile communication systems: State of the art and 5g research directions. *IEEE Wireless Communications*, 23(6):81–87, 2016.
- [14] Andriyan Bayu Suksmono. A simple solution to the uncertain delay problem in usrp based sdr-radar systems. *arXiv preprint arXiv:1309.4843*, 2013.