Shifting-Window Kernel Recursive Least Squares for Self-Interference Cancellation in 5G Transceivers

Simulations based on the Research Paper by Christina Auer, Thomas Paireder, Mario Huemer
Rishi Nandha V - EE21B111
EE6110 - Adaptive Signal Processing

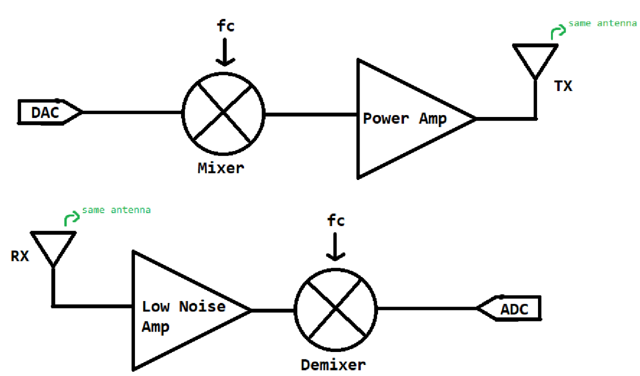
November 24, 2023

1 Introduction

1.1 Self-Interference in RF Transceivers

Radio Frequency Transceivers are components used for both sending and receiving Radio Frequency Signals simultaneously with the same antenna. The transmitted signal and received signals are generally in different frequency bands which enables such simultaneous utilization of the same antenna. Nevertheless, in a practical RF Transceiver, the numerous non-idealities will invariably cause some noise in the received signal. Techniques to mitigate such noisiness are developed every day and continue to be an active area of development.

Noise in RF Transceivers are mainly caused by the random noises in the environment, interference from other signals in the surrounding and self-interference. Self-Interference is caused when the signal transmitted by the transceiver affects the signal received by it. Although these are made to be in different frequency bands, the non-idealities in the Analog parts used in the transceiver distort this self-interfered signal to cause noise in the received signal's band of interest. For completeness, here is a picture depicting a typical RF Transceiver's TX and RX halves with the parts labelled.



The Mixer modulates any given broadband signal into a radio frequency signal which is then amplified by the Power Amplifier and transmitted using the Antenna (TX). The same Antenna receives signals (RX), amplifies them with a Low Noise Amplifier and demodulates it back to the broadband signal.

1.2 Complexity in MI-MO 5G Systems

Modern communication systems such as LTE and 5G use techniques such as Orthogonal Frequency Division Multiplexing (OFDM) and implement a Multiple-Input-Multiple-Output (MI-MO) system. Typically, Adaptive Filters used to actively cancel this interference used to have a model of the interference phenomenon itself being implemented so that the interference can be filtered out surgically. But with an MI-MO system, the complexity of such modelling increases manifold with several degrees of freedom. Thus for an effective online deployment, we require a way to auto-regressively filter out any non-linear interference without having prior knowledge of the interference. Thus we use *Kernel Adaptive Filters* in such 5G and LTE systems. In particular, we use the *Kernel Recursive Least Squares* (KRLS) algorithm

1.3 Comparison of Sparsification Techniques in Kernel Adaptive Filters

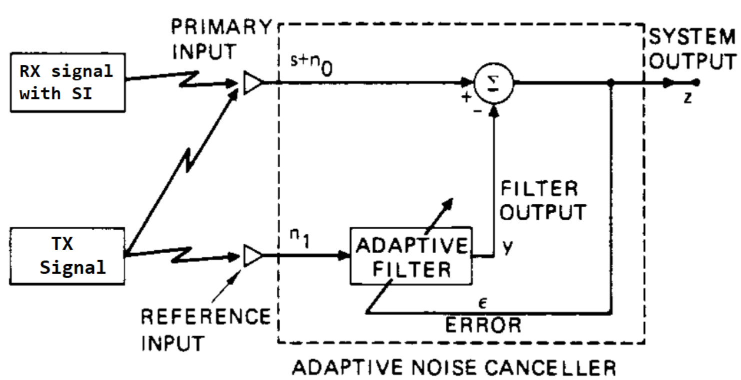
Sparsification techniques are techniques used to narrow down upon the data-points that will be used for auto-regression. If all the data-points since the start of a filter were used for computation each time, the computational complexity would approach $\mathcal{O}\left(n^3\right)$. Thus, we use methods to restrict our attention to a few of these data-points thereby reducing our computational cost to $\mathcal{O}\left(kn^2\right) = \mathcal{O}\left(n^2\right)$. Two such methods are Approximately Linear Dependent Sparsification and Sliding Window Sparsification.

Approximately Linearly Dependent (ALD) Sparsification is a very common technique. Any additional data-point in the input signal is discarded if it is Linearly Dependent on the already existing set of data-points to a certain error bound. Sliding Window (SW) Sparsification is taking only a certain number of input samples that preceeded the current one in a shifting structure. In this paper, we hypothesize that SW is a suitable choice for the Kernel Adaptive Filter in a 5G Transceiver because it responds well to a sudden change in the self-interference.

2 Filter Design

2.1 Implementation for Active Noise Cancellation

Our Adaptive Filter is the Transmitted Signal as the input signal (say n_1). The output signal is subtracted from the Receiver's signal. Then the result is fed as the error signal. Let the received signal be represented as a sum of the ideal signal s, gaussian noise n_0 and self-interference n_1^*. Then below is the equivalent block diagram for the filter.



Let the filter's output be y. Then, $\epsilon^2 = E[(s+n_0+n_1^*-y)^2] = E[(s+n_0)^2] + E[(s+n_0)(n_1^*-y)+(y-n_1^*)^2]$. The Expected Values have been grouped such that the components correlated with n_1 are separated out from the components that are not. $s+n_0$ is independent of n_1 and this expression is minimized when y is approaches n_1^*. Thus this cancels out Self-Interference.

2.2 Kernelization

Linear Adaptive Filters are extended to be able to model non-linear relationships by mapping the input data to a higher dimension and then performing the same filtering actions on this augmented set of data. For this we first choose a function $\kappa : \mathcal{R}^{1\times n} \times \mathcal{R}^{1\times n} \to \mathcal{R}$ such that this can be written as:

$$\kappa(x_1, x_2) = \phi(x_1)\phi^T(x_2)$$

This $\phi : \mathcal{R}^{1\times n} \to \mathcal{R}^{1\times m}$; $m > n$ is known as a feature map. We estimate $\hat{y}$ as $\phi(x)\omega$ now instead of using x directly. So that weights updation in SGD based becomes $\omega_{i+1} = \omega_i + \mu\phi^T(x_i)e_i$ where $e_i = d_i - \phi(x_i)\omega_i$. Since we will be starting with an initial guess of 0 for ω we can write $\hat{y}_{i+1}$ as:

$$\hat{y}_{i+1} = \mu\sum_{j=1}^{i}\phi(x_{i+1})\phi^T(x_j)e_j = \sum_{j=1}^{i}\kappa(x_j, x_{i+1})\alpha_j$$

This can be conveniently written using the "Feature Matrix" K as shown below and weights α as $\hat{y} = K\alpha$

$$K = \begin{bmatrix} \kappa(x_n, x_n) & \kappa(x_{n-1}, x_n) & \dots \\ \kappa(x_n, x_{n-1}) & \kappa(x_{n-1}, x_{n-1}) & \dots \\ \dots & \dots & \dots \end{bmatrix} \; ; \; \hat{y} = \begin{bmatrix} \hat{y}_n \\ \hat{y}_{n=1} \\ \dots \end{bmatrix}$$

For our filter, we have $n = 1$. We can use $\kappa(a, b) = e^{\frac{-(a-b)^2}{2}}$

2.3 Sparsification

The Feature Matrix is restricted to use only a selected few x values. This is called Sparsification. A specific set of features are chosen $\varphi = \begin{bmatrix}\phi^T(x_{d1}) & \phi^T(x_{d2}) & \dots & \phi^T(x_{dM})\end{bmatrix}^T$ and we have $K = \varphi\varphi^T$. For our filter, we use a Sliding Window, in which we take φ to contain the most recent M set of features.

3 Self-Interferences

3.1 TxH2 Self-Interference

Second order Harmonics at TX caused when the signal to be transmitted goes through the Power Amplifier and gets distorted due to the non-linearity in the gain are abbreviated as "TxH2".

Let $f(t)$ be the TX broadband signal that needs to be modulated and transmitted. After modulation the Signal would be $\Re\left\{f(t)\cdot e^{(j2\pi f_{TX}t)}\right\}$. Second Order terms in the Power Amplified signal would vary with $\Re\left\{\left(f(t)\cdot e^{(j2\pi f_{TX}t)}\right)^2\right\}$. These are then received by the RX and Then the signal is finally demodulated back to broadband. Any phase delay or band-pass filtering can be modelled using a convolution with some $h_1(t)$. Thus the Self-Interference in the final demodulated broadband signal would be:

$$\boxed{SI_{TxH2} = \Re\left\{h_1(t) * \left((f(t))^2 e^{j2\pi(2f_{TX}-f_{RX})t}\right)\right\}}$$

3.2 IMD2 Self-Interference

Second order Inter-Modulation Distortion caused due to the non-idealities in Demodulation at RX are abbreviated as "IMD2".

$\Re\left\{f(t) \cdot e^{(j2\pi f_{TX} t)}\right\}$ is the TX Signal received by the RX. Any delay or filtering can be modelled with a $h_1(t)$. Thus the self-interference in the signal that goes into the LNA and the Demixer is $\Re\left\{h_1(t) * \left(f(t) \cdot e^{(j2\pi f_{TX} t)}\right)\right\}$. A nonlinearity of the LNA or an undesired coupling between the RF Signal and local oscillator (LO) terminals of the mixer in combination might create even-order inter-modulation distortions. We restrict our attention to only second-order distortions out of these. In the below expression, $h_{DC}(t)$ is an impulse response to model the rejection of DC component.

$$\boxed{SI_{IMD2} = \Re\left\{\left(h_1(t) * \left(f(t) e^{j2\pi f_{TX} t}\right)\right)^2 * h_{DC}(t)\right\}}$$

4 Simulation

4.1 Setup

For simplicity, we ignore the factor that models the modulation for now. That is, our Self-Interference model now is:

$$SI = \frac{(h(n) * x(n))^2}{5}$$

Now to test the hypothesis that a Sliding Window KRLS filter is resilient to sudden changes in the interference pattern we use two different filters $h(n)$ for first 500 and the next 500 samples. We use the two filters below:

$$H_1(z) = 1 + 0.07z^{-1} + 0.5z^{-2} + 0.8z^{-3}$$
$$H_2(z) = 1 - 0.4z^{-1} - 0.66z^{-2} + 0.7z^{-3}$$

We send a randomized binary signal $x(n)$ through this filter. We make the ideal signal that the RX shouldve received without self-interference a uniformly switching binary signal with an added gaussian white-noise.

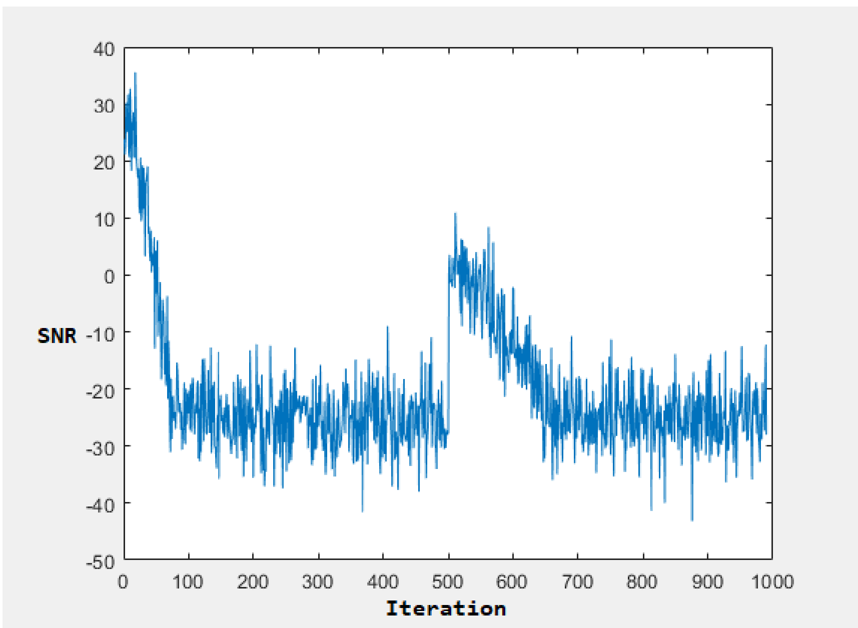
```
clear

TX = randi([0 1], 1, 1000);
RX_ideal = awgn(repmat([0 1],1, 500),30,'measured');
H1 = [1 0.07 0.5 0.8];
H2 = [1 -0.4 -0.66 0.7];
SI = [filter(H1, 1, TX(1:1:500)),filter(H2, 1, TX(501:1:1000))];
SI = (SI.^2)/5;
RX = RX_ideal + SI

save('data.mat','RX','TX','RX_ideal')
```

4.2 Algorithm

```
M = 50;
y_pred = zeros(1,M-1);
e = zeros(1,M-1);
E = 0.001;
P = eye(M)/(1 + E);
for n = M:1000
    G = P(2:M,2:M) - (P(2:M,1)*P(1,2:M))/P(1,1);
    k = kappa(x, n-M+2, n, n-M+1, n-M+1);
    a = G*k;
    g = 1/(kappa(x,n,n,n,n)+E-(k')*a);
    P = [G + a*(a').*g , -a.*g ; -(a').*g , g]
    alpha = P*(y(n-M+1:n))';
    y_pred = [y_pred, ((kappa(x,n-M+1,n,n-M+1,n-M+1))')*alpha];
end
function result = kappa(x,r_start, r_end ,c_start, c_end)
    result = zeros(r_end-r_start+1, c_end-c_start+1);
    for r = r_start:r_end
        for c = c_start:c_end
            result(r-r_start+1,c-c_start+1) = exp((-(x(r)-x(c))^2)/2);
        end
    end
end
```

4.3 Results

The filter did indeed respond well to the sudden change introduced.



5 Bibliography

- https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4410463
- https://ieeexplore.ieee.org/document/9723127
- https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9147041
- https://ieeexplore.ieee.org/document/1315946
- https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9320545
- https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1451965
- https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9638460