In-Band Full-Duplex Communications

In-band full-duplex communications:
1. Up to twice the throughput wrt TDD & FDD
2. No additional bandwidth
3. No wasted time or frequency resources

Fundamental Challenge
Self-interference (SI) signal is much stronger than the desired signal and needs to be cancelled!

• In principle, SI cancellation should be easy since the digital transmitted signal is known
• In practice, the digital signal does not tell the whole story!

Polynomial Non-Linear Digital Cancellation
Captures IQ imbalance, PA non-linearities (up to order $P$, memory $M$), and channel memory ($L$) using a complex polynomial model:

$$ y(n) = \sum_{p=0}^{P} \sum_{q=0}^{P} \sum_{m=0}^{M+L-1} h_{p,q}(m)x(n-m)^q x^*(n-m)^p $$

Number of parameters: $n_{\text{poly}} = (M + L) \left( \frac{P+1}{2} \right) \left( \frac{P+1}{2} + 1 \right)$

Complexity Analysis
Best-case scenario assumptions:
• Basis function computation is free
• Complex mults require 3 real mults and 0 real additions

Complexity:
1. Real additions: $n_{\text{ADD,poly}} = 2(n_{\text{poly}} - M - L - 1)$
2. Real multiplications: $n_{\text{MUL,poly}} = 3(n_{\text{poly}} - M - L)$

Neural Network Non-Linear Digital Cancellation
Two-step digital cancellation:
1. Linear digital cancellation: $\hat{y}_{\text{lin}}(n) = \sum_{m=0}^{M+L-1} h_{1,1}(m)x(n-m)$
2. Train a neural network to cancel: $y_{\text{nn}}(n) \approx y(n) - \hat{y}_{\text{lin}}(n)$

Single-layer NN with $n_h$ neurons and ReLU activation functions.

Complexity Analysis
1. Real additions: $n_{\text{ADD,NN}} = (2M + 2L + 3)n_h$
2. Real multiplications: $n_{\text{MUL,NN}} = (2M + 2L + 2)n_h$

Experimental Setup
• Dataset: Full-duplex testbed with 53 dB analog cancellation
• General: 10 MHz OFDM signal, 10 dBm transmit power, 20,000 samples (90% training, 10% test), $M + L = 13$
• Polynomial: $P = 7$, LS training
• NN: $n_h = 17$, MSE cost, Adam optimizer ($\lambda = 0.004, B = 32$)

Self-Interference Cancellation Results
Cancellation performance:

NN training convergence:

Complexity Comparison

<table>
<thead>
<tr>
<th></th>
<th>Polynomial</th>
<th>NN</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additions</td>
<td>492</td>
<td>493</td>
<td>0%</td>
</tr>
<tr>
<td>Multiplications</td>
<td>741</td>
<td>476</td>
<td>36%</td>
</tr>
</tbody>
</table>

Conclusions & Future Work
• Neural network seems very promising: same performance as the complex polynomial model, but with lower complexity
• Convergence and complexity of training need to be compared
• Scenarios with higher non-linear cancellation have to be compared
• Hardware implementations have to be compared