Through simulation and hardware results, we demonstrate that the proposed algorithm is robust to noise and computationally efficient compared to the existing methods.

Hardware contributions are two-folded:

- We design a hardware for modulo operation, which can handle signals up to $D^7$-bit ADC (say $D^0$ provides the folding time instant information).
- We propose a fast and robust algorithm that uses this extra one-bit information.

Desirable

- High sampling rate
- Expensive and power consuming ADCs

Motivation

- Sampling is an important building block in an ADC
- We design a hardware for modulo operation, which can handle signals up to $D^7$-bit ADC (say $D^0$ provides the folding time instant information).
- We propose a fast and robust algorithm that uses this extra one-bit information.

Through simulation and hardware results, we demonstrate that the proposed algorithm is robust to noise and computationally efficient compared to the existing methods.

Hardware contributions are two-folded:

- We design a hardware for modulo operation, which can handle signals up to $D^7$-bit ADC (say $D^0$ provides the folding time instant information).
- We propose a fast and robust algorithm that uses this extra one-bit information.

Through simulation and hardware results, we demonstrate that the proposed algorithm is robust to noise and computationally efficient compared to the existing methods.

**4. Hardware**

- Modulo board able to fold signals up to 10KHz
- ADC’s DR [-1.25V, 1.25V]. Hardware can fold signals which are eight times larger than ADC’s DR
- Whenever the input signal goes beyond the ADC’s DR, a trigger signal is generated by using comparators
- The trigger then activates a direct voltage generator that adds to the input signal to bring it within the DR

**5. User Interface**

**6. Hardware Results**

<table>
<thead>
<tr>
<th></th>
<th>$B^2R^2$</th>
<th>LASSO-$B^2R^2$</th>
<th>LASSO-$B^2R^2$ With Extra-Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low Sampling Rate</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>Robust to Noise</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Computational Time</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

Conclusion:

- We proposed a fast and robust algorithm to recover the residual signal in modulo sampling
- We designed a hardware prototype to bring the proposed theory to practice