Imperial College London

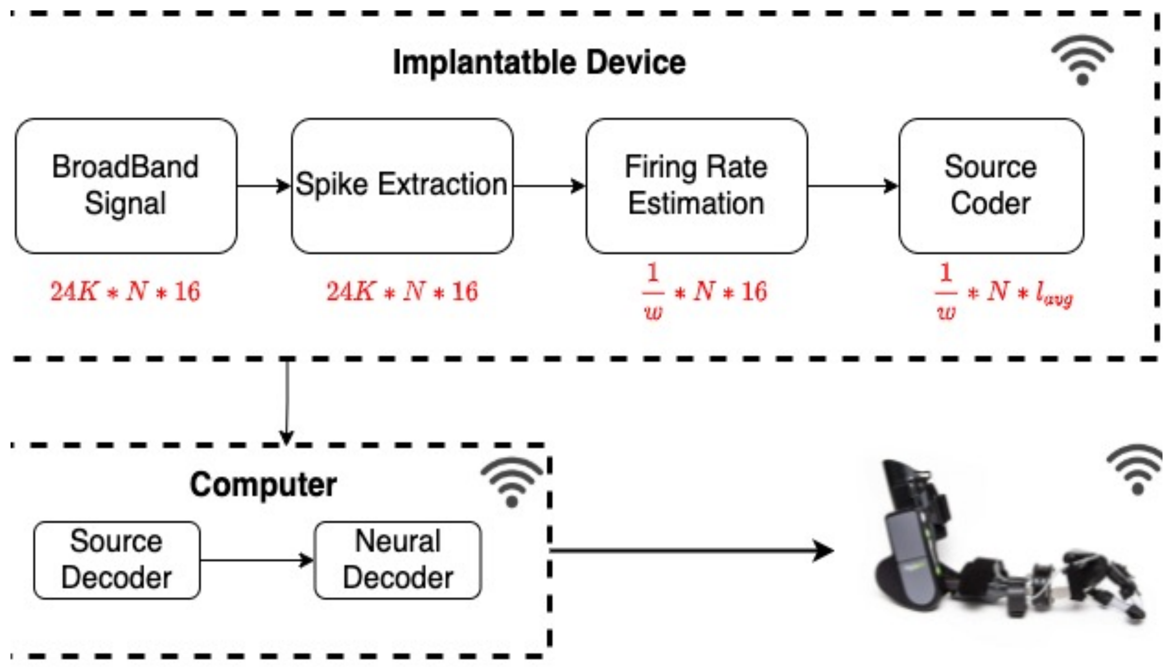
Brain-machine Interface Firing Rate Estimation

Department of EEE, Applied Machine Learning

Jingchao Zeng, Timothy G. Constandinou, Zhang Zheng, Savolainen Oscar

Introduction

With the development of wireless **brain-machine interface** (BMI) applications, neural signals with multiple channels are collected. The **multi-unit activity** (MUA) spike is a common signal for BMI decoding (Neuralink, BrainGate). Usually, the **high-sampling** MUAs are detected and transmitted wirelessly. The spikes are then estimated by the neural firing rate method and used for decoding on PC. Due to high transmitted rates in this **off-implant** approach, it is desirable to estimate the firing rate **on-implant**, **saving communication bandwidth.**

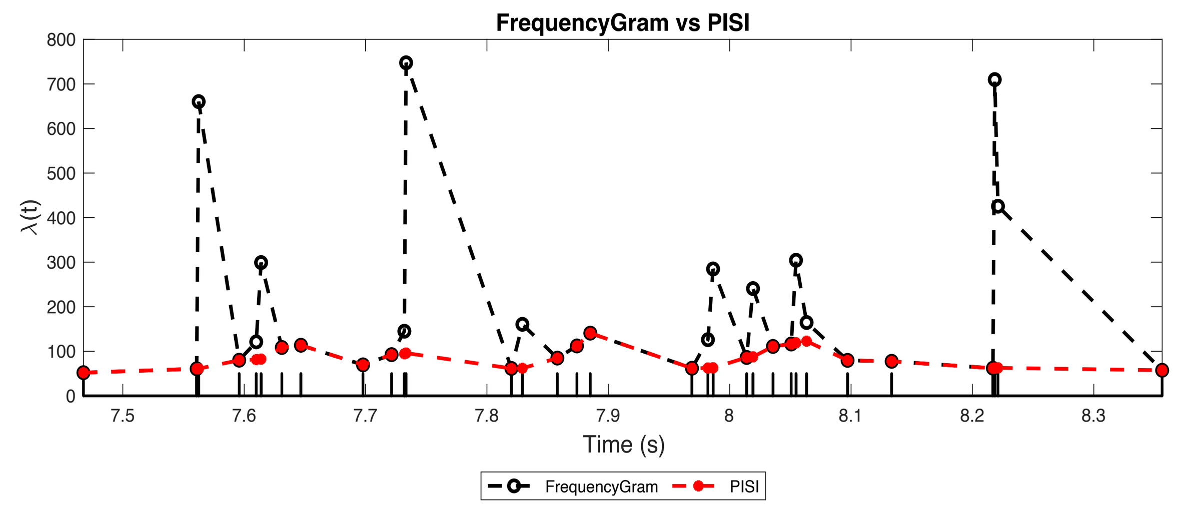


The goal for this project is to implement a hardware-friendly firing rate estimation method, balancing decoding performance, data rate and power consumption.

Methodology

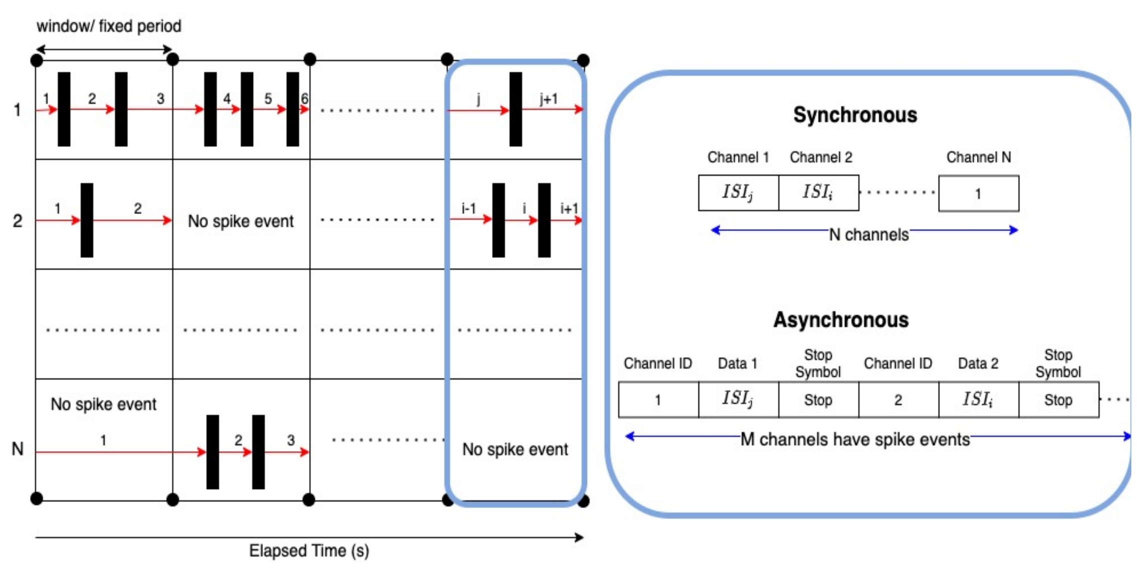
1) Penalised Interspike Interval (PISI)

- Theory: the ISIs indicate **how frequent neuron fires**
- Frequencygram method can capture the **abrupt change** of spike activities but **suffers from variability** of spike interval
- PISI **penalises this variability** by checking if difference with previous intervals is larger than threshold



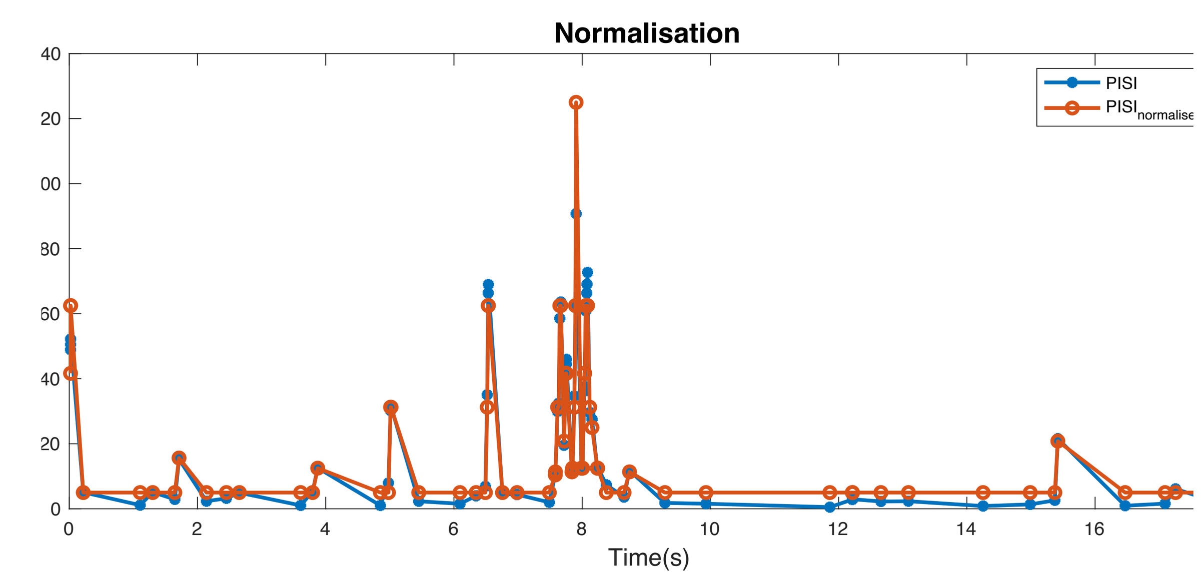
2) Data Organisation

- Sending data with **fixed sampling rate** $f_s = \frac{1}{window}$
- **Synchronous and asynchronous** organisation



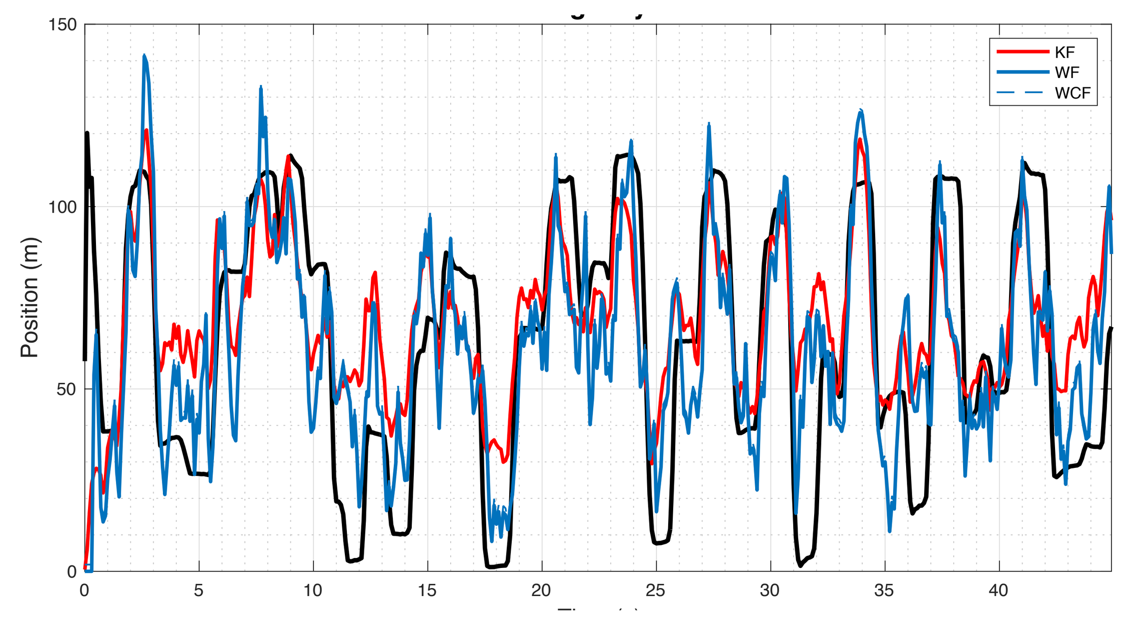
3) Data Compression

- Implement **Huffman and delta coder** to reduce the bit width of PISI estimates
- **Uniform normalisation** is proposed to reduce the dynamic range of estimates in Huffman encoding



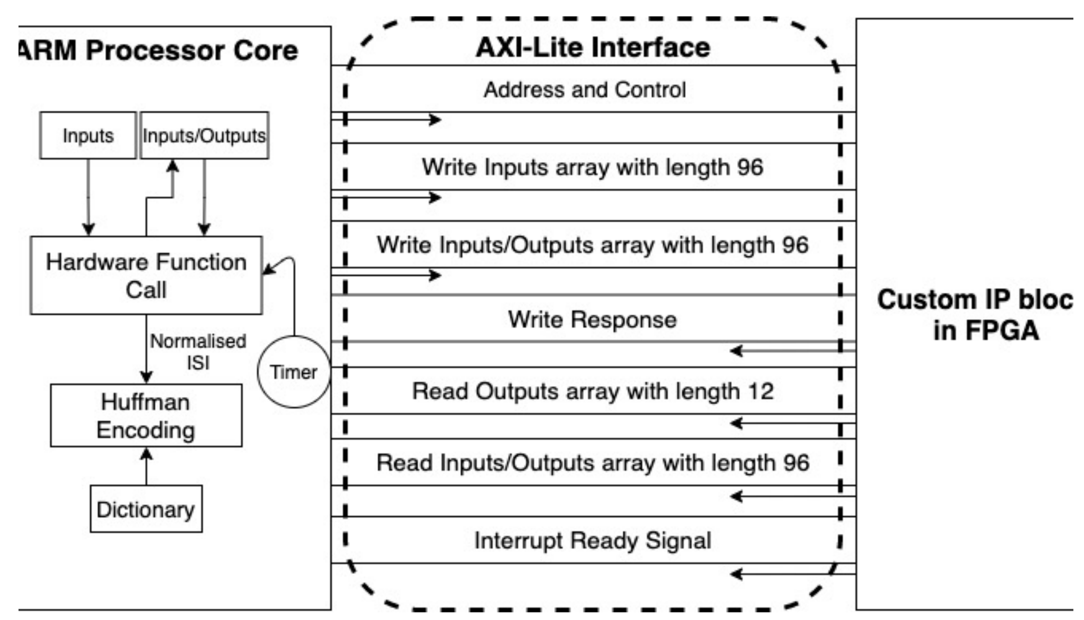
4) Neural Decoding

- The normalised estimates are used to predict 2-dimensional position and velocity in **WF, WCF and KF** decoders



5) Hardware Implementation

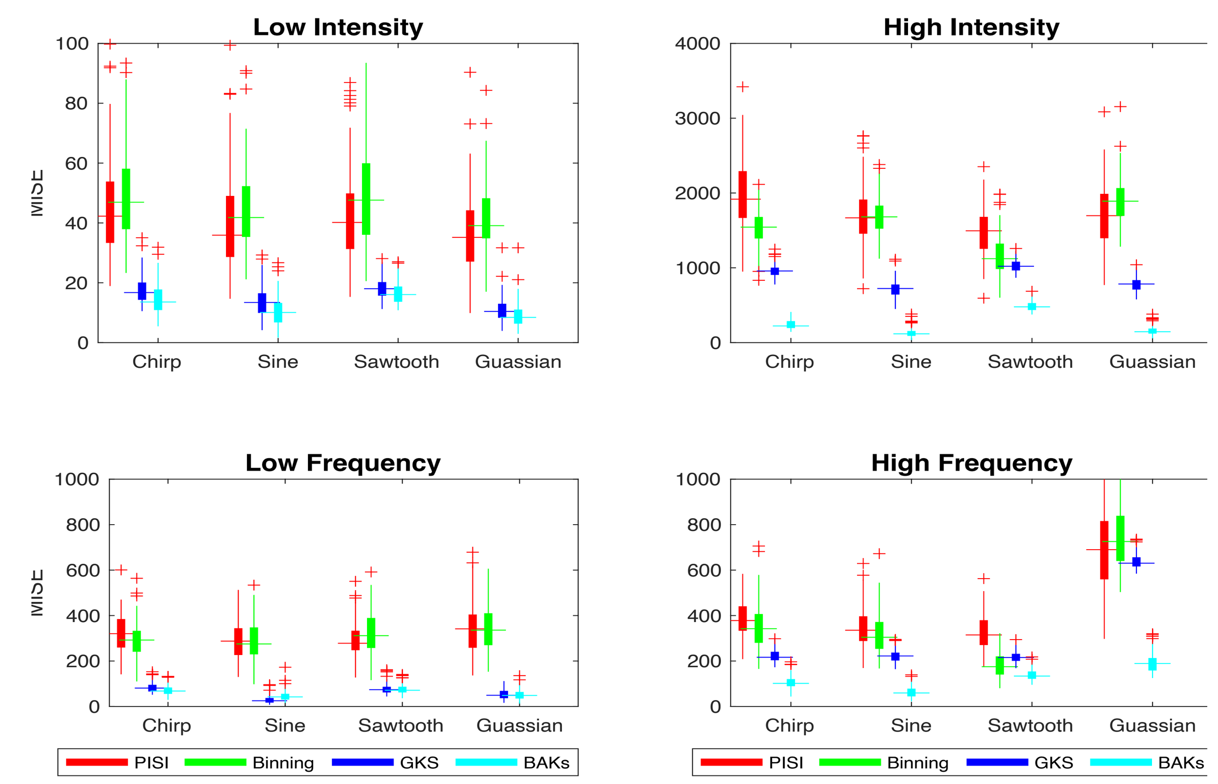
- **Create IP core:**
  - Fixed-point optimisation (dynamic bit width + bit shifting)
  - Throughput optimisation (pipeline + unroll + partition)
- **Communicate with IP core in Xilinx FPGA:**
  - Hardware synthesis (timing, power, resource usage)
  - Debug and program (Uart)



Results & Discussion

1) Synthetic datasets

- Generate underlying rate function to compare PISI with other state-of-the-art methods
- PISI performs better in spike trains with **low intensity and frequency** compared with binning



2) Real-world Datasets

- Huffman encoding with normalisation function can improve coding efficiency by four times (H = 1.45, l_{avg}= 2.18)
- Asynchronous organisation outperforms synchronous one in a tiny window (w <= 125 ms)
- Data rates are significantly reduced from 16*24k bits/s/channel to 2.18 * f_s bits/s/channel
- PISI estimates can achieve better decoding performance on position (CC = 0.72)
- Compared with binning, PISI consumes lower power and high throughput

Type	w	Latency	BRAM	FF	LUT	Power	Pos	Vel
PISI	0.2	400	6	436	664	21	0.72	0.62
Bin	0.2	620	7	427	699	23	0.66	0.68

Conclusion

- **Summary:**
  - PISI estimate accurately in **rare spike events**
  - PISI only requires around **1 kbps** data rate in 96 channels to achieves high decoding accuracy **CC = 0.65 on average**
  - PISI consumes **21 mW** and takes **400 clock cycles**
- **Future work:**
  - Hardware Implementation of the asynchronous compression in PISI method
  - Implement machine learning models to further improve decoding performance