An Energy-Efficient Kalman Filter Architecture with Tunable Accuracy for Brain-Computer Interfaces

Guy Eichler, Joseph Zuckerman, Luca Carloni

Columbia University

CS@CU Computer Science

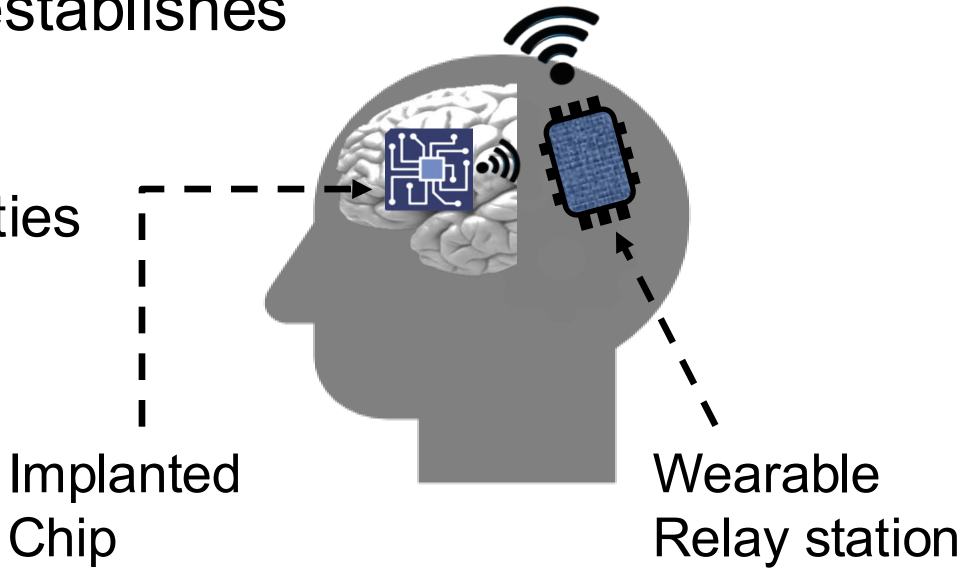
The Chips to Systems Conference 62

Sponsored by CEDA sigda

Brain Computer Interface (BCI)

A Brain-Computer Interface (BCI) is a system that establishes a connection between the brain and the outside world

- **The goal is to improve quality of life** – aid with disabilities
- **Real-time computation** within the boundaries of the reaction time of the brain (~0.18 seconds)
- **Meet low-power constraints** for wearable devices in the body-area network (~200 mW)



Our Focus: predicting motion from neural data recordings

Kalman Filter (KF)

One of the most popular algorithms for motion prediction in BCI

- Returns a motion prediction at each time step
- **Main bottleneck** is the computation of a matrix inverse

Existing implementations of the KF – **not** tailored to process neural data under the constraints of real-world BCI systems

function KALMAN_FILTER$(F, Q, H, R, \vec{x}_{n-1}, P_{n-1}, \vec{z}_n)$
  //Predict
  $\vec{x}_n = F \cdot \vec{x}_{n-1}$
  $P_n = F \cdot P_{n-1} \cdot F^t + Q$
  //Update
  $\vec{y} = \vec{z}_n - (H \cdot \vec{x}_n)$ //Innovation
  $S = H \cdot P_n \cdot H^t + R$
  $\boxed{K = P_n \cdot H^t \cdot S^{-1}}$ //Compute Kalman Gain
  $\vec{x}_n = \vec{x}_n + K \cdot \vec{y}$
  $P_n = (I - K \cdot H) \cdot P_n$
  return $\vec{x}_n, P_n$

Comparing Matrix Inverse Approaches

- Gaussian elimination (Gauss) is the standard method to **calculate** the matrix inverse
  - Floating-point divisions and Internal data dependencies → not ideal for hardware
- **We want to accelerate the KF with specialized hardware**
- We can **approximate** the matrix inverse
  - With reasonable error (~10%)
- **Integration of different methods inside the KF** – tested with animal brain data
  - Newton-Raphson method (Newton) provides the best results

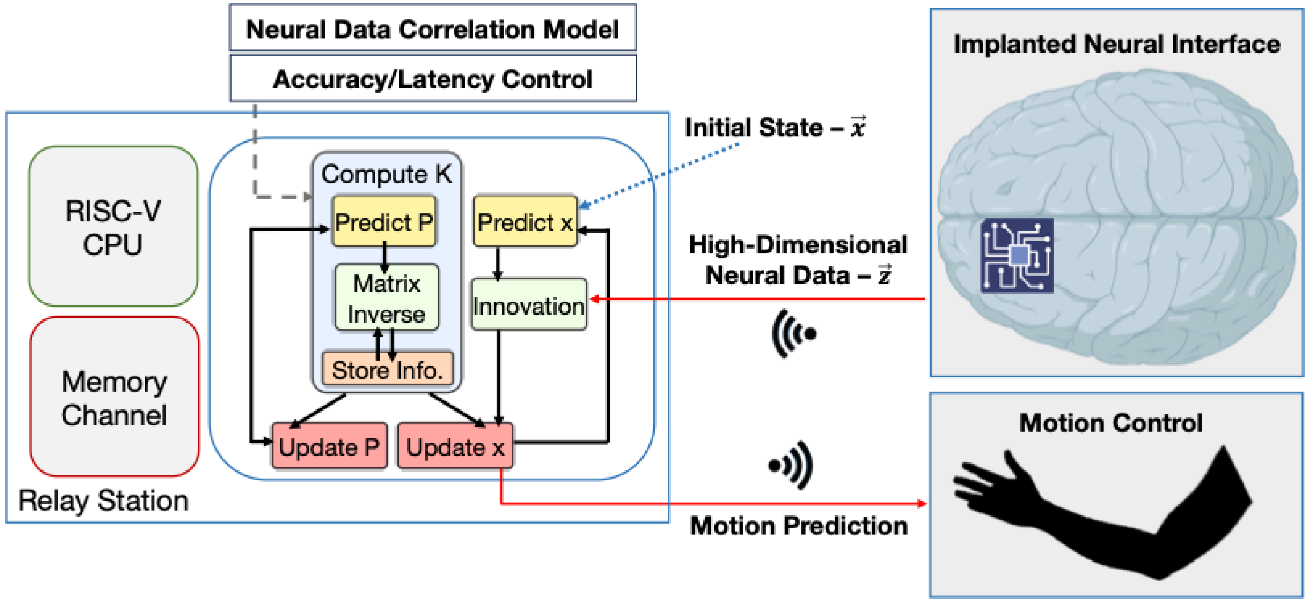
Accuracy Metric	Gauss [50]	IFKF [23]	Taylor [22]	SSKF [31]	Newton [29]
MSE	3.8×10^{-12}	53.8	0.05	0.1	6.6×10^{-6}
MAE	7×10^{-7}	2.7	0.08	0.06	0.0004
*Max. Difference (%)	0.008	2.2×10^{4}	9.7×10^{2}	5.3×10^{2}	4
*Avg. Difference (%)	0.0001	350	9	4.8	0.035

*These scores are normalized with respect to the KF output from [Glaser et al. eNeuro 2020]

KalmMind

A framework and architecture for the agile development and design-space exploration of configurable KF hardware accelerators specialized for BCI

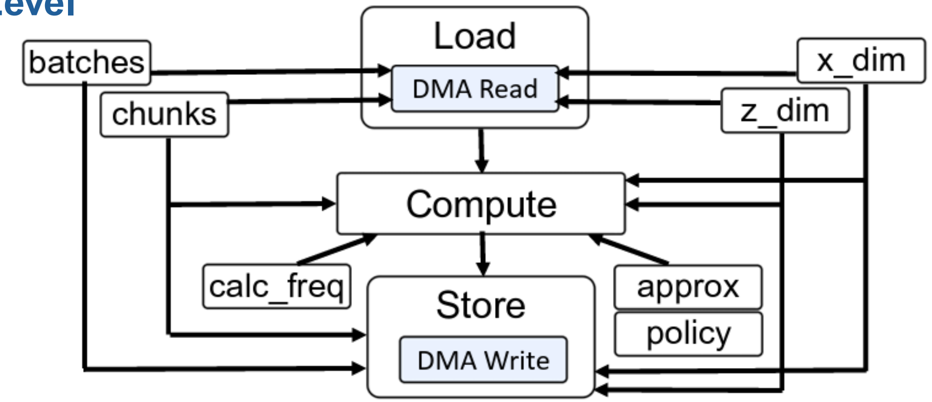
- A new **algorithm-hardware technique** that uses approximations
- A unique feature of **tunable accuracy/latency/energy efficiency**
- Leveraging **spatiotemporal correlation of neural data**
- High configurability for **diverse neural data**



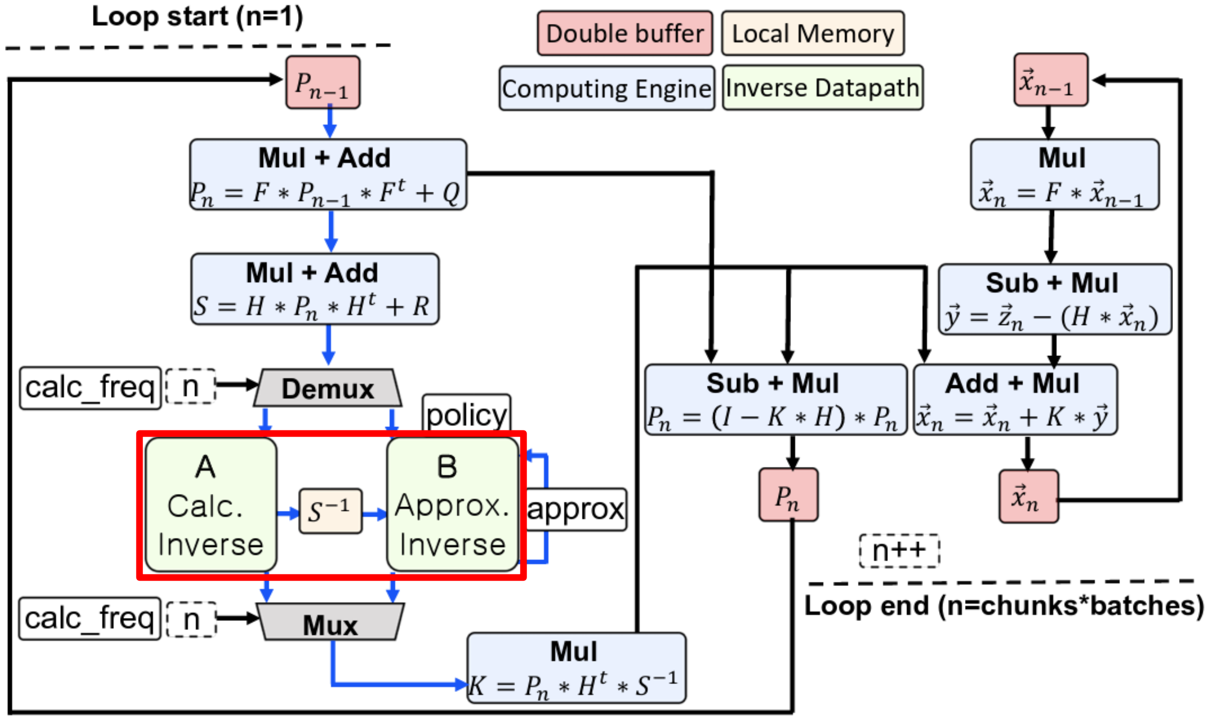
Configurable Accelerator Architecture

Create a KalmMind-based hardware accelerator!

Top Level

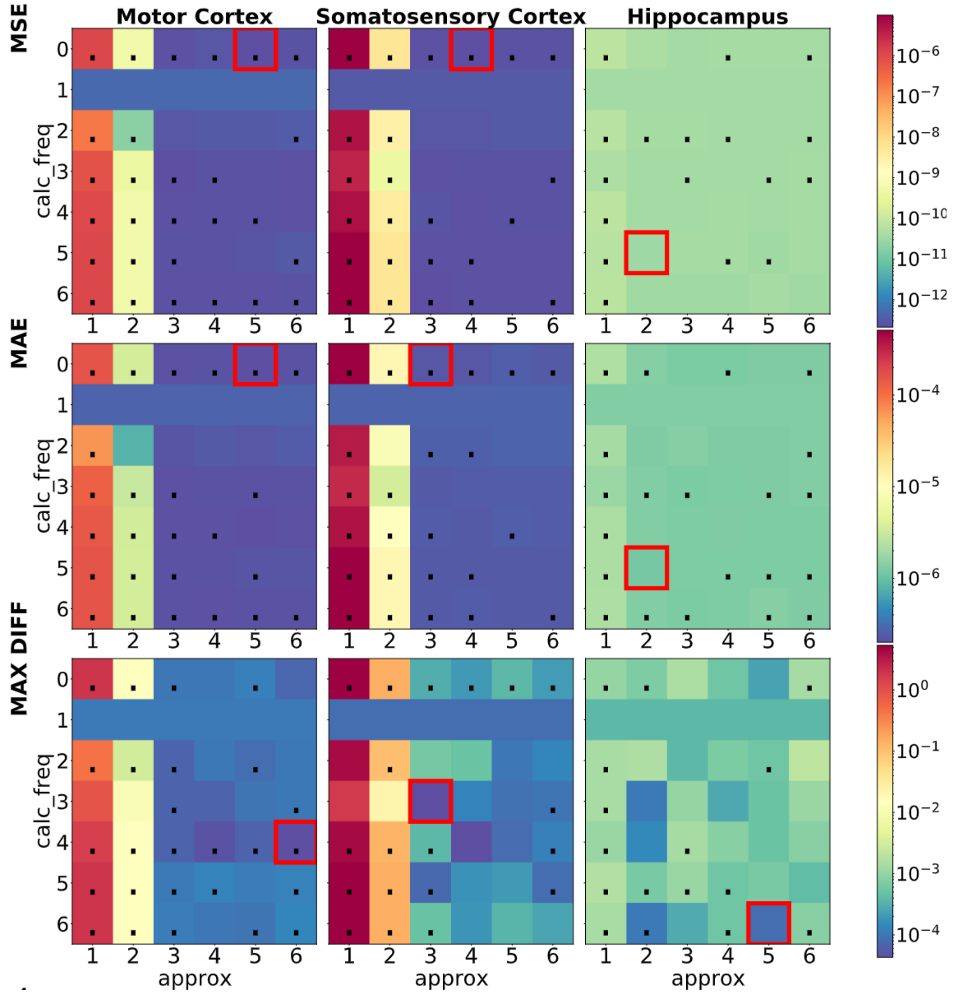


Compute Level



- **Combination of matrix inverse approximation and calculation**
  - Approximations are efficient but less accurate
  - Calculations are more accurate but less efficient
- Configuration for interleaving **calculation/approximation patterns**

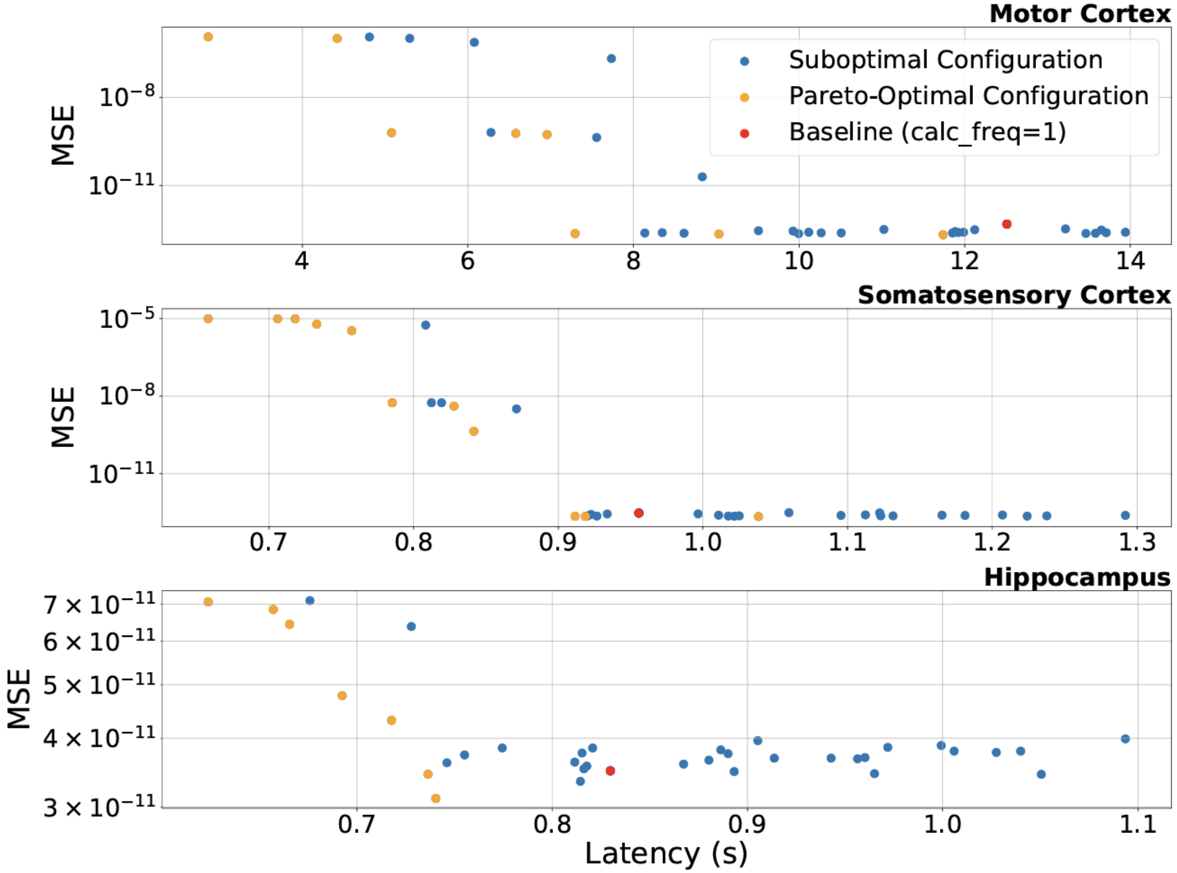
Results – Accuracy Analysis



	MSE	MAE	MAX DIFF
Motor	$2.1\times10^{-13}-1.1\times10^{-6}$	$2\times10^{-7}-1.6\times10^{-4}$	$4.3\times10^{-5}-1.91$
Soma.	$2.2\times10^{-13}-9.9\times10^{-6}$	$2.3\times10^{-7}-5.1\times10^{-4}$	$3.5\times10^{-5}-5.3$
Hippo.	$3.1\times10^{-11}-7.1\times10^{-11}$	$1.2\times10^{-6}-2.2\times10^{-6}$	$8.2\times10^{-5}-2.1\times10^{-3}$
Baseline	4.8×10^{-13}, 3×10^{-13}, 3.5×10^{-11}	2.7×10^{-7}, 2.7×10^{-7}, 1.4×10^{-6}	1.1×10^{-4}, 8.5×10^{-5}, 3.8×10^{-4}

- Three animal neural datasets
- Configuring different interleaving patterns between **Gauss calculation/Newton approximation**
- **Wide accuracy range** for each neural datasets
- **Better accuracy than the baseline** (only Gauss)
- Up to **78% better accuracy**

Results – Accuracy vs. Latency

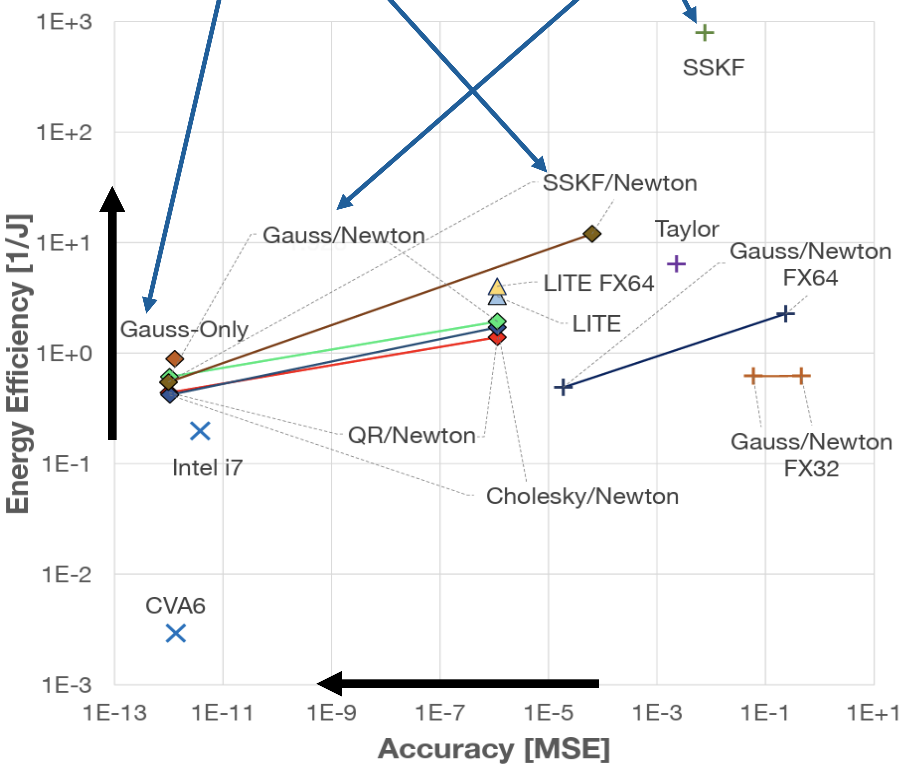


- **Full SoC integration** of the Gauss/Newton accelerator and a RISC-V CVA6 CPU using **ESP**
- **Experiments on FPGA** using custom Linux-based software applications
- **Design-space exploration –** accuracy vs. latency
- Up to **55% better accuracy and 4.4x speedup**

Results – Accuracy vs. Energy Efficiency

SSKF/Newton achieves **15.3x better energy efficiency** than Gauss-only

Gauss/Newton achieves 10^9x **better accuracy** than SSKF



Type	Method	LUT	FF	BRAM	DSP	Power [W]	Perf. [sec]	Energy [J]	Accuracy [MSE]
Software	Intel i7	N/A	N/A	N/A	N/A	78.6	0.065	5.1	3.8×10^{-12}
	CVA6	43996	29922	36	27	0.177	1927	341	1.3×10^{-12}
Hardware: Calc./Approx. Datapath	Gauss/Newton	22119	18725	228	252	0.185	2.8–8.9	0.52–1.64	1.03×10^{-12}–1.1×10^{-6}
	Cholesky/Newton	22429	20126	360	268	0.207	2.8–11.5	0.58–2.38	1.05×10^{-12}–1.1×10^{-6}
	QR/Newton	24842	21259	385	258	0.236	3.04–9.6	0.72–2.27	1.02×10^{-12}–1.1×10^{-6}
	Gauss/Newton FX32	19646	12131	195.5	217	0.146	4.25	0.354	5.9×10^{-2}–0.46
	Gauss/Newton FX64	34831	26109	369	534	0.18	2.44–11.3	0.44–2.04	1.9×10^{-5}–0.24
Hardware: One-way Datapath	LITE	15591	13405	146.5	193	0.114	2.688	0.306	1.14×10^{-6}
	LITE FX64	14782	8075	267	347	0.11	2.268	0.249	1.14×10^{-6}
	SSKF/Newton	18798	16961	204.5	240	0.158	0.53–11.6	0.08–1.82	9.9×10^{-13}–6.3×10^{-5}
	SSKF	8403	6752	19.5	102	0.051	0.03	0.0015	7.63×10^{-3}
	Taylor	15006	13437	118	230	0.155	1.203	0.186	2.3×10^{-3}
	Gauss-Only	12386	10290	102.5	153	0.098	12.507	1.225	1.3×10^{-12}

A variety of KalmMind-based accelerators – different calculation and approximation methods

- All accelerators (except Gauss-only) completed 100 KF iterations in under 5 seconds – **real-time**, consuming up to ~200mW **– low-power**
- Wide ranges of accuracy and energy efficiency
- **Clear advantages over the state-of-the-art**

Conclusion

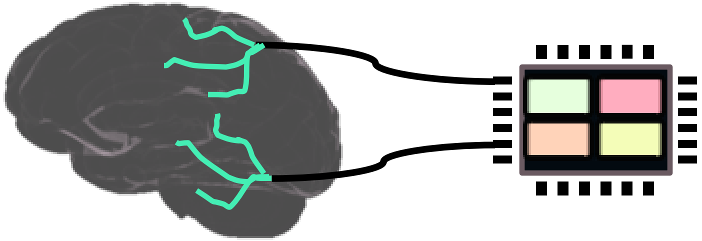
- **BCI applications benefit from algorithm-hardware co-design!**
- KalmMind provides **the first architecture to facilitate the design of configurable KF hardware accelerators for BCI**, offering flexibility and uniquely supporting fine-grained control over latency and accuracy to address the diversity of brain data.
- The goal of our work is to advance research on hardware architectures for embedded BCIs and to accelerate the development of **practical, real-world BCI systems and applications.**
- KalmMind can be extended for other application domains using the Kalman filter.
- The contributions of this work have been released to the public domain: https://github.com/GuyEichler/KalmMind

Questions?

Guy Eichler

Columbia University, PhD

guyeichler@cs.columbia.edu



62

THE CHIPS TO SYSTEMS CONFERENCE

SPONSORED BY CEDA sigda