

UCLA

Posters

Title

Collecting High-Rate Data Over Low-Rate Sensor Network Radios

Permalink

<https://escholarship.org/uc/item/6dq7q4zh>

Authors

Ben Greenstein
Alex Pesterev
Christopher Mar
et al.

Publication Date

2005



Collecting High-Rate Data Over Low-Rate Sensor Network Radios

Ben Greenstein, Alex Pesterev, Chris Mar, Eddie Kohler, Jack Judy, Shahin Farshchi, Deborah Estrin
CENS Systems Lab – <http://lecs.cs.ucla.edu>

Introduction: Collecting High-Rate Data with Motes is Hard!

Acquisition Challenges

- High interrupt load interferes with other application subsystems
- Delayed interrupt handling induces sampling jitter
- Limited memory forces a quick turnaround (at 8kHz, cannot store 1 second of data in RAM)

Communication Challenges

- Shared radio channel cannot support aggregate data traffic produced by neighborhood
- Lack of flow/congestion control overwhelms network
- Limited RAM shortens forwarding queues
- Data traffic collides with data processing tasking messages

Data Processing Challenges

- High-rate data must be processed before transmission
 - When a system is bandwidth limited, better to plan which sets are lost rather than leave it to chance
- Motes provide limited support for signal processing
 - 8MHz MCU, 10KB RAM, no FPU
- Sensor networks have limited debugging visibility
- Heavy MCU load interferes with other subsystem tasks

Heterogeneity Challenges

- Mote/ puserver platform diversity complicates programming
- Collaborative processing among motes and puserservers difficult to coordinate

Problem: How do we process high-rate data to overcome bandwidth limitations?

Engineering Systems Need Tuning, Calibration

- Researchers don't know best way to filter data because they haven't seen such spatially dense data before
- Interesting data is application- and environment-dependent and time varying
- Spatial correlation is not always well understood

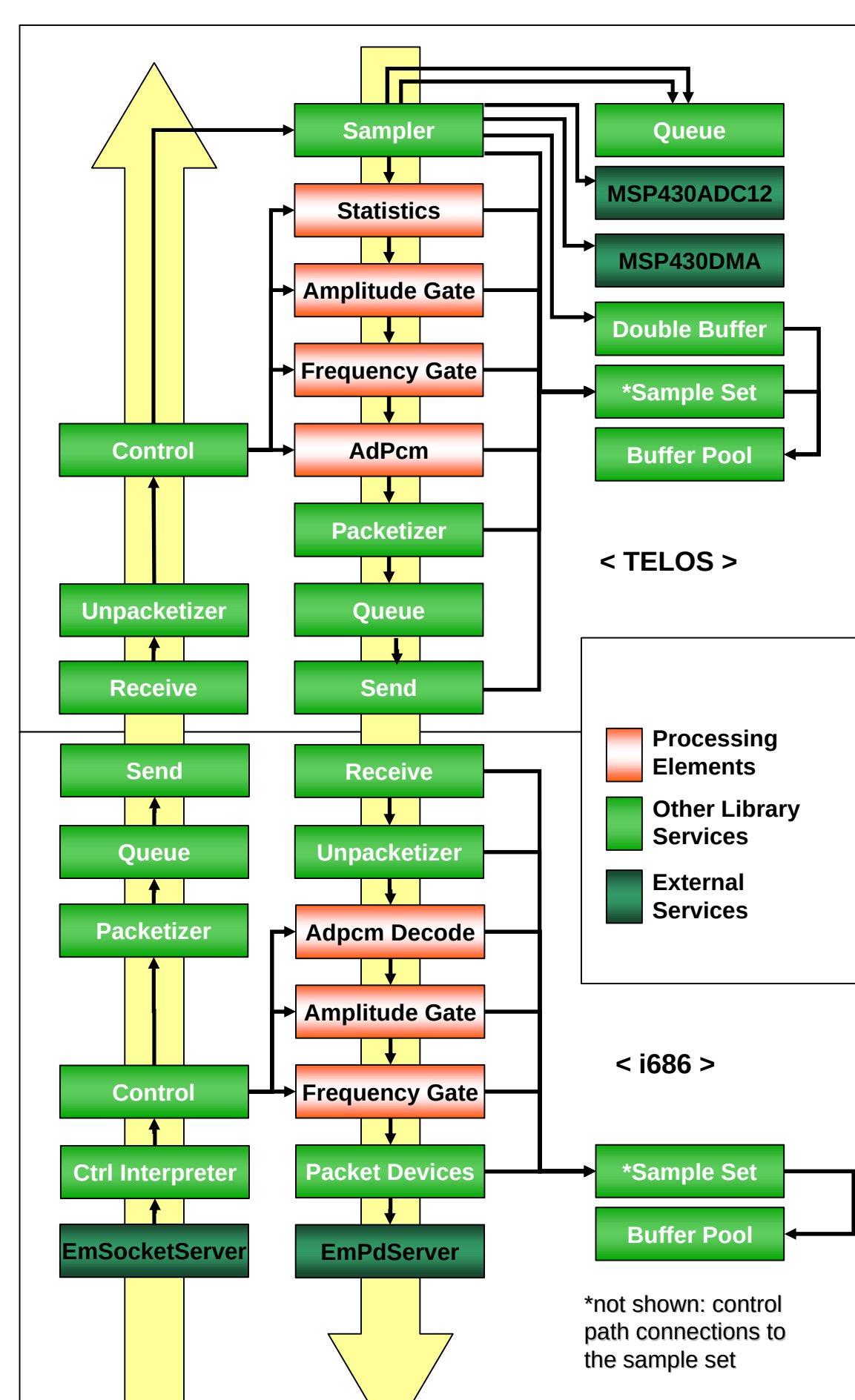
Interactive Control over Data Processing

- For calibration, hypothesis tests, and pattern searches, it's best to collect representative waveform data for back-end processing
- Given bandwidth limitations, best to transfer data processing to sensor nodes to return as much interesting information as possible

VanGo: Software Architecture for High-Rate Data Collection

Cross-Platform Software Architecture

- **Data Acquisition**
 - DMA transfers samples from TelosB's 12-bit ADC
 - Auricle, an acoustics application, uses a microphone
 - NeuroMote, a neural monitoring application, uses a bio-interface circuit connected directly to probes implanted in a rodent's head
- **Data Processing**
 - Measurement, classification, and compression operate on motes for communication efficiency, puserservers for calibration
 - Processing elements written in nesC, even when operating on a puserver
- **Multiple Hop Transmission**
 - Data collected using MultihopLQI, with modifications for forwarding high-rate data
 - Control messages transmitted using Drip, an efficient flooding protocol
- **Control**
 - Socket interface to enable and control processing elements
 - Syntax exposes filtering knobs directly

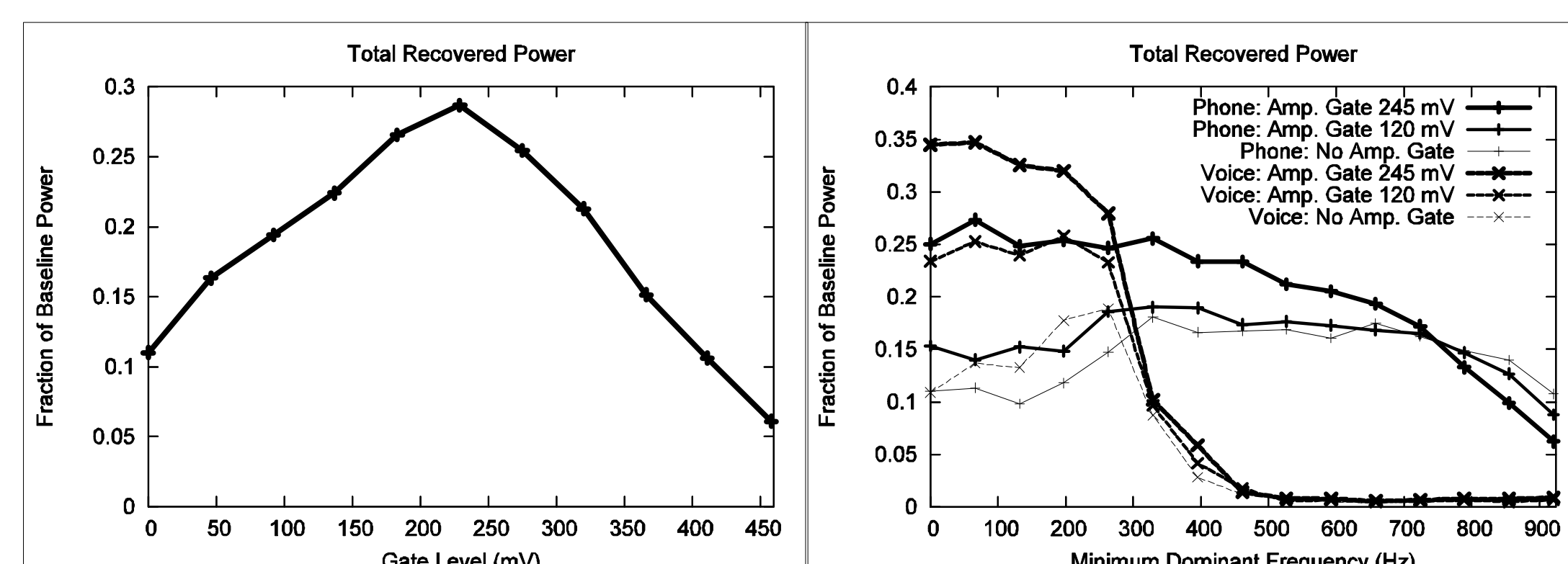


Processing Elements

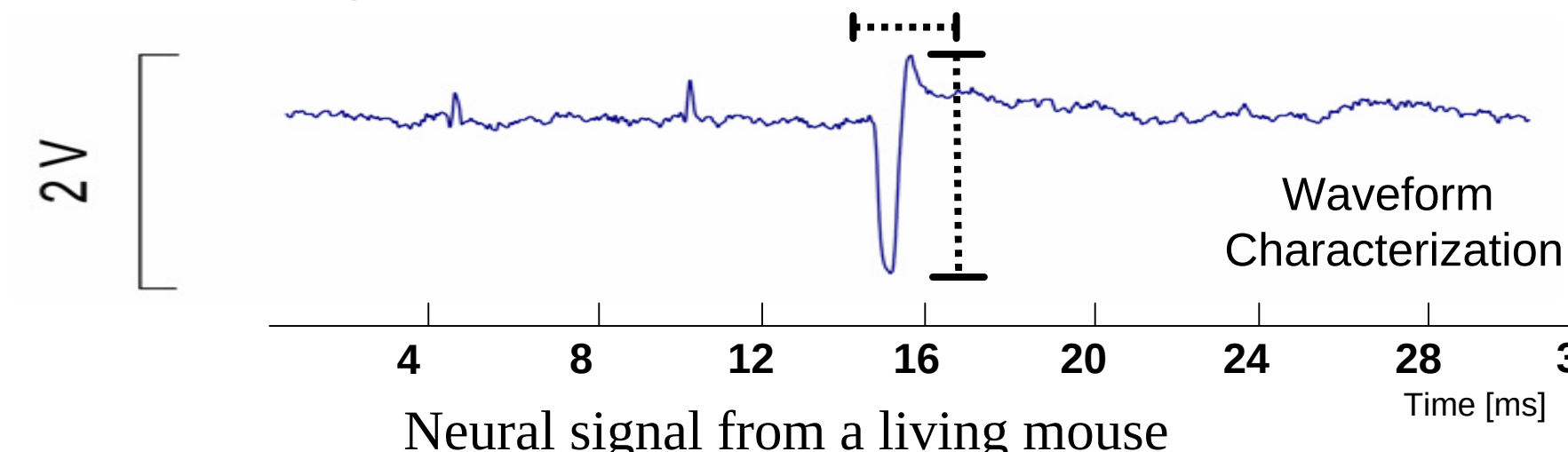
- **Measurement**
 - Statistics (mean, std. dev., mean dev.)
- **Classification**
 - Amplitude Gate
 - Dominant Frequency Gate
 - Spike Detection
- **Compression**
 - ADPCM (3:1, lossy)
 - Format Conversions (pack, truncate)

Filter	Execution Time (ms)
ADPCM	8.345
Amplitude Gate	0.628
with summarization	0.638
Dominant Frequency Gate	0.776
with summarization	0.786
Statistics	3.790
Format	0.529
Fragment	1.010
SpikeDetector	0.860
with compression	1.017

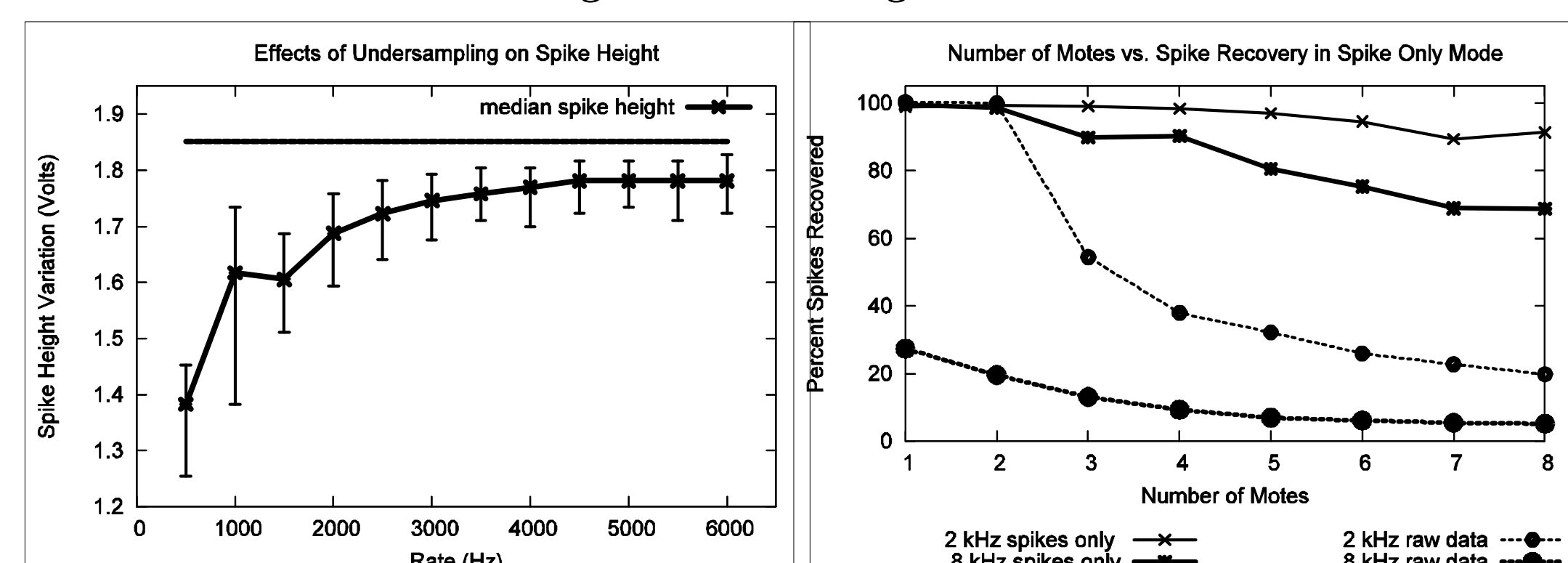
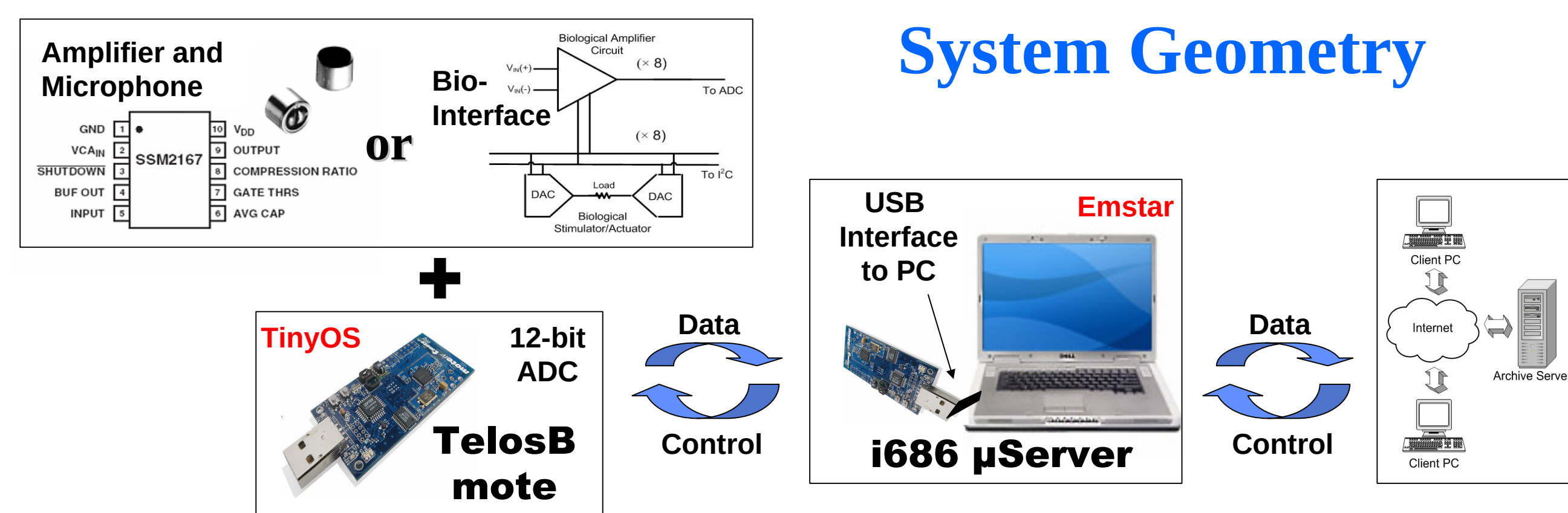
Auricle (acoustics application)



NeuroMote* (neural signal application)



System Geometry



*A collaboration with UCLA Neuroengineering, Paul Nuyujukian and Neschae Fernando

