High Rate Machine Learning for Forecasting Time-Series Signals

UNIVERSITY OF ARKANSAS

Atiyehsadat Panahi
Ehsan Kabir
David Andrews
Miaoqing Huang

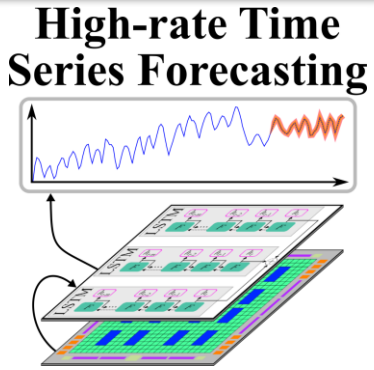
UNIVERSITY OF South Carolina

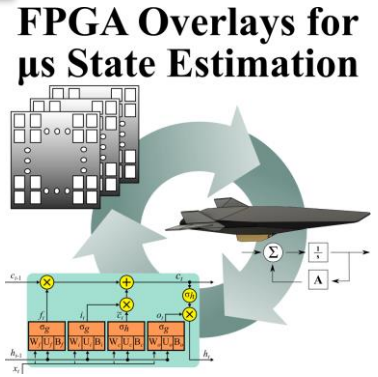
Austin Downey

ARTS-Lab Adaptive Real-Time Systems Laboratory

Jason D. Bakos

HeRC Heterogeneous and Reconfigurable Computing Group





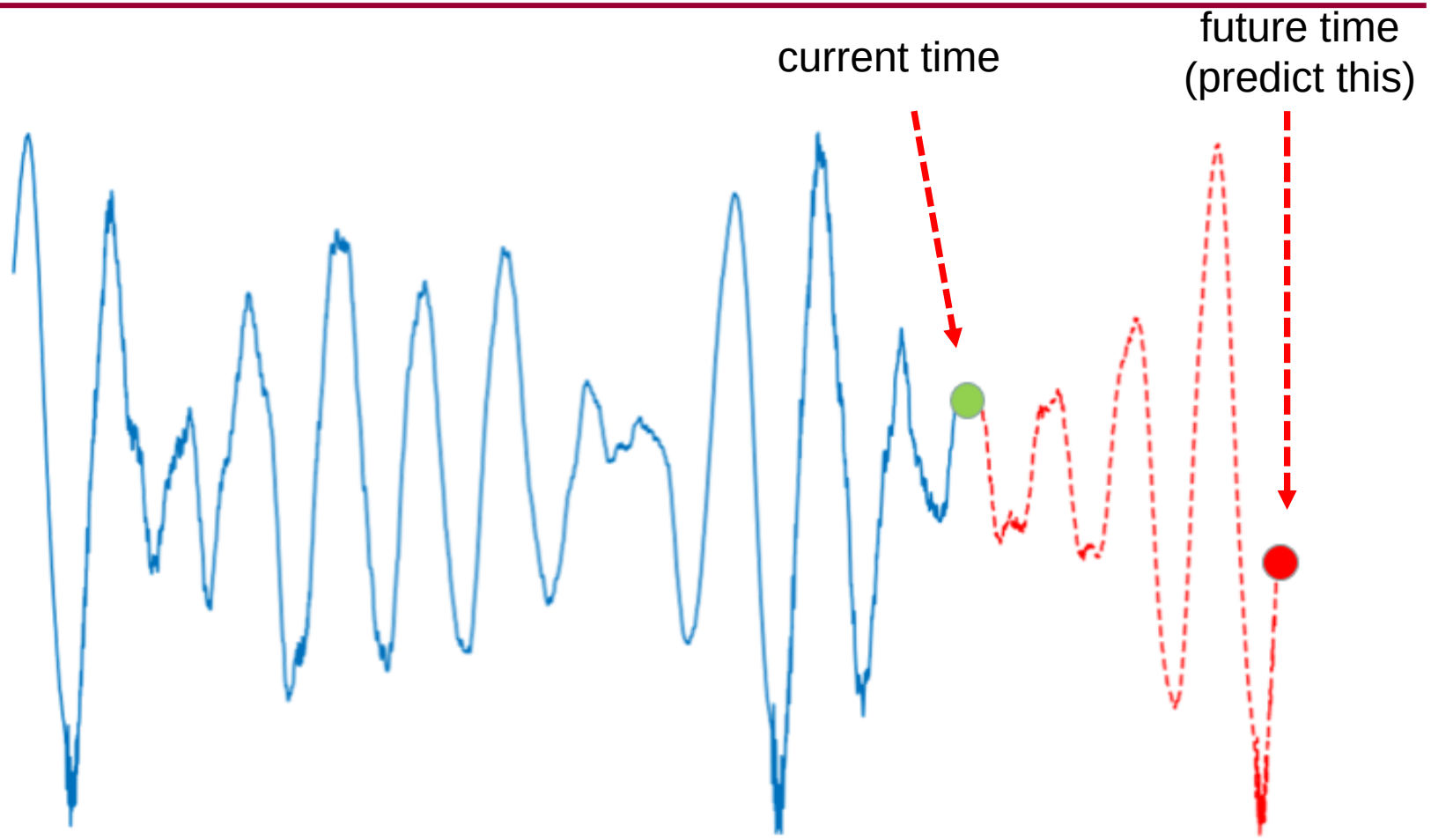
This material is based upon work supported by the National Science Foundation under Grant No. 1956071.

Objectives

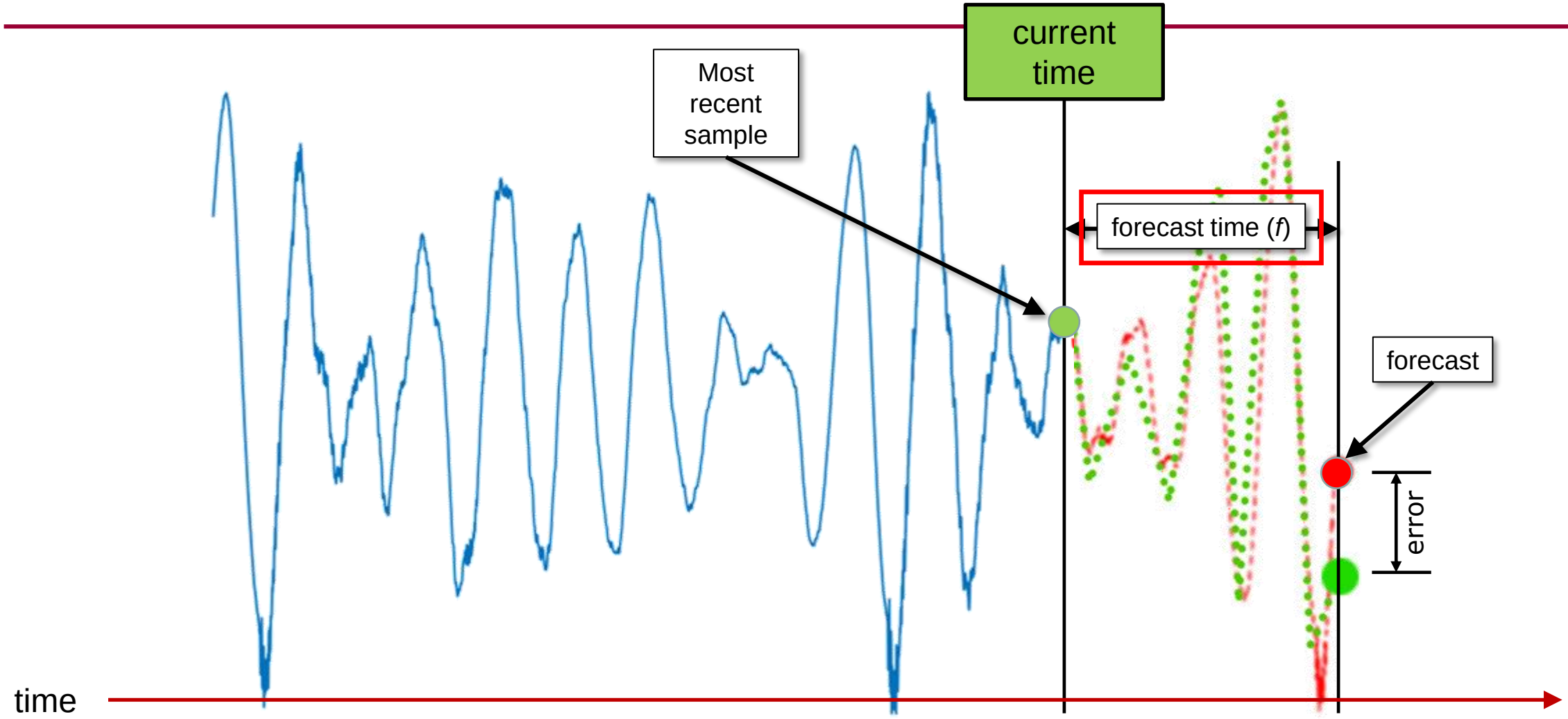
Objective:
Simultaneous forecast + learn for time series

Performance metrics:
1. Forecast accuracy
2. Re-training time
3. Latency

Contributions:
- Algorithms
- HLS-based implementation
- Overlay-based implementation

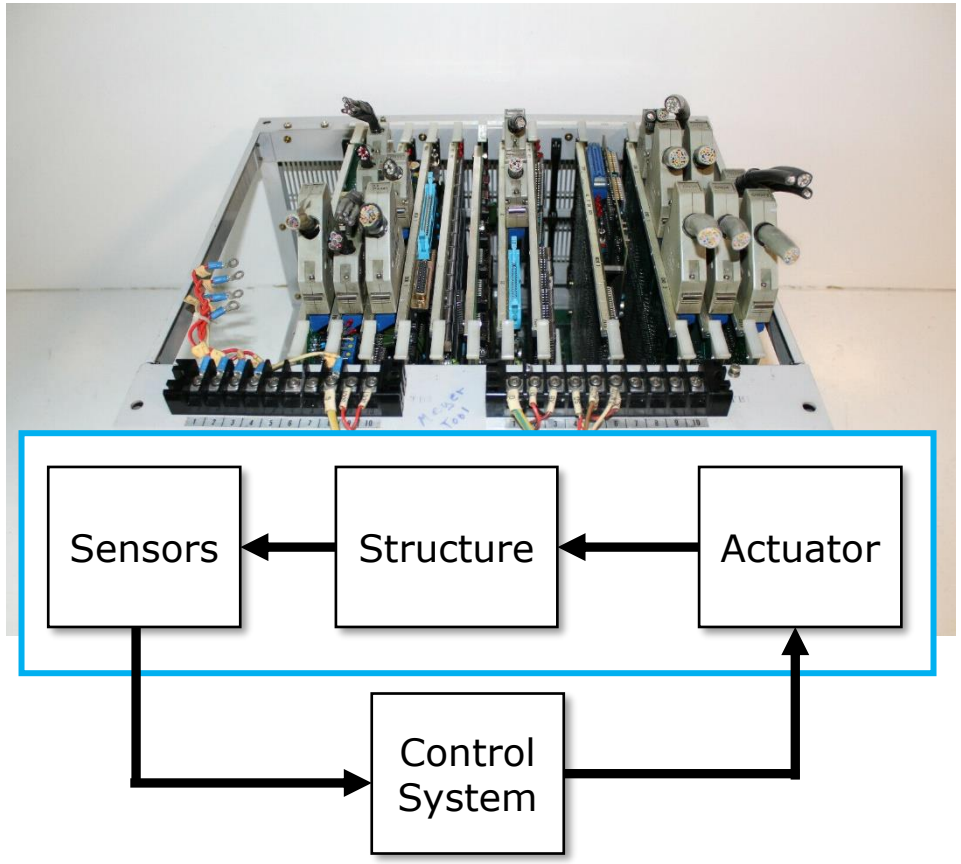


Time Series Forecasting

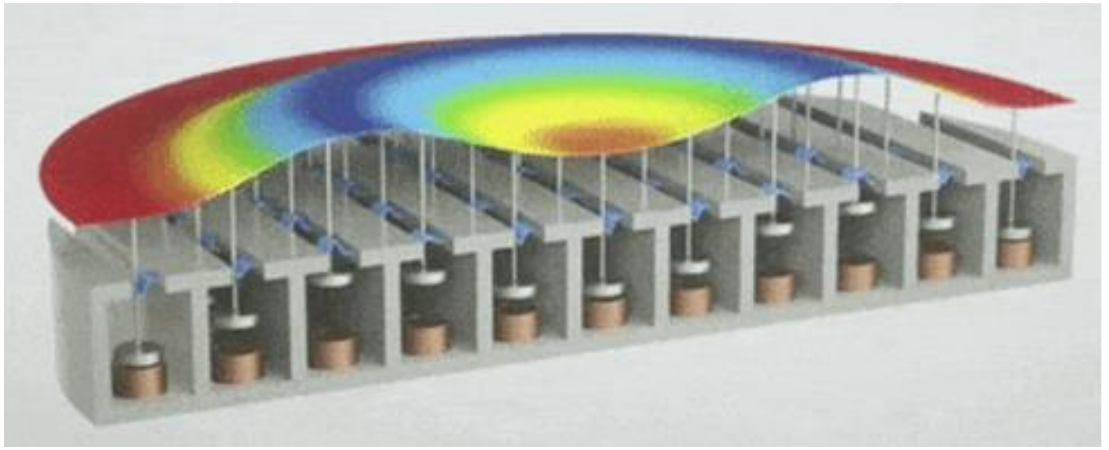


Applications: Control of Active Structures

- Active Vibration Control



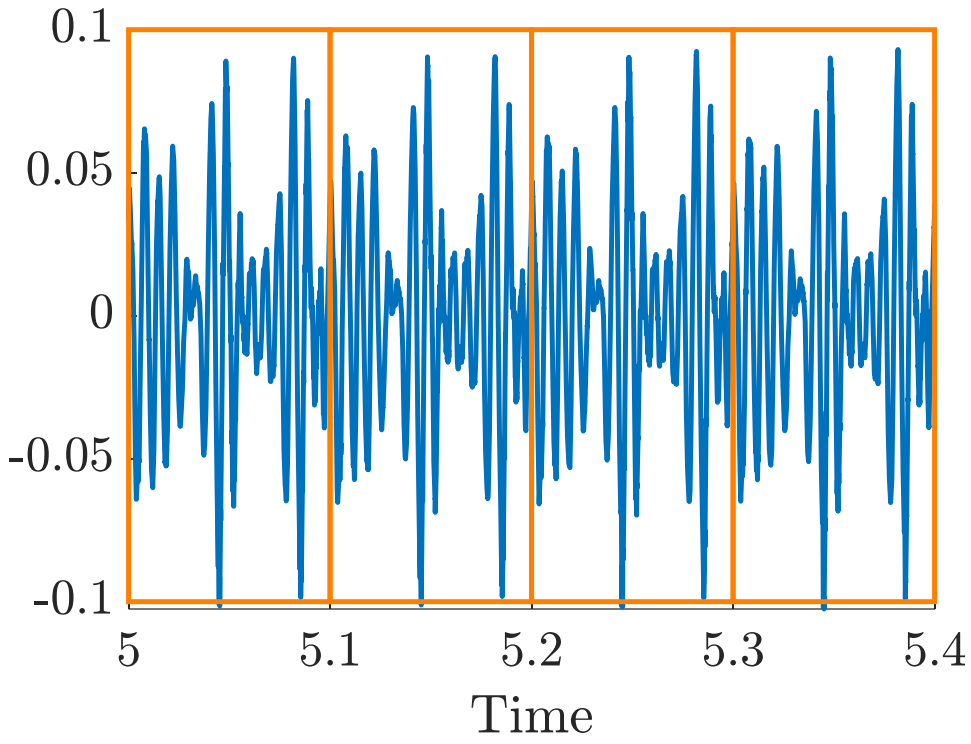
- Deformable Mirrors



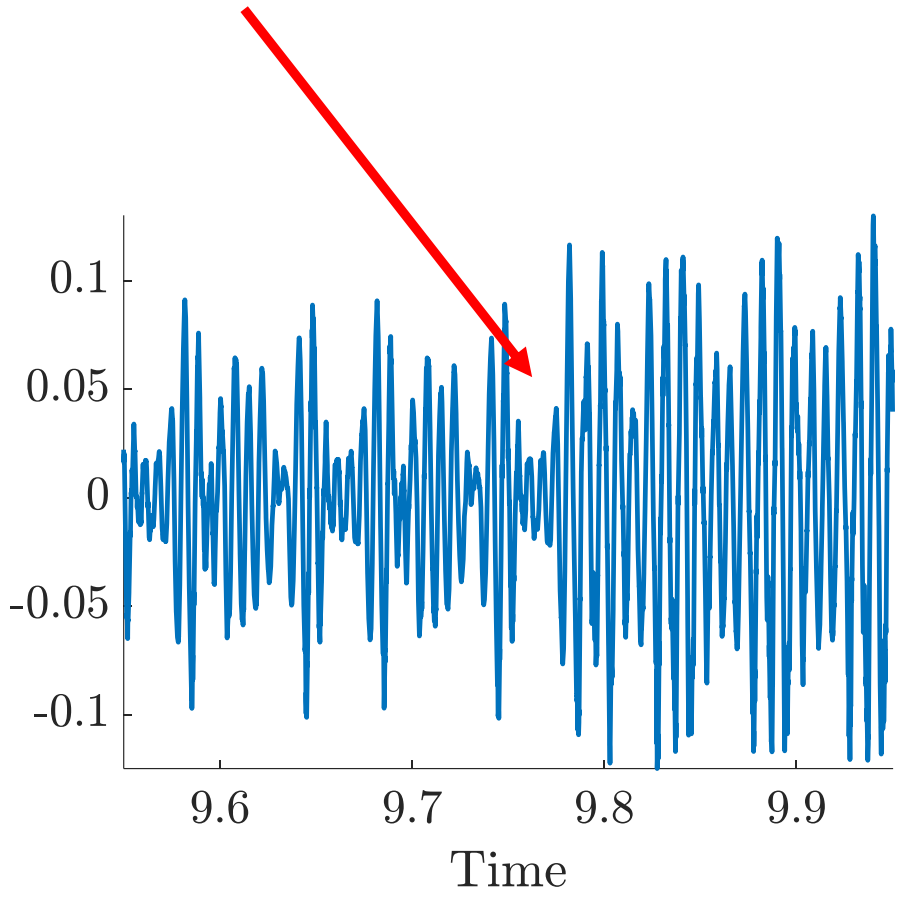
* ALPAO Corp.

Approach

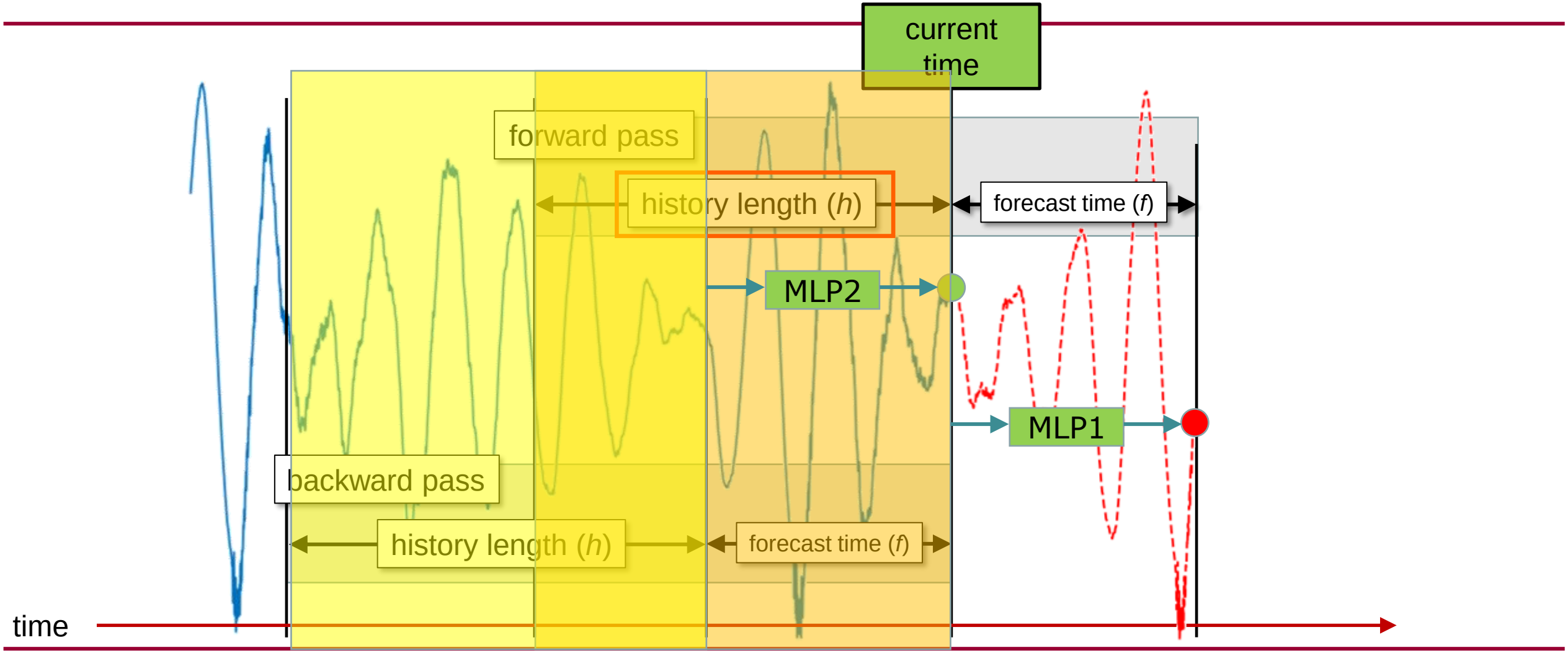
- Signal must be periodic
- Period unknown and may be too long for timely relearn



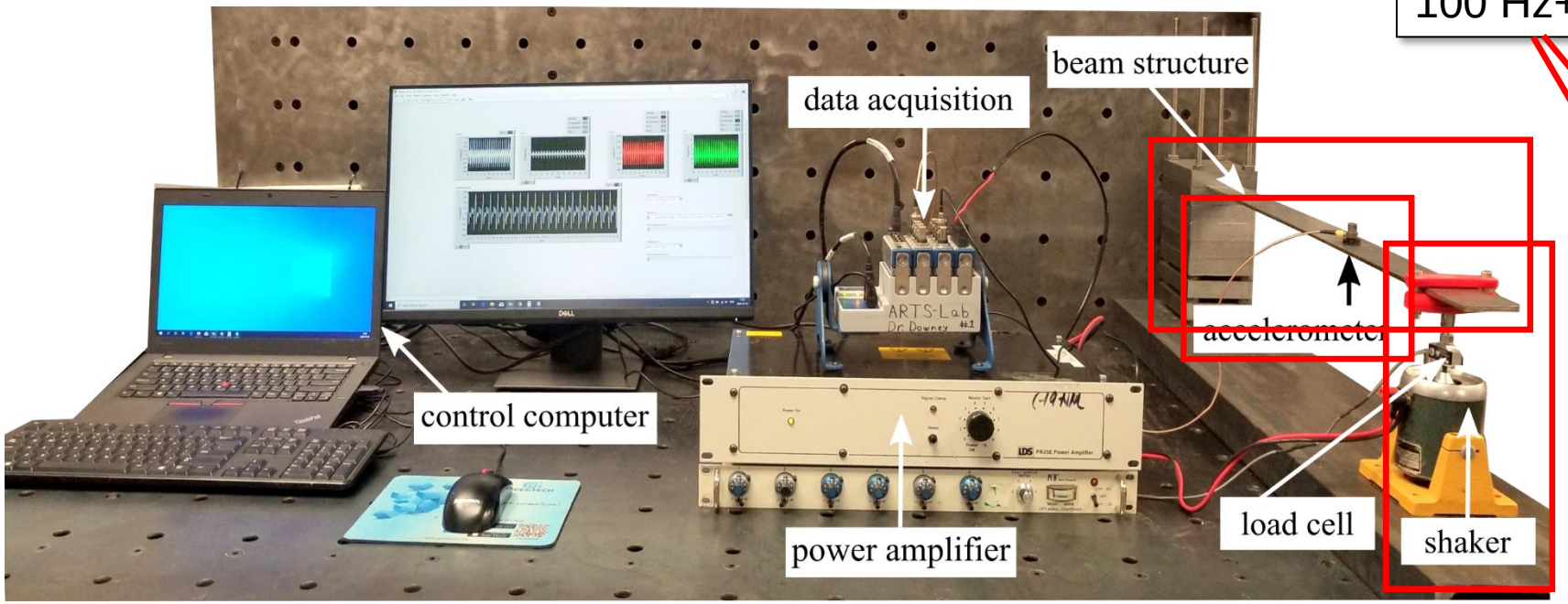
- Nonstationarity



Approach: MLP-Based Model



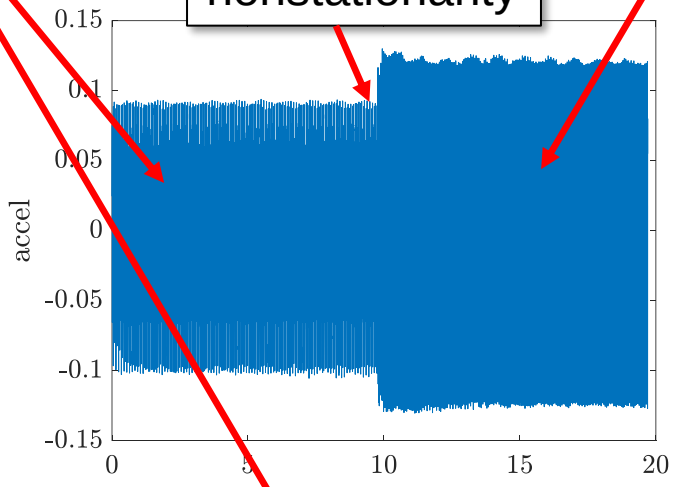
Test Data

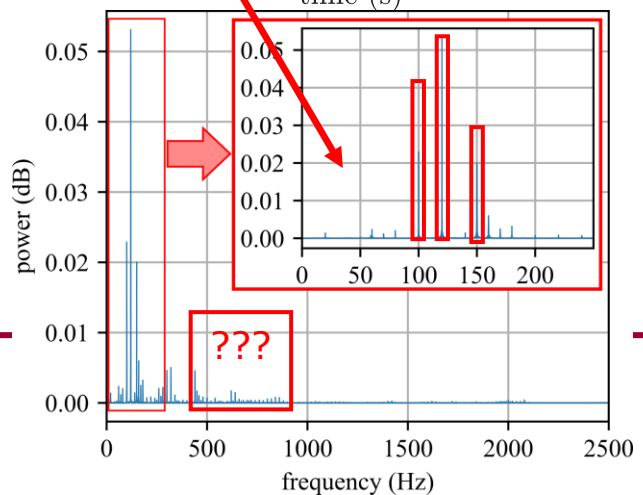


stimulus:
100 Hz+120 Hz+150 Hz

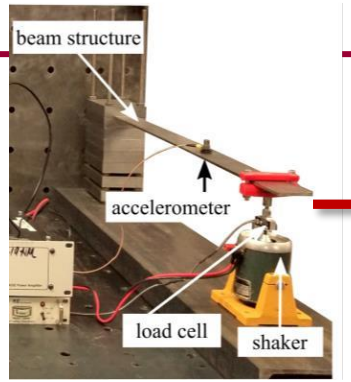
stimulus:
100 Hz+120 Hz

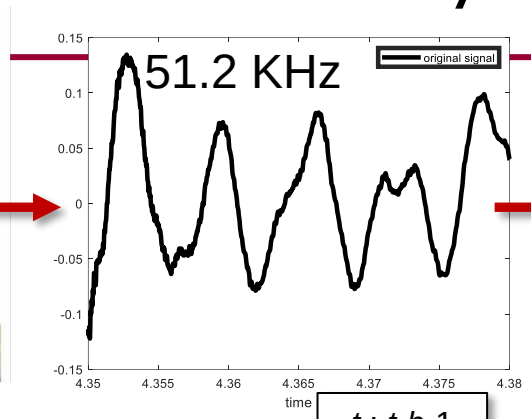
nonstationarity





System Design

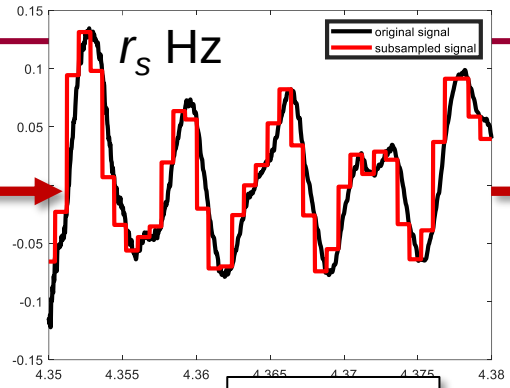


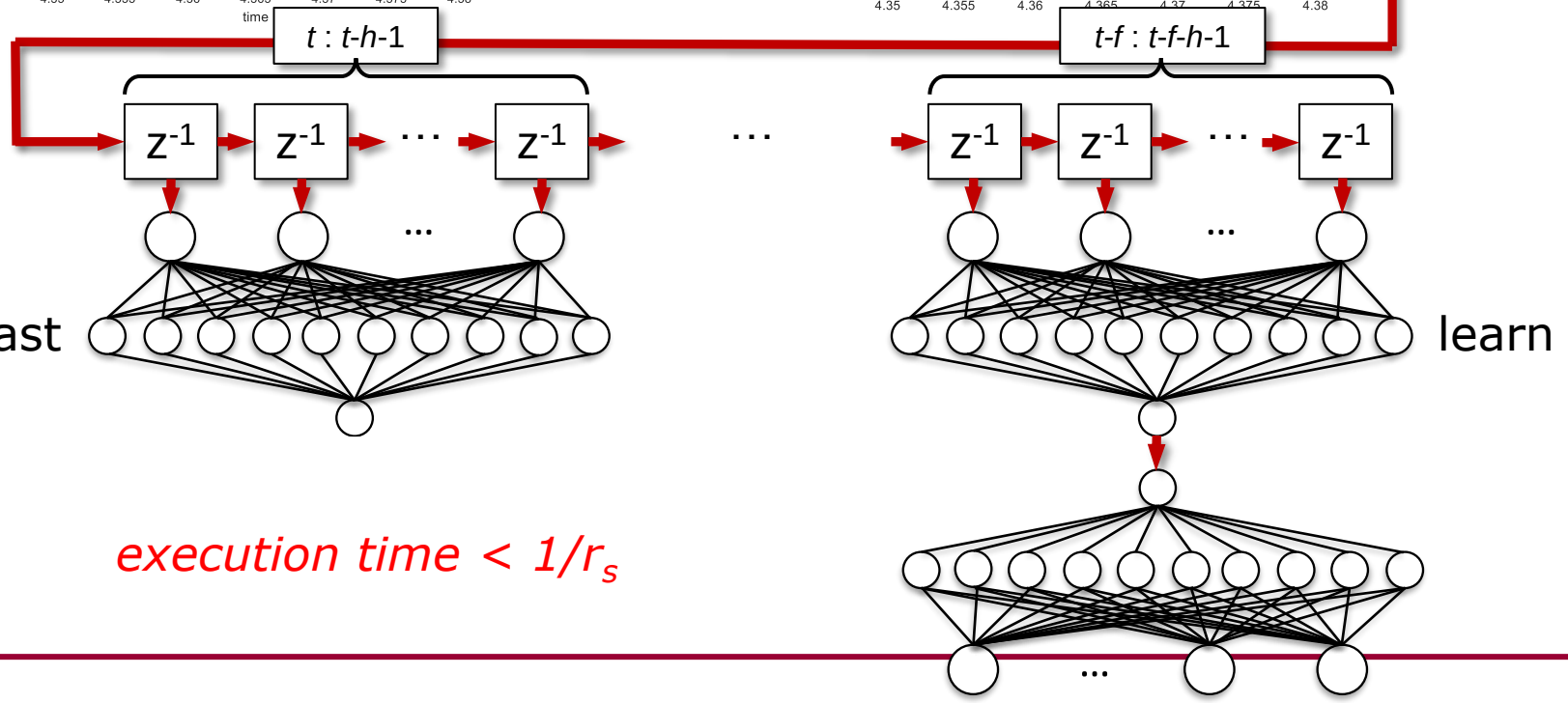


Subsample & Quantize

rate r to rate r_s

word size n





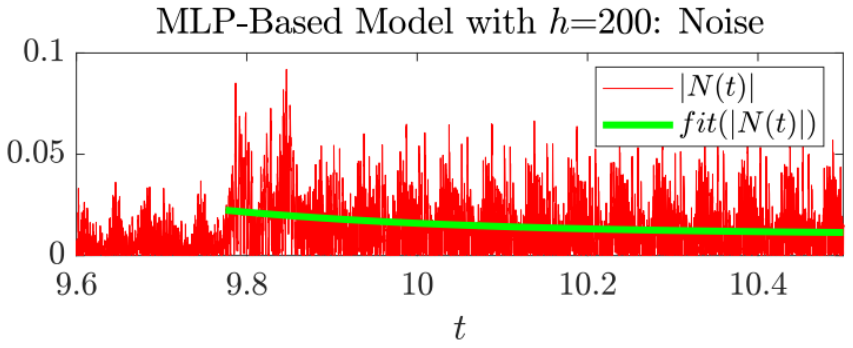
execution time < $1/r_s$

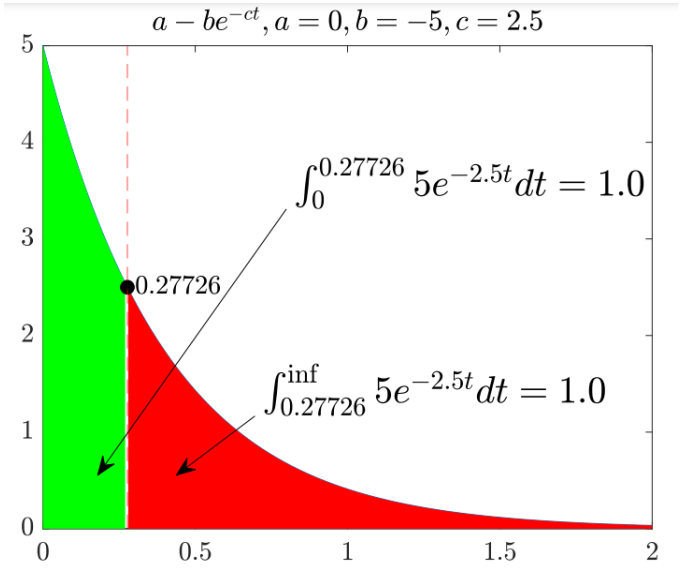
Performance Metrics

- **Forecast accuracy:**
  - $error = output[t] - input[t+f]$
  - $SNR_{db} = \log_{10} \frac{rms(original\ signal)^2}{rms(error)^2} \times 20$

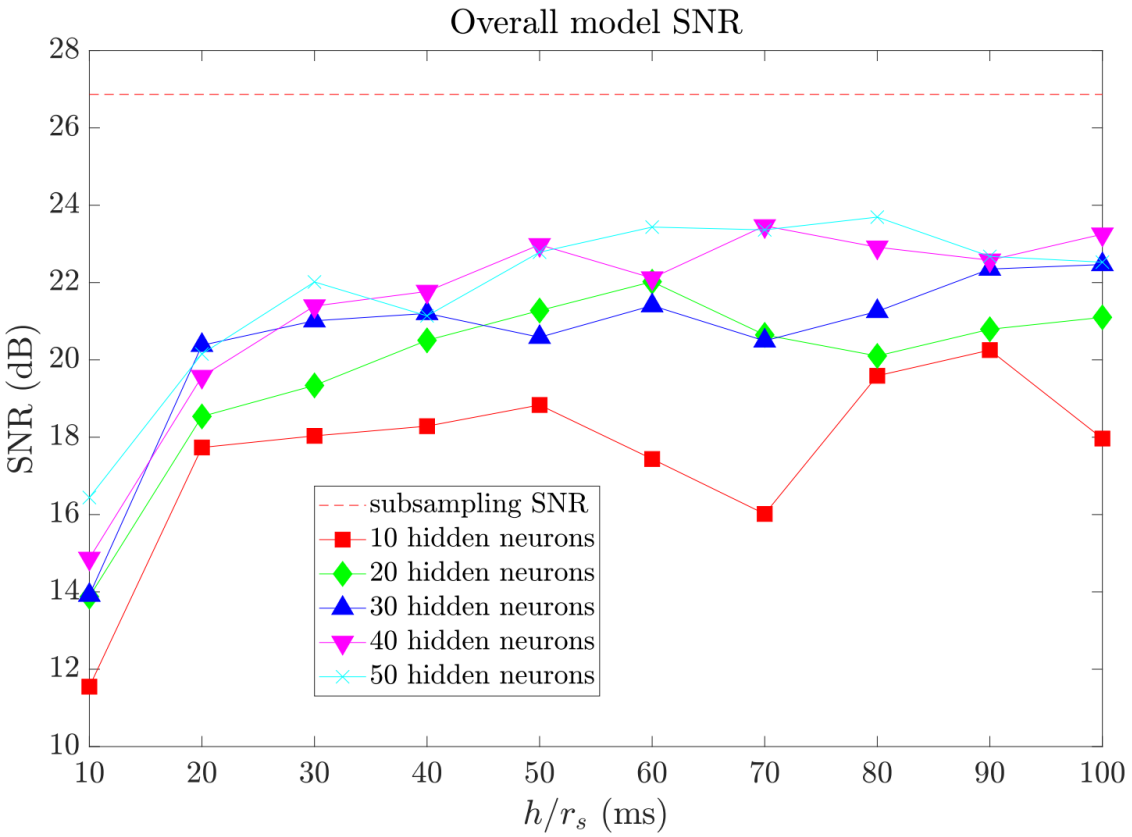
- **Re-training time:**
  1. Fit absolute error to $a - be^{-ct}$
  2. Find "center of gravity" of curve: $\frac{\ln\frac{1}{2}}{c}$

- **Parameters:**
  1. History length (h)
  2. Hidden layer size (s)
  3. Subsample rate (r_s)
  4. Data width (n)

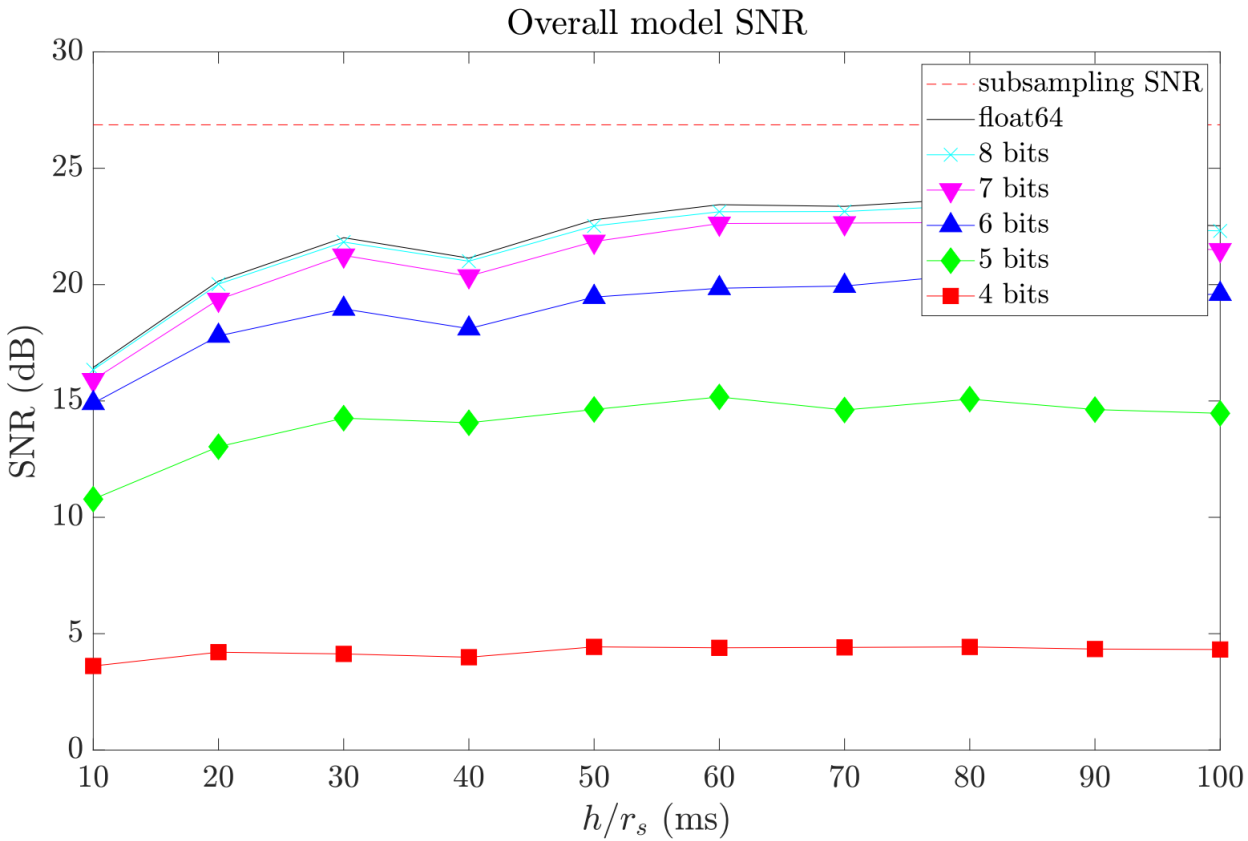




Impact of h, s, r_s, and n on Accuracy

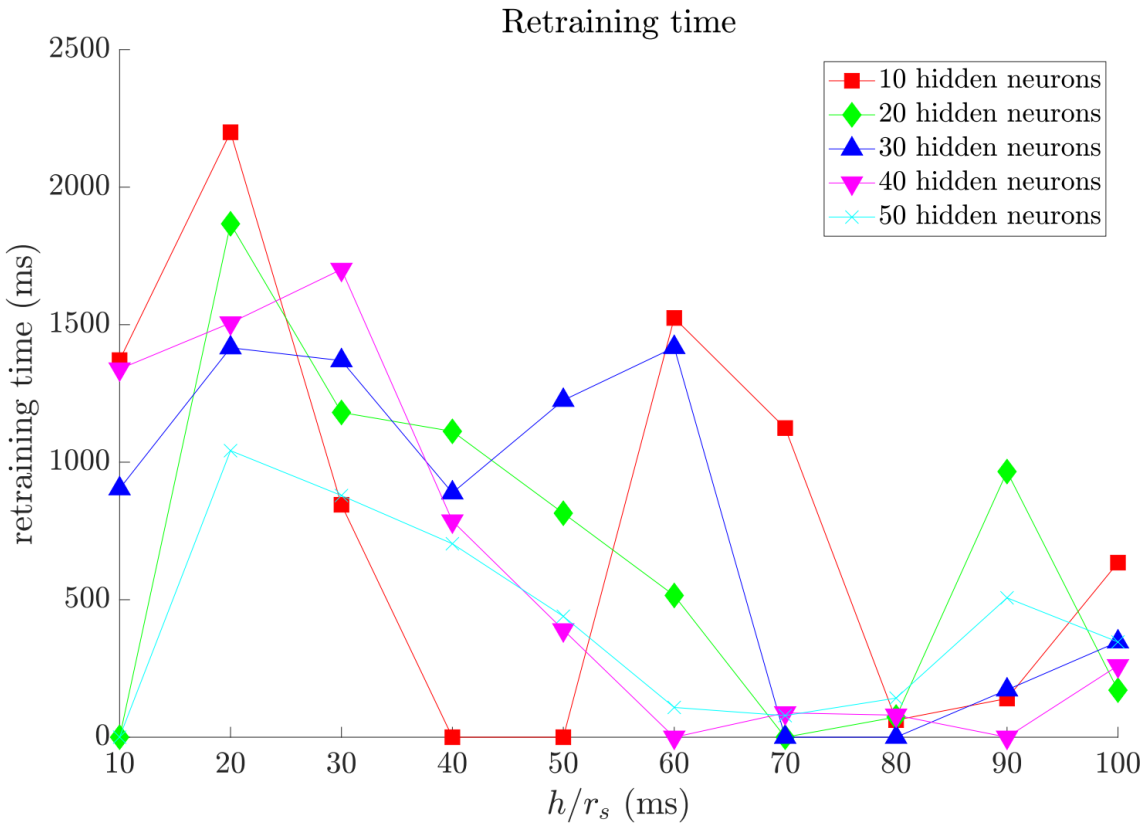


Subsampled @ 2500 Hz

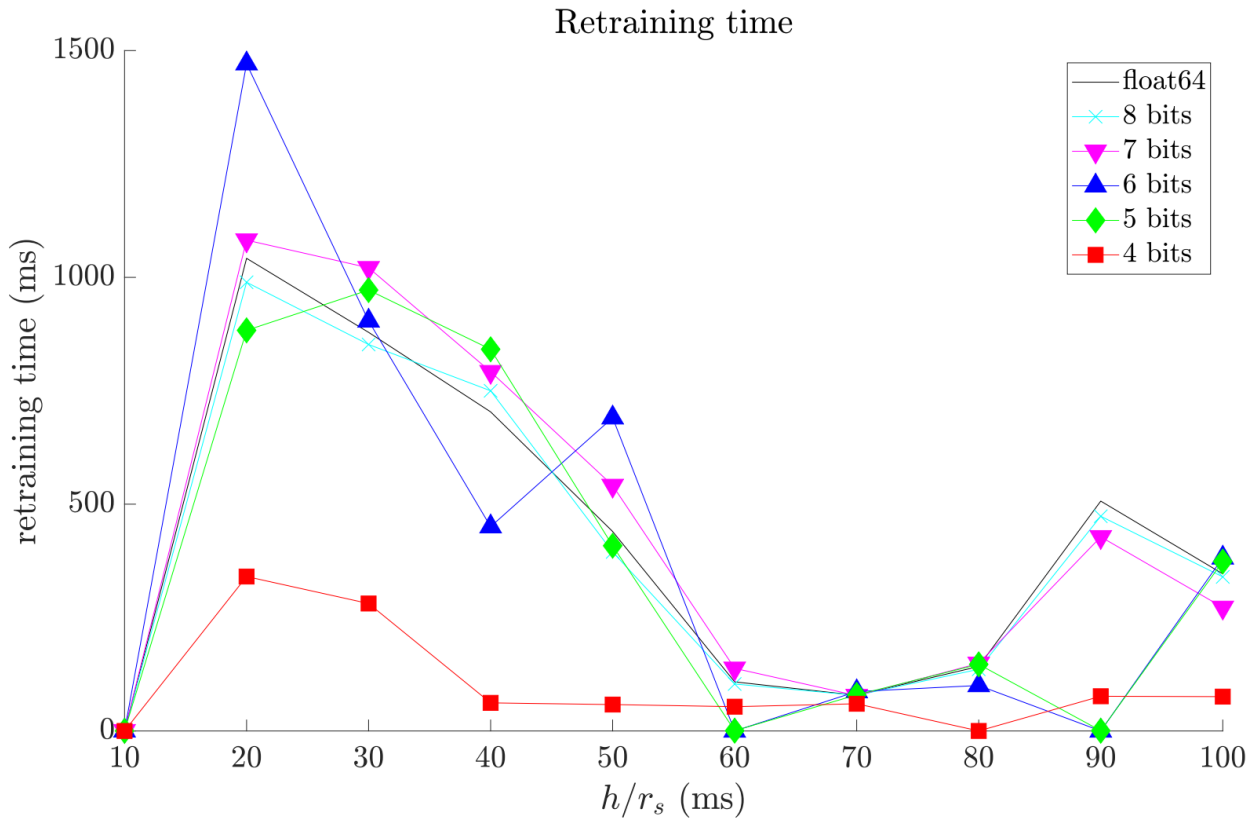


Subsampled @ 2500 Hz
50 hidden neurons

Impact of h, s, r_s, and n on Retraining Time

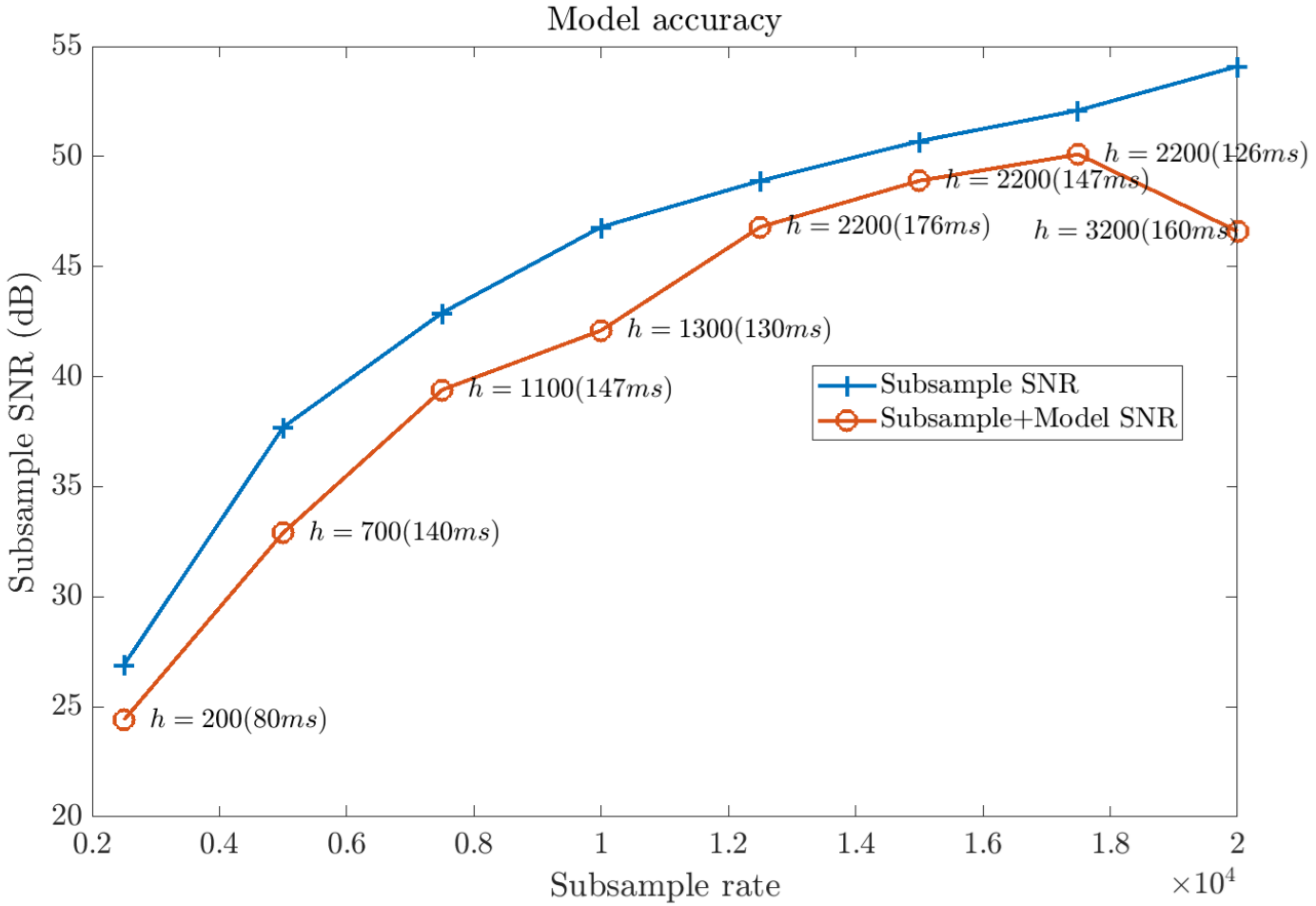


Subsampled @ 2500 Hz



Subsampled @ 2500 Hz
50 hidden neurons

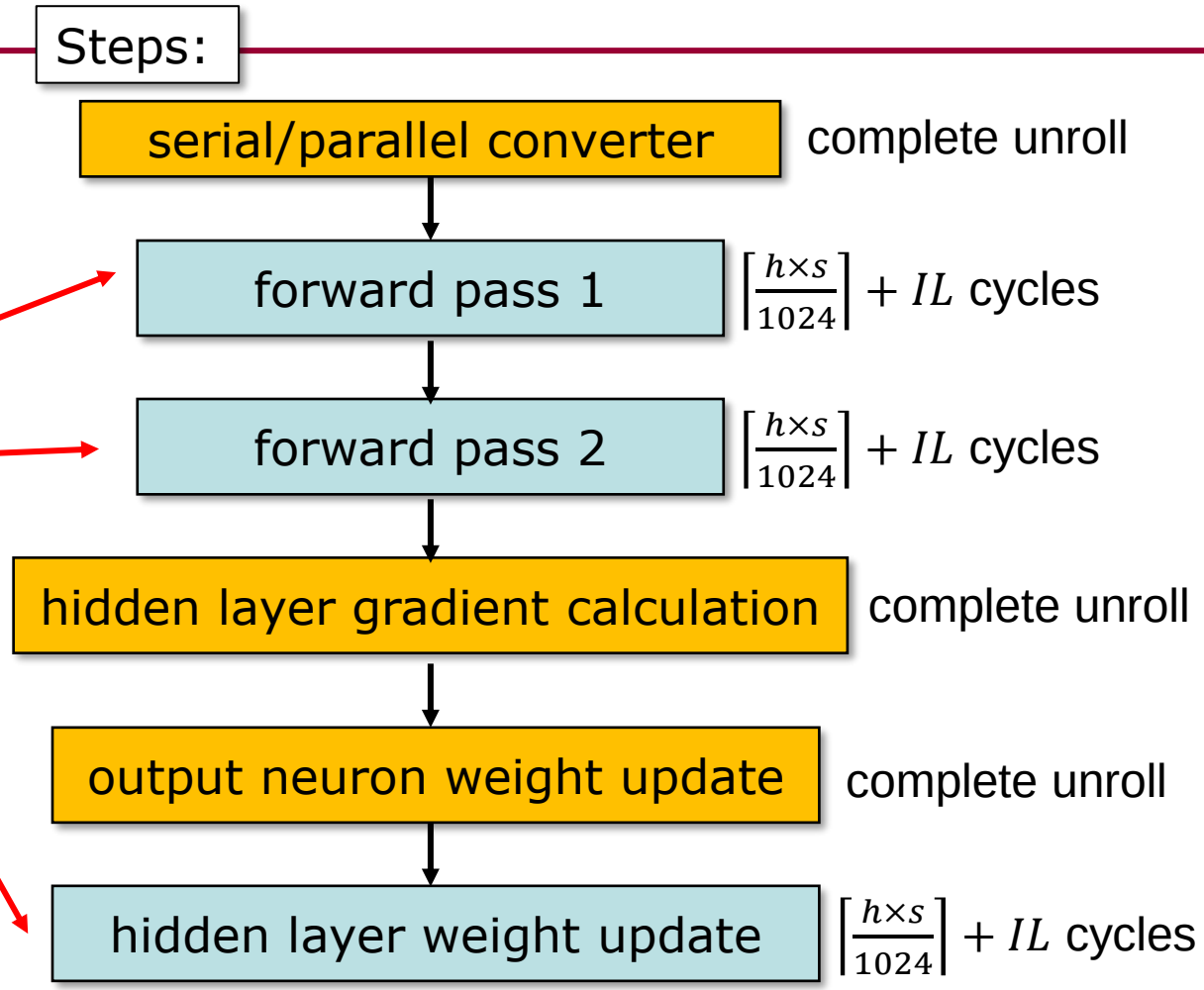
Deployment Results



HLS Architecture

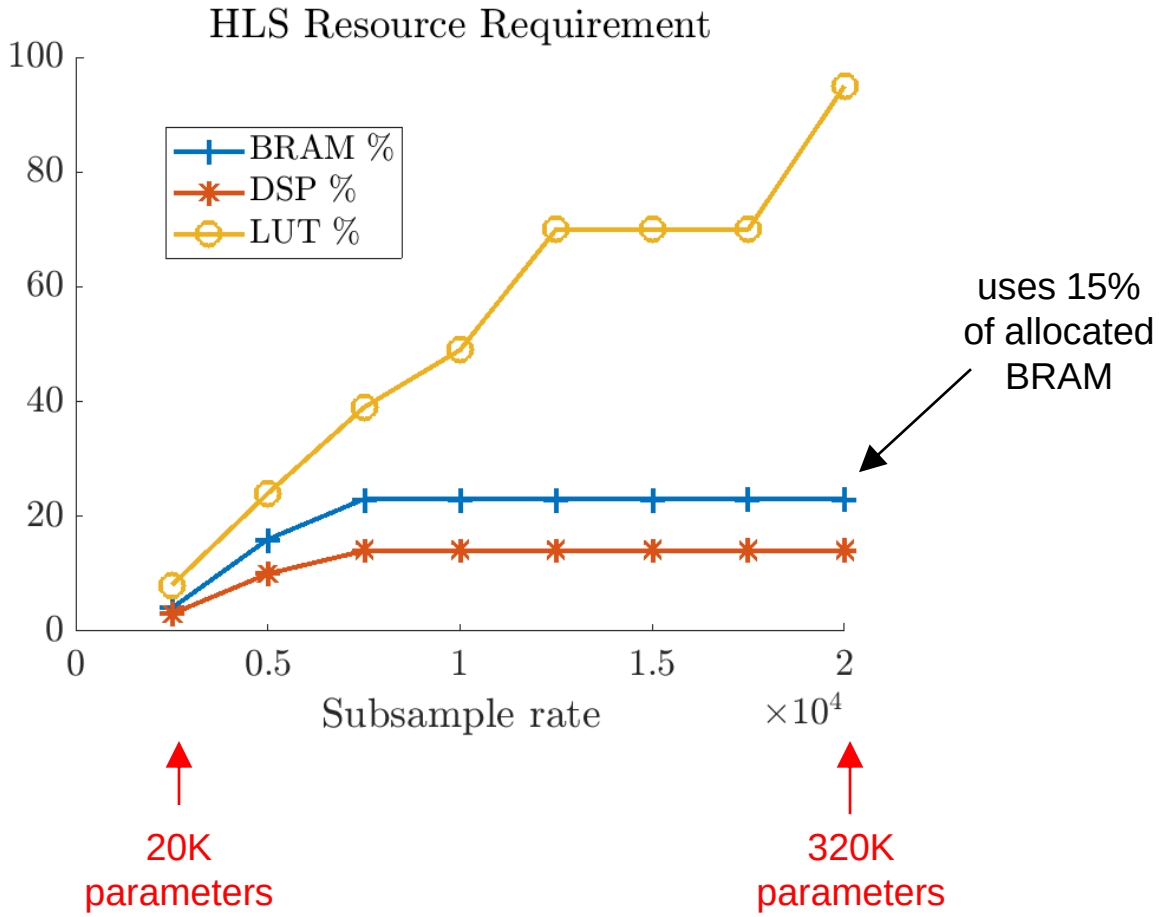
- Vivado HLS 2019.1

- Pipelined loops (hidden layer):
  1. Forward pass 1 (forecast)
  2. Forward pass 2 (loss calculation)
     - Latency = $\left\lceil \frac{h \times s}{1024} \right\rceil + IL$ cycles
       - h = history length
       - s = hidden layer size
     - *IL = ~2*h (for > 400MHz)*
  3. Hidden layer weight update
     - Latency = $\left\lceil \frac{h \times s}{1024} \right\rceil + IL$ cycles
     - IL = ~1 - 7 cycles



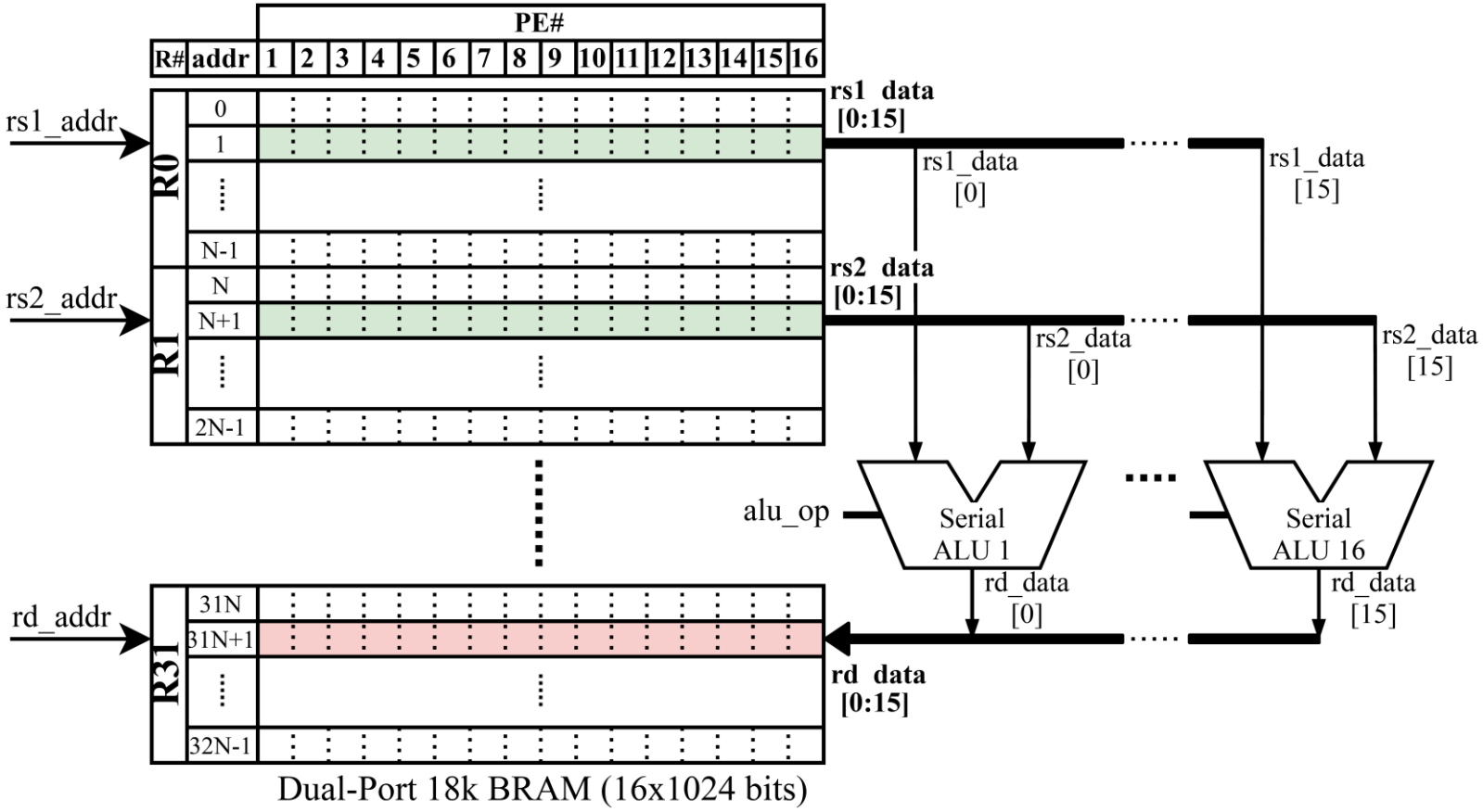
HLS Implementation

- Targeted Virtex Ultrascale+ VU9P
  - 484 MHz
- Has fixed BRAM/DSP usage for 1024-banks
  - 1 MB allocated weight capacity
  - Largest model uses only 15% of allocated RAM
- Current design limited by LUT usage



Array Processor Memory

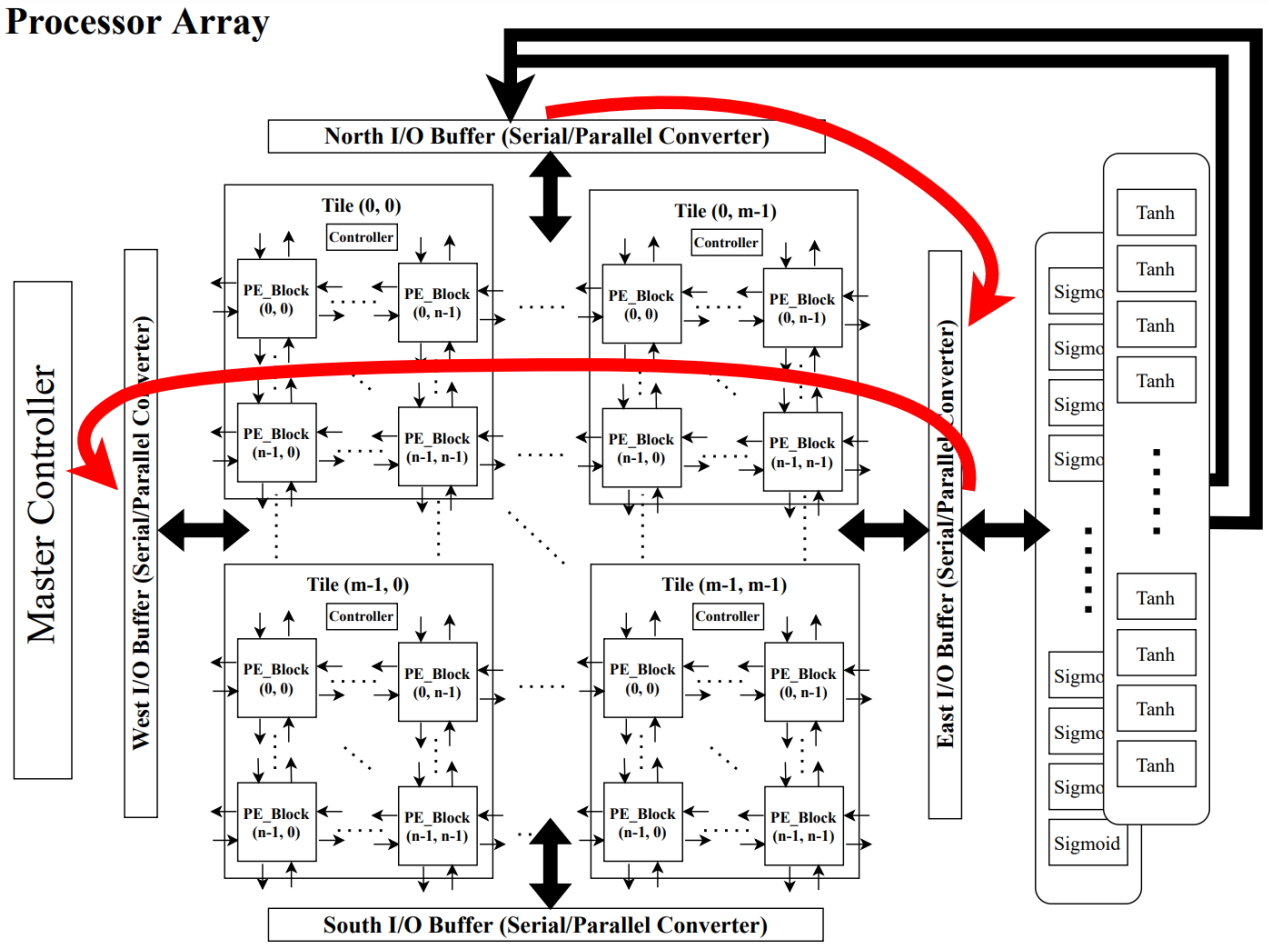
- SPAR-2 processor
  - 2D array of 1-bit PEs
  - Latencies (8-bit):
    - add = 16 cycles
    - mult = 80 cycles
    - n-way reduction = $\lceil \log_2 n \rceil \times 18$ cycles
  - 4x4 block of PEs associated with one BRAM



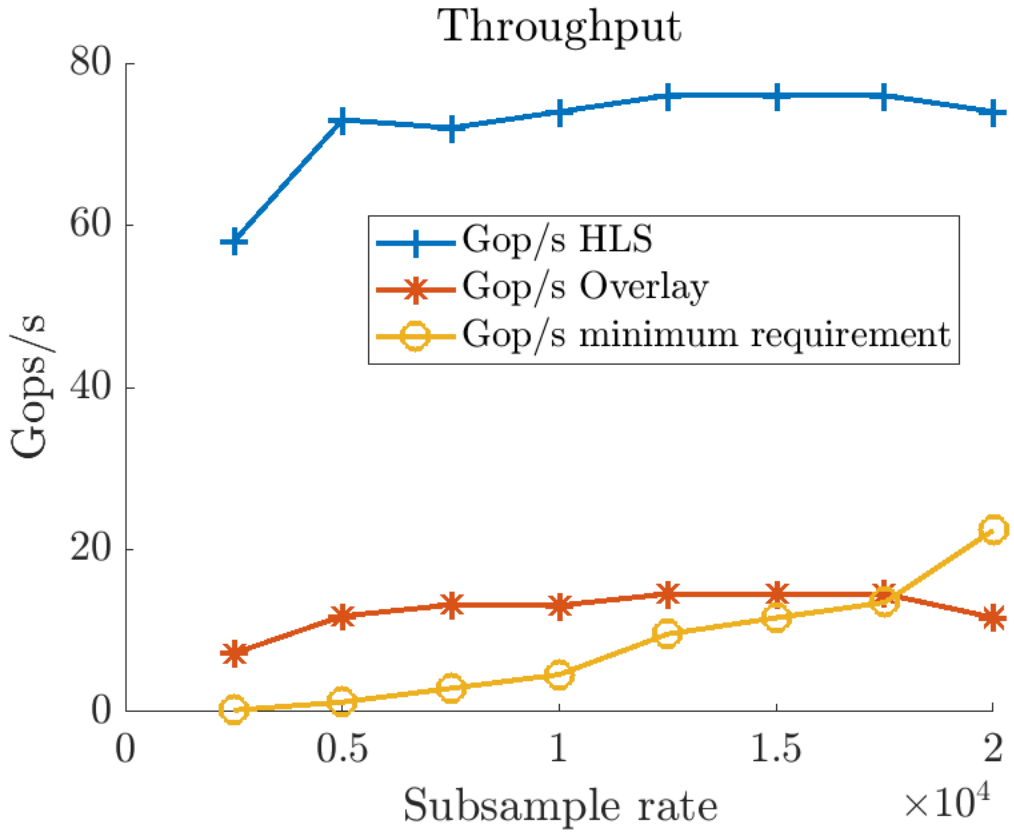
A. Panahi, S. Balsalama, A. T. Ishimwe, J.M. Mbongue, D. Andrews, "A Customizable Domain-Specific Memory-Centric FPGA Overlay for Machine Learning Applications," *FPL 2021.*

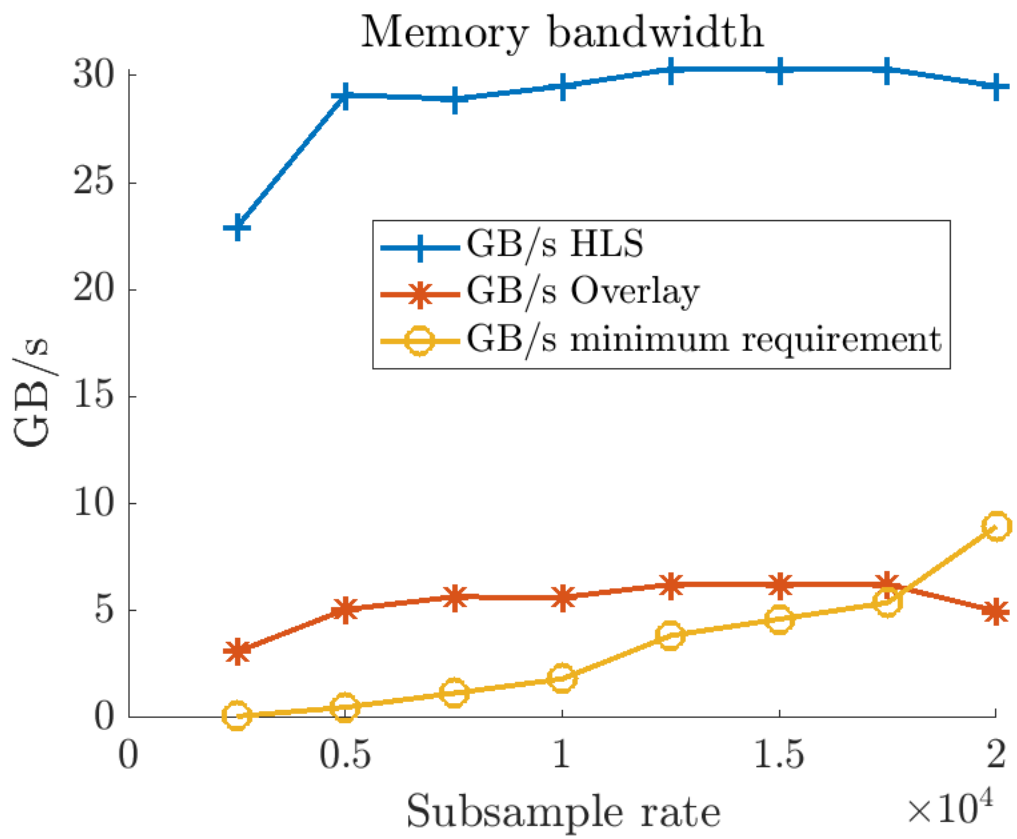
Array Processor Architecture

- Structure:
  - 5x5 blocks/tile
  - 5x5 tiles/grid
  - 10K PEs
- PEs can exchange with neighbors
- 10 MB capacity, best performance when weights < 160 KB
- Custom instructions added for backpropagatation
- vs HLS:
  - Overcomes HLS's 1024-bank limit
  - Limited by multi-cycle adds and lower Fmax of 330 MHz



Performance Results





Conclusions

- Real-time, data driven simultaneous forecasting and learning of time series signals
- Developed two implementations of the system
- Current work:
  1. Dynamically adjust learning rate to improve re-training time
  2. Add support for LSTM forecasters

Thank you!