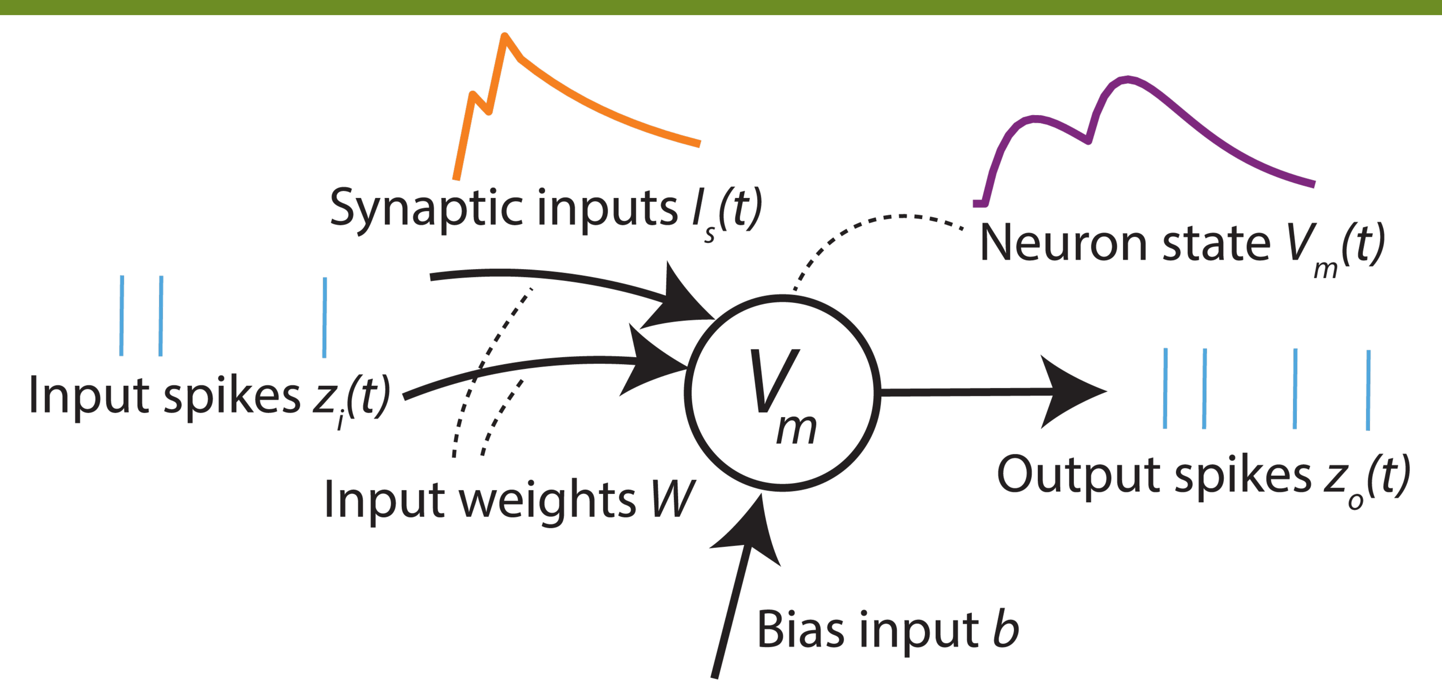
Codefest Presentation

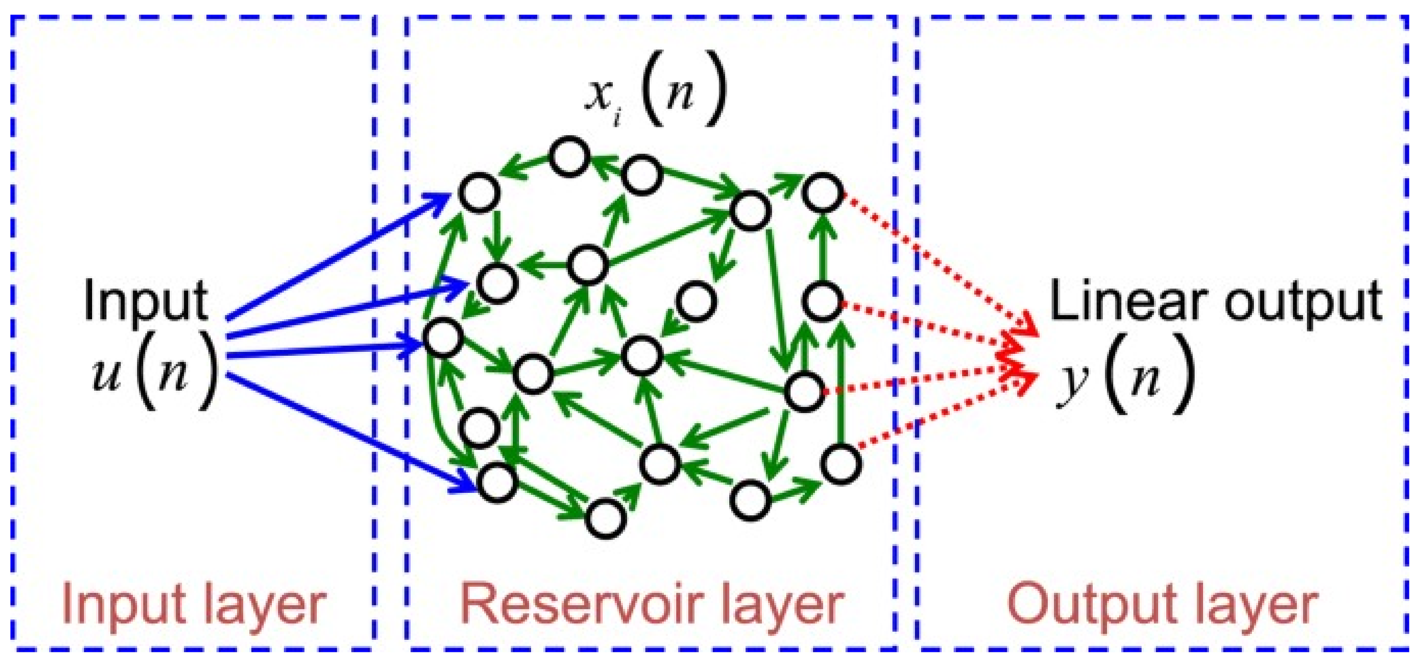
Tucker X Mastin | Department of Electrical and Computer Engineering, Portland State University

What are you trying to do?

I am building an analog model of a Leaky Integrate-and-Fire (LIF) spiking neuron, with the goal of assembling it into a Liquid State Machine (LSM) - a biologically-inspired recurrent neural network that processes temporal input data. In this architecture, the internal dynamics are untrained and only the readout layer is trained, making it an ideal candidate for analog acceleration.

Visualizations





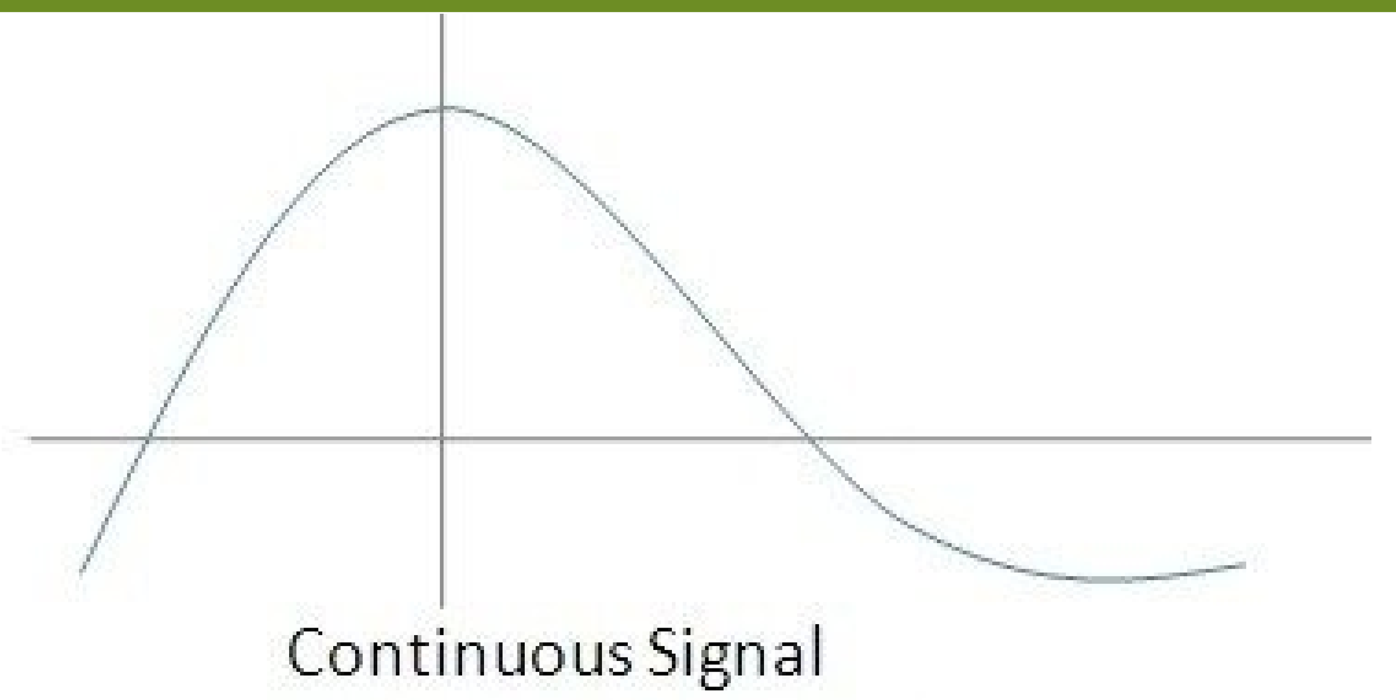
How have others implemented and/or accelerated this algorithm?

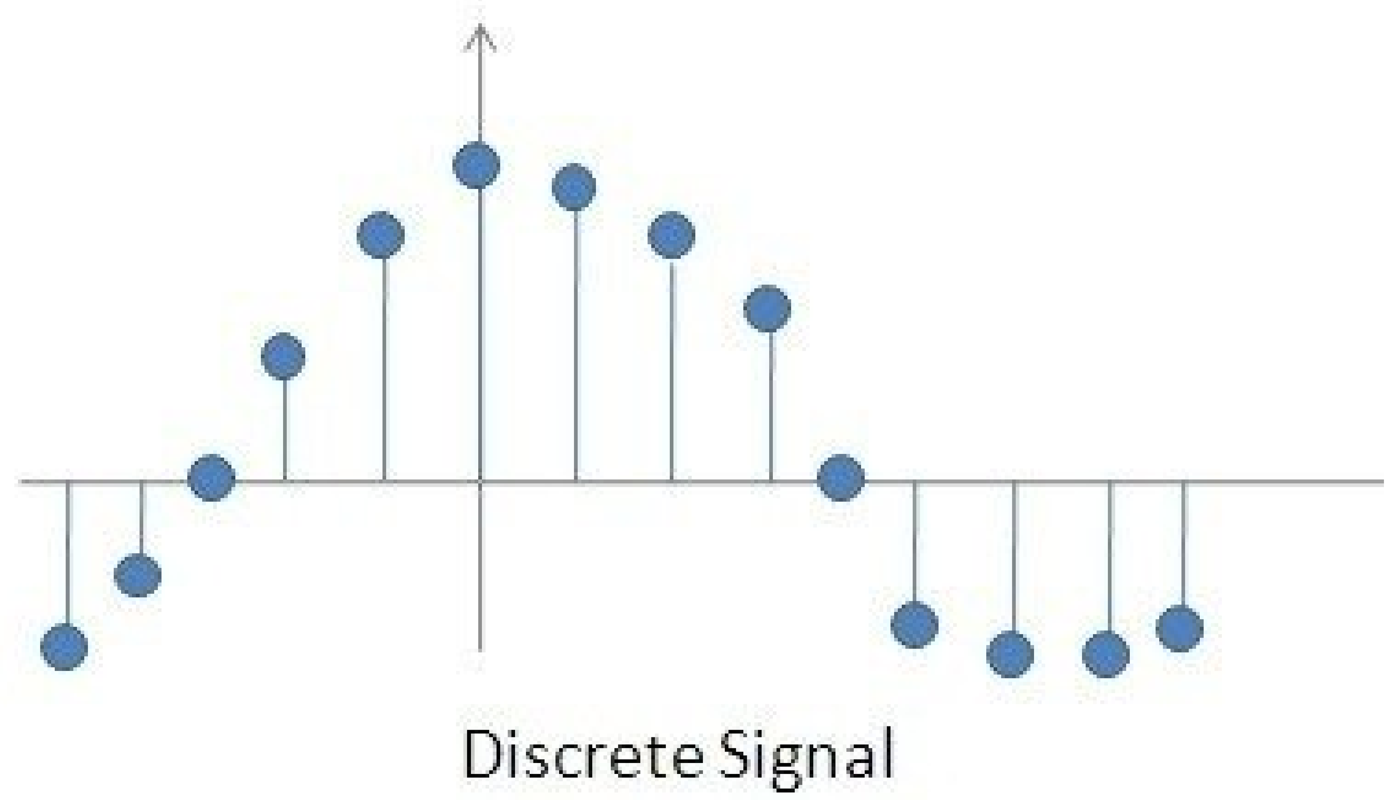
Spiking neural networks and LSMs have been implemented in digital hardware (e.g., FPGAs, GPUs, and neuromorphic platforms such as Intel's Loihi). Most software implementations (e.g., `snntorch`, Brian2, NEST) emulate spiking dynamics using clocked digital logic.

What are you doing differently?

Unlike clocked digital models, this implementation runs in continuous time and may enable a better approximation to biological dynamics. The analog nature also opens the door for ultra-low power neuromorphic inference systems.

Discrete VS Continuous Time





What have you accomplished so far?

- Designed and simulated a working LIF neuron circuit in SPICE.
- Verified leaky integration and reset behavior under constant current input.
- Generated a digital spike output (`spike_out`) when membrane voltage exceeds a threshold.

What will you do next?

- Expand to a multi-neuron reservoir with interconnections.
- Add synaptic connectivity (resistive or capacitive).
- Record spike trains and train a software-based readout layer (e.g., logistic regression) on benchmark time-series tasks.
- Quantify energy per spike and compare with digital baselines.
- Explore scaling and layout feasibility toward an analog ASIC.