Indirect Training of a Spiking Neural Network for Flight Control via Spike-Timing-Dependent Synaptic Plasticity

Greg Foderaro, Craig Henriquez, Silvia Ferrari
Laboratory for Intelligent Systems and Controls (LISC)
Department of Mechanical Engineering and Materials Science
Duke University

CDC 2010
Atlanta, Georgia
12/15/10
Introduction and Motivation

- Develop biologically plausible neurodynamic programming (NDP) algorithms to solve complex control and estimation problems
- Reverse-engineer neuronal cultures *in silico* and train simulated spiking neural networks (SNNs) for adaptive flight control
- Utilize algorithms to determine signal inputs for light-sensitive biological neuronal networks in neuroscience research
Background

- Spiking Neural Networks (SNNs) use the precise timing of discrete spikes to encode information
- Strong evidence of being universal function approximators*
- Potentially capable of handling large range of conditions, failures and damages online
- Gradient-based training algorithms are not applicable to SNNs

Spiking Neural Network Model

- CSIM tool for Matlab environment to construct and simulate networks.
- Uses an adaptive time-step Crank-Nicolson integrator

Hodgkin-Huxley neuron model:

\[
c_m \frac{\nu_m}{\Delta t} = \frac{E_m - V_m}{R_m} - \sum_{c=1}^{N_c} g_c(t)(V_m - E_{rev}^c) + \sum_{s=1}^{N_s} I_s(t) + I_{inject}
\]

- \(V_m\) – membrane potential
- \(C_m\) – membrane capacity
- \(E_m\) – reversal potential of leak currents
- \(R_m\) – membrane resistance
- \(N_c\) – number of channels
- \(G_c(t)\) – current conductance of channel \(c\)
- \(E_{rev}^c\) – reversal potential of channel \(c\)
- \(N_s\) – number of incoming synapses
- \(I_s\) – current supplied by synapse \(s\)
- \(I_{inject}\) – injected current
When $v_m$ exceeds the threshold potential, $V_{\text{thresh}}$, the neuron produces a spike.
Network Training

- Training consists of altering the synaptic efficacies between neurons.
- Greater connection strength corresponds to higher current propagated by a spiking neuron.

Example SNN
As with biological systems, training algorithms must not alter connection strengths manually. Instead we use spike-timing-dependent plasticity (STDP),

\[ f(\hat{t}_{pre}, \hat{t}_{post}) = \text{sgn}(\hat{t}_{post} - \hat{t}_{pre}) \cdot e^{-\frac{|\hat{t}_{post} - \hat{t}_{pre}|}{\tau_d}} \]
To train the network indirectly, feedback is provided to the system in the form of an imitated chemical reward,

\[
r(t) = [b(\hat{y}, y) + r(t - \Delta t)] \cdot e^{-(t - \hat{t})/\tau_c}
\]

The individual synaptic weights are then modified according to the rule,

\[
\Delta w_{ij}(t) = \mu \cdot r(t) \cdot f(\hat{t}_i, \hat{t}_j) \cdot g[w_{ij}(t)]
\]

Where \( f(\hat{t}_i, \hat{t}_j) \) is the STDP term which exhibits similar behavior to that of the implicit STDP mechanism, and

\[
b(\hat{y}, y) = \text{sgn}(y - \hat{y})
\]

\[
g[w_{ij}(t)] = 1 - c_1 \cdot e^{-c_2 |w_{ij}(t)| / w_{\text{max}}}
\]
Controller Architecture

Adaptive flight controller architecture
Spike Train Decoding/Encoding

- Input spike train generated as a Poisson process
- Output decoded using a leaky integrator

\[ P(n, t) = e^{-\lambda t} \frac{\lambda^t}{n!}, \quad n = 0, 1, \ldots \]
Aircraft Controller

- Adaptive controller trained using dynamic aircraft model with poor short-period flying qualities.
- Aircraft restricted to longitudinal motion
- Subjected to initial disturbance, $\alpha = 5^\circ$
- Objective: approximate optimal control law and return to steady level flight
- Minimize settling time and oscillation amplitude

Aircraft model:
\[
\dot{\alpha} = -0.334\alpha + q - 0.027u \\
\dot{q} = -2.52\alpha - 0.387q - 2.6u
\]

$\alpha$ – angle of attack, $q$ – pitch rate, $u$ – control

Optimal control law:
\[
u^* = 2.03\alpha + 1.318q
\]
Results

- After training, the SNN adaptive controller improves the aircraft’s settling time and oscillation amplitude.
- However, it cannot yet closely approximate optimal control law.

![Aircraft dynamics with trained and untrained SNN adaptive controller.](image1)

![Cumulative absolute error relative to optimal control law over each training run.](image2)
Results

Stability Analysis

- Poincaré map

\[ P : x \rightarrow [\alpha, q] \]

- System is seen to orbit an attractor from several initial conditions

- Does not exhibit highly chaotic or unstable behavior

Poincaré map of aircraft dynamics modified by SNN controller. The plot shows four separate runs with initial conditions, \( x = [5,0] \) and \([-5,0] \)
Summary and Conclusions

- Biologically plausible reward-modulated Hebbian learning algorithm for training SNNs for adaptive flight control
- Performance improvement through indirect training of SNN
- Can be applied to neuroscience research using light-sensitive biological neuronal networks

Future work
- Improve spike train encoding and decoding
- Emulate SNNs on memristor chips
Acknowledgments

LISC Members:
Dr. Chenghui Cai
Dr. Kelli Baumgartner
Dr. Guoxian Zhang
Brian Bernard
Greyson Daugherty
Wenjie Lu
Ashleigh Swingler
Keith Rudd
Vikram Raju

Collaborators:
Prof. Antonius van Dongen,
Duke University/National University of Singapore

Sponsors:
This research is funded by the NSF grant ECS #0925407
Questions?