Spiking Neural Network (SNN) Control of a Flapping Insect-scale Robot

Taylor S. Clawson
Silvia Ferrari
Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY

Sawyer B. Fuller
Mechanical Engineering, University of Washington, Seattle, WA

Robert J. Wood
Engineering and Applied Sciences + Wyss Institute, Harvard University, Cambridge, MA

Conference for Decision and Control
Las Vegas, NV
December 13, 2016
Introduction
Overview

• Autonomous systems should be able to adapt to
  – Disturbances
  – Unmodeled dynamics
  – Environmental uncertainty
Robot Miniaturization

451 mm
Miniaturization Advantages

- Physical robustness
- Agility
- Access to small spaces
- Discrete
RoboBee
Existing RoboBee Controllers

• **PID**

• **LQR (Planar Only)**
Existing Controller Limitations

- Require manual tuning
  - Manufacturing uncertainty causes wing torque variations
- Limited maneuvering capability
  - Hovering
  - Lateral maneuvering
SNN Control Structure
Planar RoboBee Dynamics

Linear Momentum Balance

\[ m\ddot{\mathbf{v}} + \omega \times m\mathbf{v} = \mathbf{f}_d + mg + \mathbf{f}_L \]

Angular Momentum Balance

\[ I_y \ddot{\omega} + \omega \times I_y \omega = \tau_c + (\mathbf{r}_w / G \times \mathbf{f}_d) \]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_y )</td>
<td>Inertia</td>
</tr>
<tr>
<td>( m )</td>
<td>Mass</td>
</tr>
<tr>
<td>( \omega )</td>
<td>Rotation rate</td>
</tr>
<tr>
<td>( \mathbf{v} )</td>
<td>Velocity</td>
</tr>
<tr>
<td>( \mathbf{f}_d )</td>
<td>Drag force</td>
</tr>
<tr>
<td>( \mathbf{g} )</td>
<td>Gravity</td>
</tr>
<tr>
<td>( \mathbf{f}_L )</td>
<td>Lift force</td>
</tr>
<tr>
<td>( \tau_c )</td>
<td>Control torque</td>
</tr>
</tbody>
</table>
Maneuvers

Spiking Neural Network (SNN) Controller
Overview
Neuron Model

- Leaky Integrate and Fire

\[ \frac{dv}{dt} = \frac{RI - v}{\tau} + \xi \]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v )</td>
<td>Membrane potential</td>
</tr>
<tr>
<td>( I )</td>
<td>Input current</td>
</tr>
<tr>
<td>( R )</td>
<td>Resistance</td>
</tr>
<tr>
<td>( \tau )</td>
<td>Time constant</td>
</tr>
<tr>
<td>( \xi )</td>
<td>Noise</td>
</tr>
</tbody>
</table>

![Graph showing the dynamics of the Leaky Integrate and Fire model with a blue line indicating the membrane potential and a dotted line indicating the threshold. The x-axis represents time in milliseconds, ranging from 0 to 100, and the y-axis represents potential in millivolts, ranging from 0 to 1. The graph illustrates periodic depolarizations and hyperpolarizations with a stochastic component represented by the noise term.](image)
Training Algorithm

- Network weights adjusted by $\Delta w_{ij}$ so that SNN output $\hat{y}$ approaches reference control $y$

$$
\Delta w_{ij}(t) = \mu \cdot r(t)
$$

$$
r(t) = [\text{sgn}(y - \hat{y}) + r(t - \Delta t)] \cdot e^{(\hat{t}_i - t)/\tau}
$$

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu$</td>
<td>Learning Rate</td>
</tr>
<tr>
<td>$\hat{y}$</td>
<td>Actual Output</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>Time Step</td>
</tr>
<tr>
<td>$y$</td>
<td>Target Output</td>
</tr>
<tr>
<td>$t$</td>
<td>Current Time</td>
</tr>
<tr>
<td>$\hat{t}_i$</td>
<td>Spike Time</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Time Constant</td>
</tr>
<tr>
<td>$\Delta w_{ij}$</td>
<td>Weight Change</td>
</tr>
</tbody>
</table>
Spike Train Decoding

- SNN output is a series of discrete events
- Convert into continuous control signal
Spike Train Decoding

\[ \hat{y}(t) = \alpha \sum_{t^f \in S_i(T)} e^{\beta(t^f - t)} - \gamma, \quad \forall t^f < t \]

- \( \hat{y} \): Decoded output
- \( \alpha \): Scaling factor
- \( \beta \): Decay factor
- \( \gamma \): Offset factor
- \( t^f \): Spike time

- Convolve spike train with an exponential window
Simulation Results
Trajectory Comparison

- LQR-Controlled Bee
- SNN-Controlled Bee

0.3x
Error

Comparing the lateral control signals

Lateral control signal error
Conclusions and Future Work

• SNN effectively matched 2D LQR Controller

• Implement on 3D model

• Use implicit model follow beyond classical controller
Spiking Neural Network (SNN) Control of a Flapping Insect-scale Robot

Taylor S. Clawson
Silvia Ferrari
Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY

Sawyer B. Fuller
Mechanical Engineering, University of Washington, Seattle, WA

Robert J. Wood
Engineering and Applied Sciences + Wyss Institute, Harvard University, Cambridge, MA

Further questions: Taylor Clawson
tsc83@cornell.edu