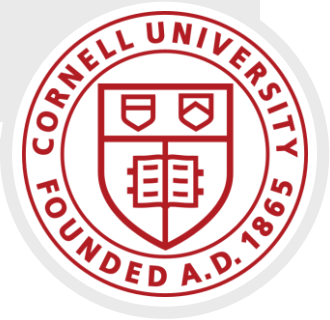
LISC

LABORATORY FOR INTELLIGENT SYSTEMS AND CONTROLS

Neuromorphic Sensing and Control of Autonomous Micro-Aerial Vehicles



Commercialization Fellowship Technical Presentation

Taylor Clawson
Ithaca, NY
June 12, 2018

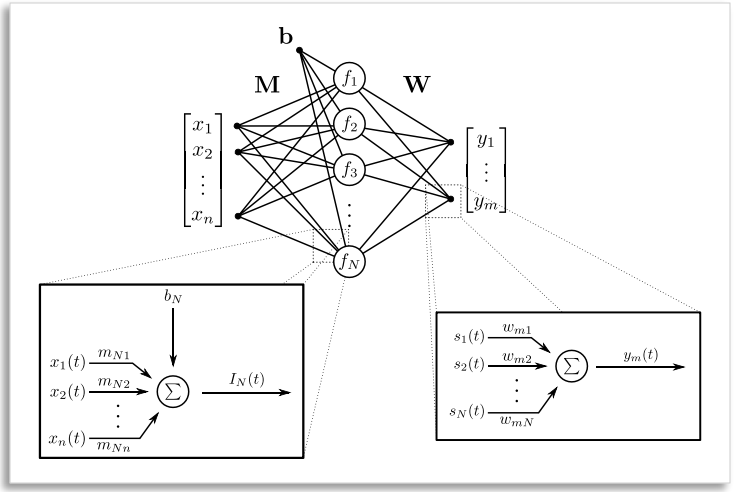
About Me: Taylor Clawson

- 3rd year PhD student in Mechanical and Aerospace Engineering
  - Advisor: Silvia Ferrari
- Past Experience
  - Mechanical Engineering B.S., Utah State University, 2013
  - Lead Developer of custom reliability analysis software at Northrop Grumman
- Graduate Research
  - Dynamics and Controls emphasizing neuromorphic sensing and control
  - Flight modeling for insect-scale robots
  - Neuromorphic autonomy for micro aerial vehicles
  - Autonomous target tracking and obstacle avoidance

Neuromorphic Sensing and Control

The Innovation

- Neuromorphic sensing and control algorithms for intelligent, energy-efficient autonomy



1

Spiking neural networks (SNNs) can learn online to improve performance or adapt to new conditions

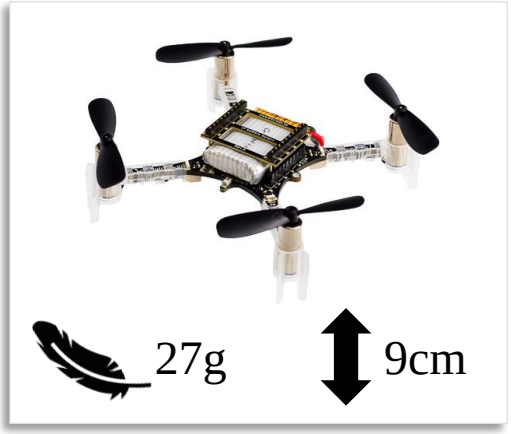
inivation (https://inivation.com/)

2

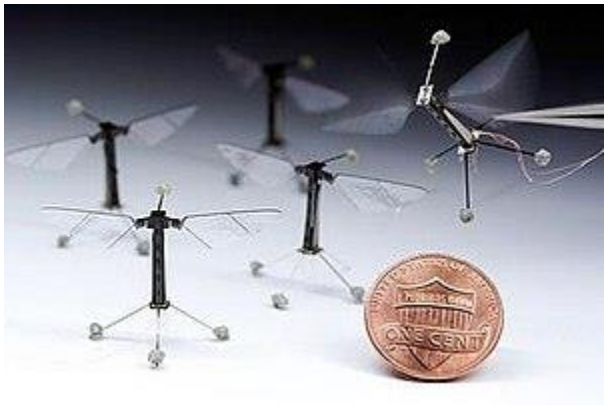
Neuromorphic cameras have 1μs temporal resolution and require at most a few milliwatts of power

Motivation: Small-Scale Autonomous Flight

- Safer: weigh less than 1 pound and thus are safe to operate near humans
- Smaller: can access narrow or confined spaces inaccessible to other vehicles
- Autonomous flight expands the capability of a single operator to monitor a larger area
  - More effective search and rescue
  - Agricultural monitoring
  - Public event security



Crazyflie 2.0 (https://www.bitcraze.io/crazyflie-2/)



RoboBee [Ma, 2013]

Neuromorphic Control

Single-layer SNN Controller

- SNN function approximation by connection weights **M**, **W**, and **b**

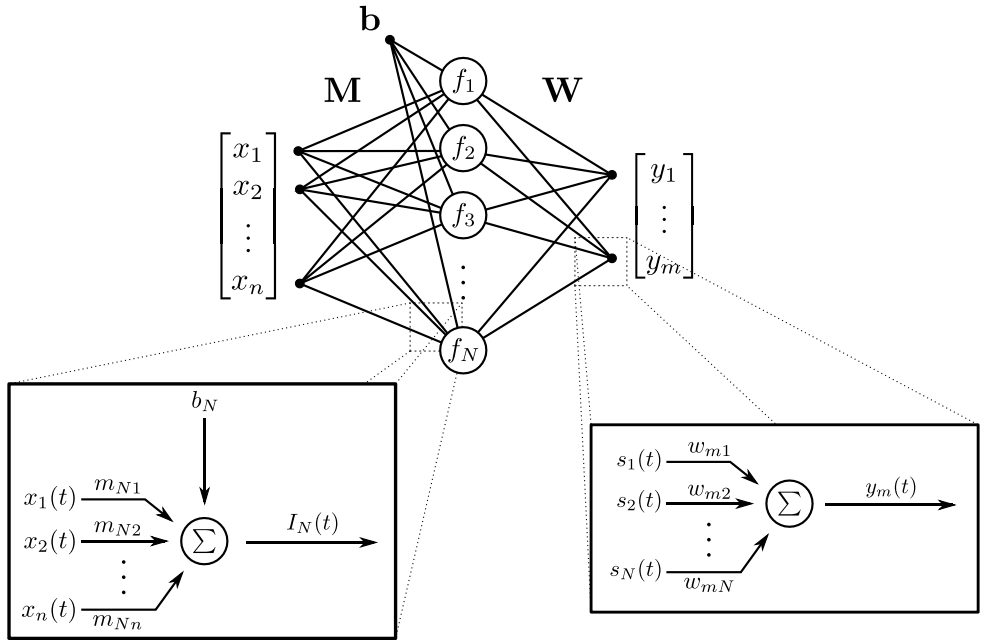
$$\mathbf{y}(t) = \mathbf{W}\mathbf{s}(t) = \mathbf{W}F(\mathbf{M}\mathbf{x}(t) + \mathbf{b})$$

- Output connection weights **W** determined offline by supervised learning

$$\mathbf{W} = \arg\min_{\mathbf{V}} \sum_j \left\| \mathbf{f}(\mathbf{x}_j) - \mathbf{V}F(\mathbf{M}\mathbf{x}_j + \mathbf{b}) \right\|^2$$

- Training data set $\mathcal{D}$ generated by a stabilizing target control law (e.g. optimal linear control)

$$\mathcal{D} = \left\{ \left(\mathbf{x}_j, \mathbf{f}(\mathbf{x}_j) \right) \middle| \; j = 1, \ldots, M \right\}$$



M	Input Connection Weights
W	Output Connection Weights
b	Input bias
$\mathbf{s}(t)$	Post-synaptic current
F	Nonlinear activation function
$\mathbf{f}(\mathbf{x}_j)$	Target control law data
M	Number of training data points

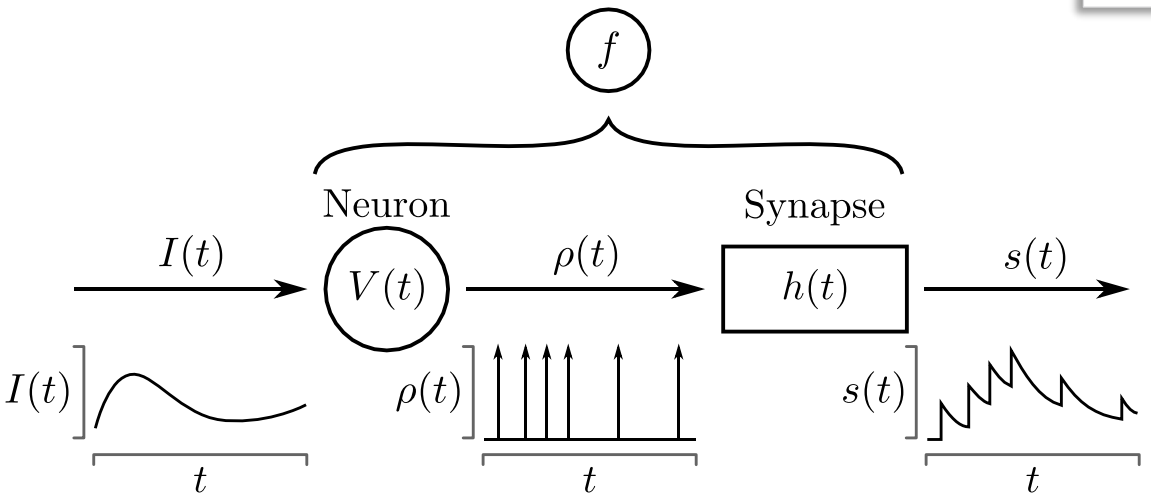
SNN Control Model

- Neurons generate spike trains $\rho(t)$ based on input current $I(t)$
- Synapses filter the spikes and generate post-synaptic current $s(t)$
- Synapses modeled as first-order low-pass filters $h(t)$

$$\rho(t) = \sum_{k=1}^{M} \rho_k(t) = \sum_{k=1}^{M} \delta(t - t_k)$$

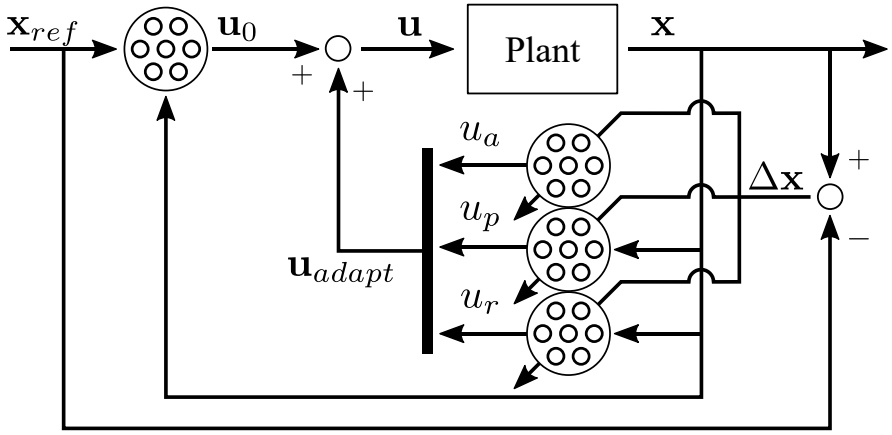
$$s(t) = \int_0^t h(t-\tau)\rho(\tau)d\tau$$

$$h(t) = \frac{1}{\tau_s} e^{-t/\tau_s}$$



δ	Dirac delta
t_k	Time of k^{th} spike
M	Spike count
τ_s	Synaptic time constant

Adaptive SNN Controller (Hovering Only)



$\mathbf{x}_{ref}$	Reference state
$\mathbf{u}_0$	Non-adaptive control input
$\mathbf{u}_{adapt}$	Adaptive control input
$\Delta\mathbf{x}$	State error
u_a	Amplitude input
u_p	Pitch input
u_r	Roll input

- First step towards full flight envelope control
- Control signal $\mathbf{u}(t)$ provided entirely by spiking neural networks

$$\mathbf{u}(t) = \mathbf{u}_0(t) + \mathbf{u}_{adapt}(t)$$

- Non-adaptive term $\mathbf{u}_0(t)$ trained offline by supervised learning to approximate traditional control law
- Adaptive term $\mathbf{u}_{adapt}(t)$ adapts online to minimize output error

[T. S. Clawson, T. C. Stewart, C. Eliasmith, S. Ferrari "An Adaptive Spiking Neural Controller for Flapping Insect-scale Robots," *IEEE Symposium Series on Computational Intelligence (SSCI)*, Honolulu, HI, December 2017]

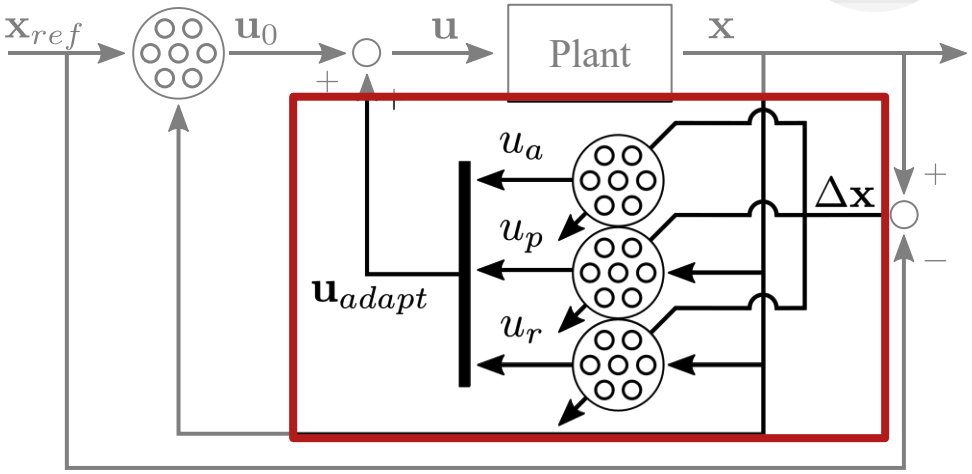
Adaptive SNN Controller (Hovering Only)

- Adaptive term $\mathbf{u}_{adapt}$ comprised of inputs for flapping amplitude u_a, pitch u_p, roll u_r

$$\mathbf{u}_{adapt}(t) = \begin{bmatrix} u_a(t) & u_p(t) & u_r(t) \end{bmatrix}^T$$

- Output weights adapt online to minimize output error

$$E(t) = \mathbf{\Lambda}^T (\Delta x(t) + \alpha \Delta \dot{\mathbf{x}}(t))$$



- Every element of $\mathbf{u}_{adapt}$ computed from a single network of 100 neurons, e.g.

$$u_a = \mathbf{W}_a \mathbf{s}_a(t)$$

- The connection weights are updated online to minimize output error

$$\dot{\mathbf{W}}_a(t) = \gamma \mathbf{s}_a(t) E(t)$$

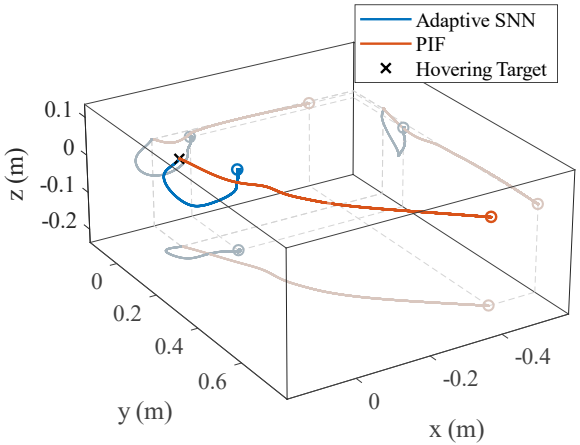
$\mathbf{\Lambda}$	Error weight matrix
Δx	State error
$\mathbf{W}_a$	Output connection weights
$\mathbf{s}_a$	Post-synaptic current
γ	Learning rate

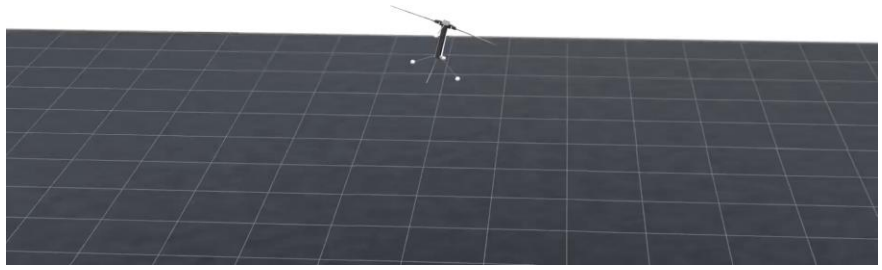
[T. S. Clawson, T. C. Stewart, C. Eliasmith, S. Ferrari "An Adaptive Spiking Neural Controller for Flapping Insect-scale Robots," *IEEE Symposium Series on Computational Intelligence (SSCI)*, Honolulu, HI, December 2017]

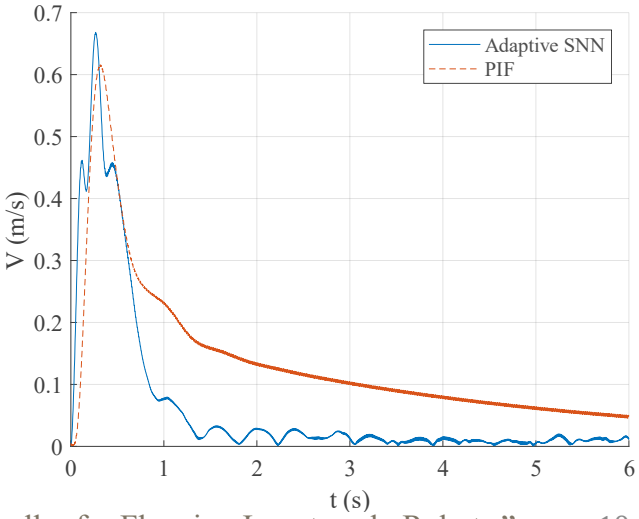
Adaptive SNN Controller (Hovering Only)

Comparison:

- SNN initialized with traditional controller
- SNN controller quickly adapts to wing asymmetries to stabilize hovering flight
- Traditional controller maintains stability, but drifts significantly from the target







[T. S. Clawson, T. C. Stewart, C. Eliasmith, S. Ferrari "An Adaptive Spiking Neural Controller for Flapping Insect-scale Robots," *IEEE Symposium Series on Computational Intelligence (SSCI)*, Honolulu, HI, December 2017]

SNN Controller – Full Flight Envelope

- SNN trained to approximate steady-state gain of gain-scheduled controller
- Gain matrices dependent on scheduling variables **a**

$$\dot{\mathbf{u}}(t) = -\mathbf{K}_1(\mathbf{a})\mathbf{x}(t) - \mathbf{K}_2(\mathbf{a})\mathbf{u}(t) - \mathbf{K}_3(\mathbf{a})\boldsymbol{\xi}(t)$$

- Steady-state gain computed using transfer function and final value theorem

$$\mathbf{G}(s) \triangleq -(s\mathbf{I} + \mathbf{K}_2(\mathbf{a}))^{-1}\mathbf{K}_1(\mathbf{a})$$

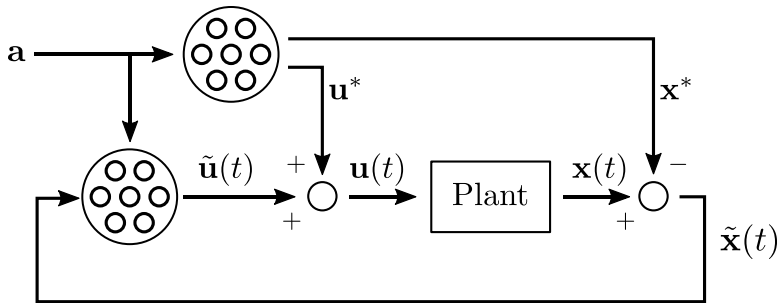
$$\mathbf{G}(0) = -\mathbf{K}(\mathbf{a})_2^{-1}\mathbf{K}_1(\mathbf{a}) \triangleq \mathbf{K}_{ss}(\mathbf{a})$$

- Network output weights computed to approximate steady-state gain matrix $\mathbf{K}_{ss}$

$$\mathbf{W} = \underset{\mathbf{v}}{\operatorname{argmin}} \sum_j \left\| \mathbf{K}_{ss}(\mathbf{a}) - \mathbf{V}F(\mathbf{M}\mathbf{a}_j + \mathbf{b}) \right\|^2$$

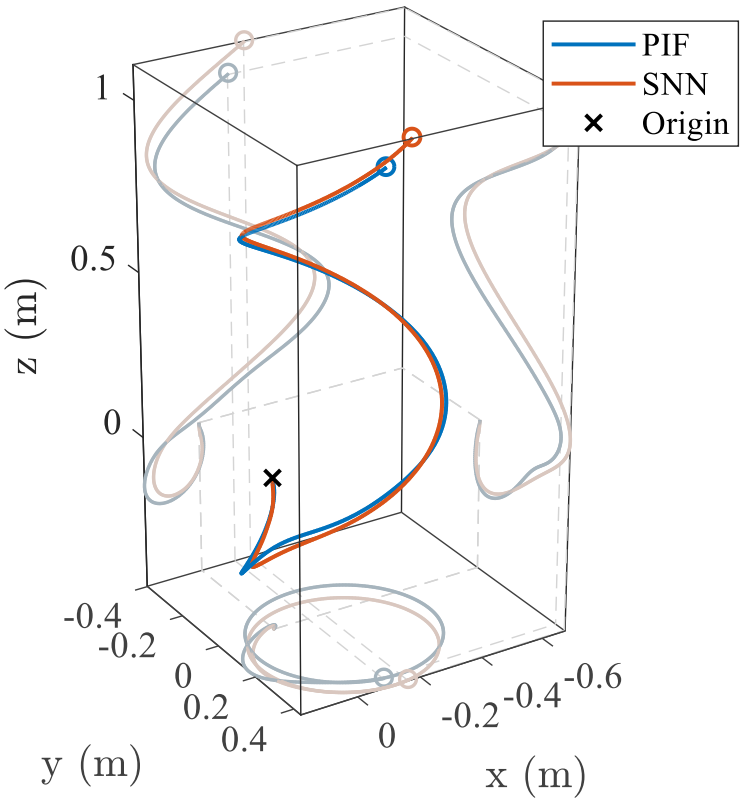
- SNN Control input is a linear transformation of post-synaptic current

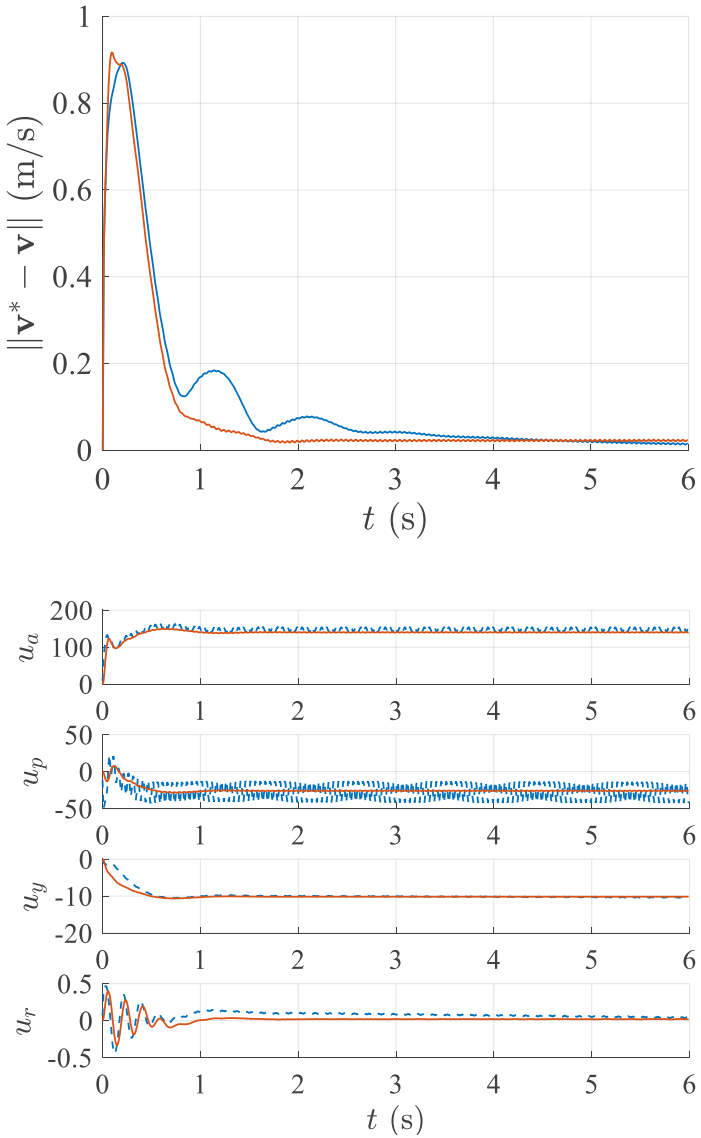
$$\mathbf{u}(t) = \mathbf{W}(\mathbf{a})\mathbf{s}(t)$$



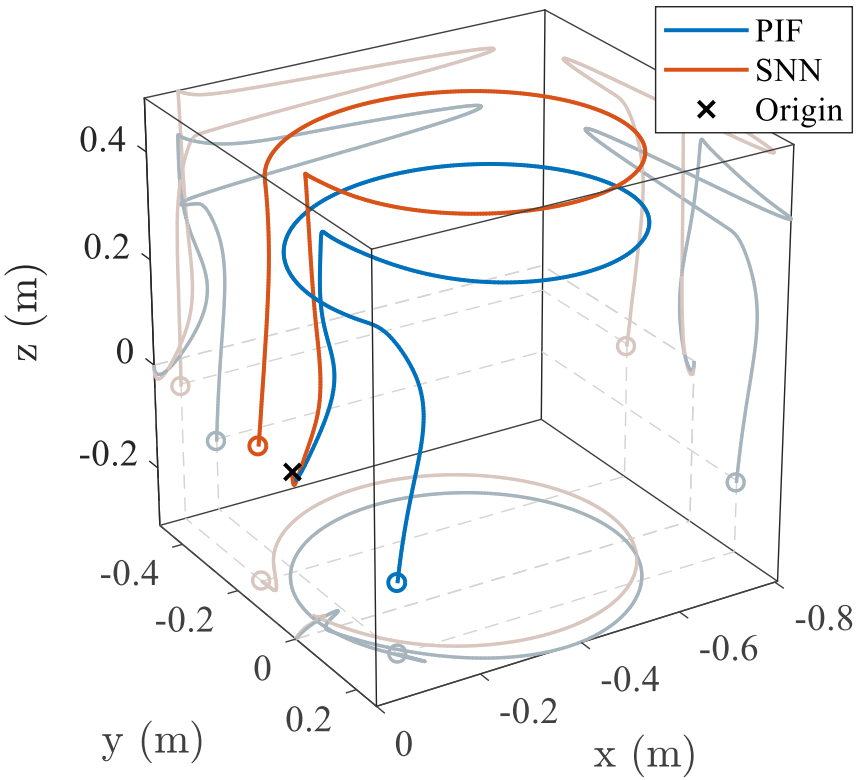
$\mathbf{K}_i$	PIF gain matrices	$\mathbf{x}$	State deviation
$\mathbf{u}$	Control deviation	ξ	Integral of output error
$\mathbf{a}$	Scheduling variables	$\mathbf{G}(s)$	Transfer function
s	Laplace variable	$\mathbf{K}_{ss}$	Steady-state gain matrix
$\mathbf{W}$	Output connection weights	$\mathbf{s}$	Post-synaptic current

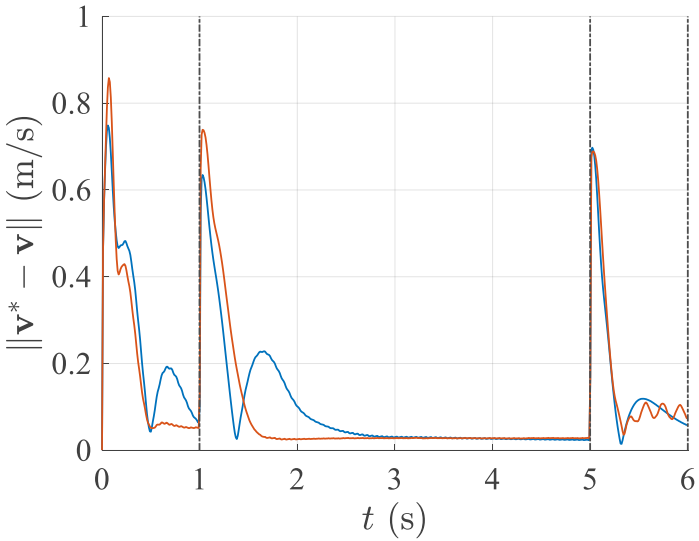
SNN Control – Climbing Turn

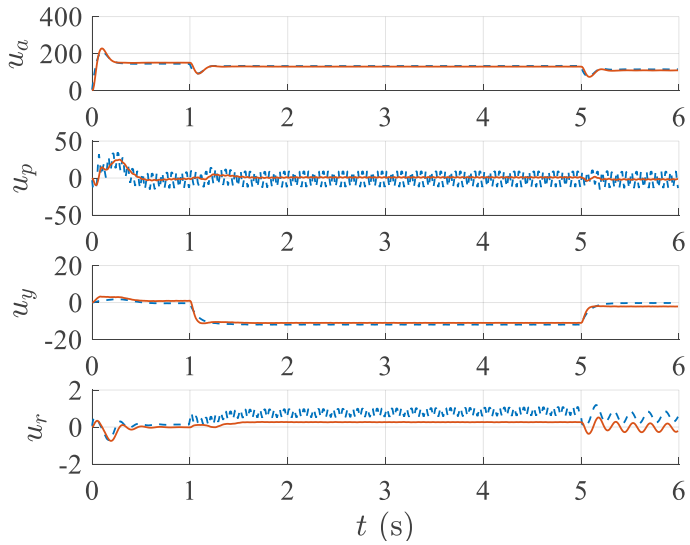




SNN Control – Complete Turn







Neuromorphic Sensing

Exteroceptive Sensing Motivation

- Onboard exteroceptive sensors required for full flight autonomy
- Fast dominant time scales of small-scale flight require high sensing rate and low latency
  - Traditional sensors consume large amounts of power for high sensing rate (e.g. ~100 watts for high speed camera)
  - High data rate requires additional data processing
- Neuromorphic vision sensors have 1μs temporal resolution and require at most a few milliwatts of power [Lichtsteiner, ’08], [Brandli, ’14]

Image Credit: inivation (https://inivation.com)

Background: Neuromorphic Vision

- Neuromorphic cameras generate asynchronous events instead of frames
- An event at (x, y) is generated at time t_i, with polarity

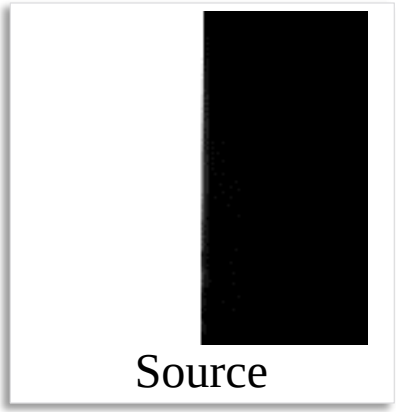
$$p_i = \begin{cases} 1, & \text{if } \ln(I(x,y,t_{i-1})) - \ln(I(x,y,t_i)) = -\theta \\ -1, & \text{if } \ln(I(x,y,t_{i-1})) - \ln(I(x,y,t_i)) = \theta \end{cases}$$

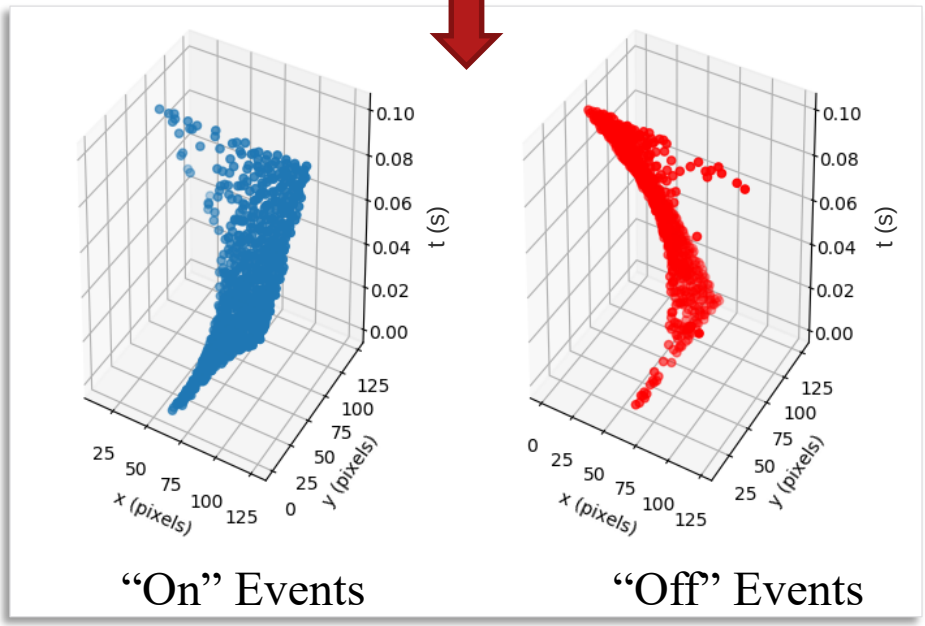
- "On" events when $p_i = 1$
- "Off" events when $p_i = -1$
- The ith event $\mathbf{e}_i$ is described by the tuple $\mathbf{e}_i = (x, y, t, p)_i$

$$x, y \in \mathbb{N}^+ \qquad t \in \mathbb{R}^+ \qquad p \in \{-1, 1\}$$

- The set of all events is

$$\mathcal{E} = \{\mathbf{e}_i \,|\, i = 1, \ldots, N\}$$

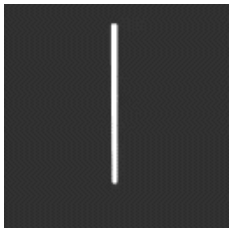




The Optical Flow Problem

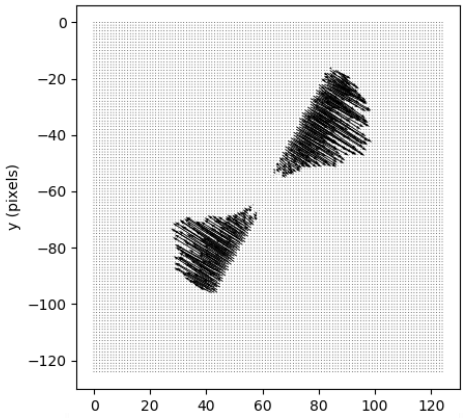
Standard Optical Flow Problem

- Assume: $\frac{dI(x,y,t)}{dt} = 0$



- Determine horizontal and vertical flow (v_x, v_y) from

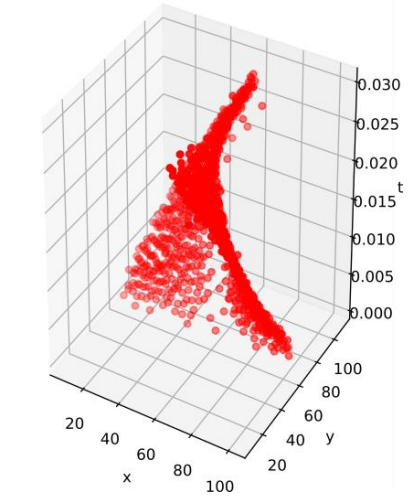
$$\frac{dI(x,y,t)}{dt} = \begin{bmatrix} I_x(x,y,t) & I_y(x,y,t) \end{bmatrix} \begin{bmatrix} v_x(x,y,t) \\ v_y(x,y,t) \end{bmatrix} + I_t(x,y,t) = 0$$



Neuromorphic Optical Flow

- Coordinates of some point $\mathbf{r} = \begin{bmatrix} r_x & r_y \end{bmatrix}^T$ in the image plane determined by optical flow

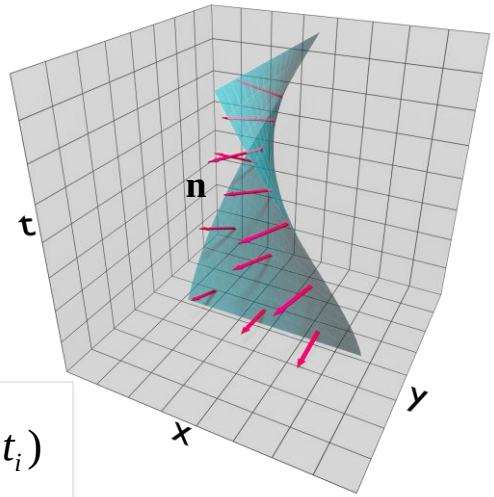
$$\begin{bmatrix} r_x(t_2) - r_x(t_1) \\ r_y(t_2) - r_y(t_1) \end{bmatrix} = \int_{t_1}^{t_2} \mathbf{v}(\tau) d\tau \approx \begin{bmatrix} v_x dt \\ v_y dt \end{bmatrix}, \qquad \mathbf{v}(\tau) = \begin{bmatrix} v_x(\tau) \\ v_y(\tau) \end{bmatrix}$$



- Scattered events are generated by motion of the point
- Determine optical flow by estimating the motion of points in the scene using the scattered events

Neuromorphic Optical Flow

- Existing neuromorphic optical flow methods rely on optimization [Benosman, '14], [Rueckauer, '16]
- Estimate continuous motion from discrete events
- Introduce continuous event rate *f* through convolution of events with continuous kernel *K*



$$f(x,y,t) = K(x,y,t) * E(x,y,t)$$

$$E(x,y,t) = \sum_{i=1}^{N} \delta(x - x_i, y - y_i, t - t_i)$$

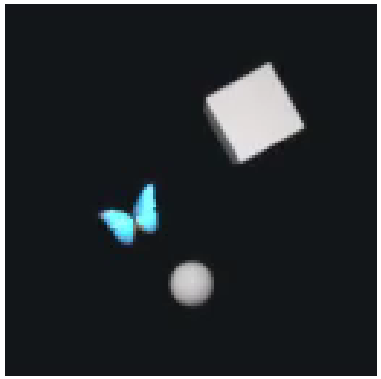
- Assume gradient **n** of event rate is normal to the motion of points in the scene
- Speed of the motion is inversely proportional to magnitude of gradient
- Optical flow is written directly in terms of the event rate gradient

$$\mathbf{n} = \begin{bmatrix} a & b & c \end{bmatrix}^T$$

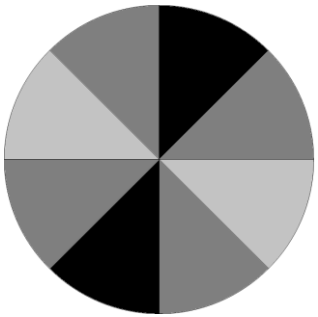
$$\mathbf{w} = \begin{bmatrix} \frac{\partial t}{\partial x} & \frac{\partial t}{\partial y} \end{bmatrix}^T = \begin{bmatrix} -\frac{a}{c} & -\frac{b}{c} \end{bmatrix}^T$$

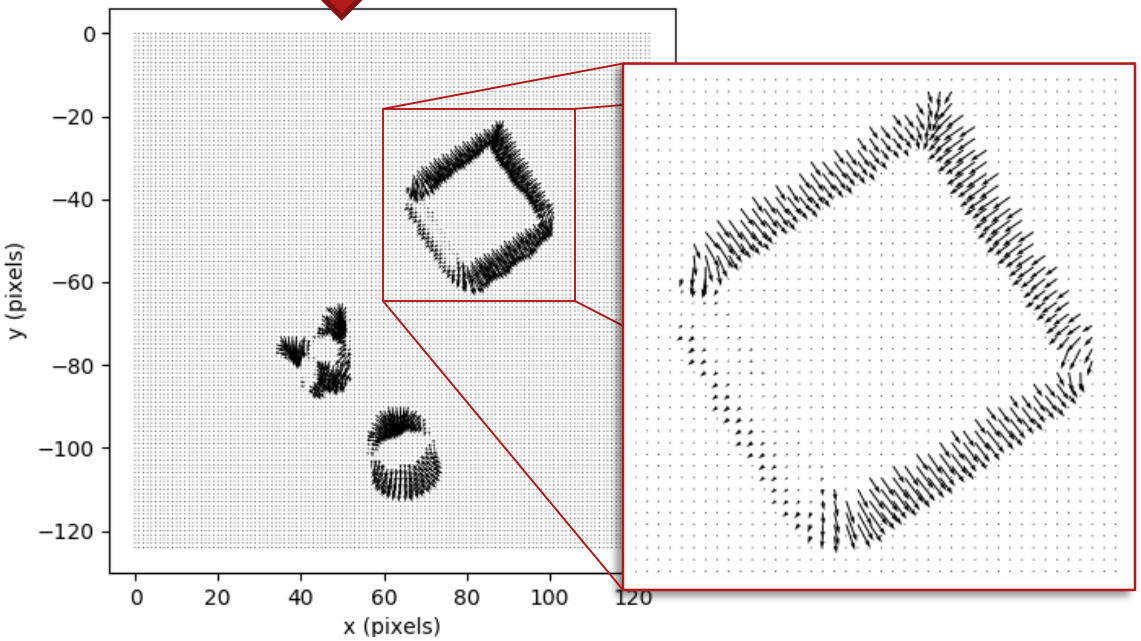
$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \left(\frac{1}{\|\mathbf{w}\|} \right) \frac{\mathbf{w}}{\|\mathbf{w}\|} = -\frac{c}{a^2 + b^2} \begin{bmatrix} a \\ b \end{bmatrix}$$

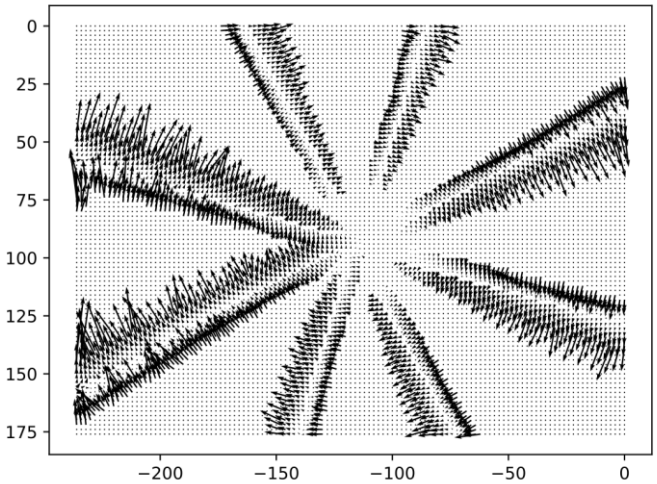
Neuromorphic Optical Flow Results



Mean processing time per event: 0.7 μs







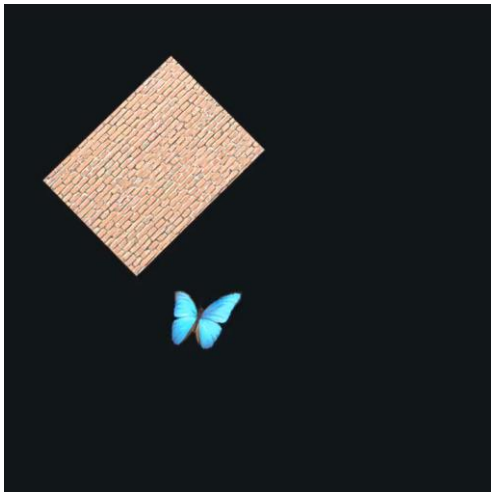
Neuromorphic Motion Detection

Detect motion relative to the environment
using a rotating neuromorphic camera

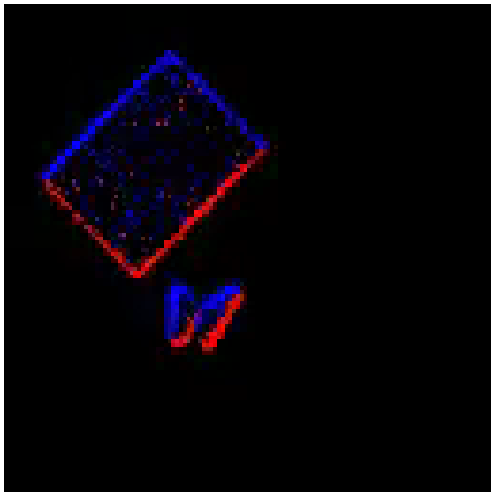
Assumptions

- Known camera motion
- Camera motion dominated by rotation
- Total derivative of pixel intensity is zero

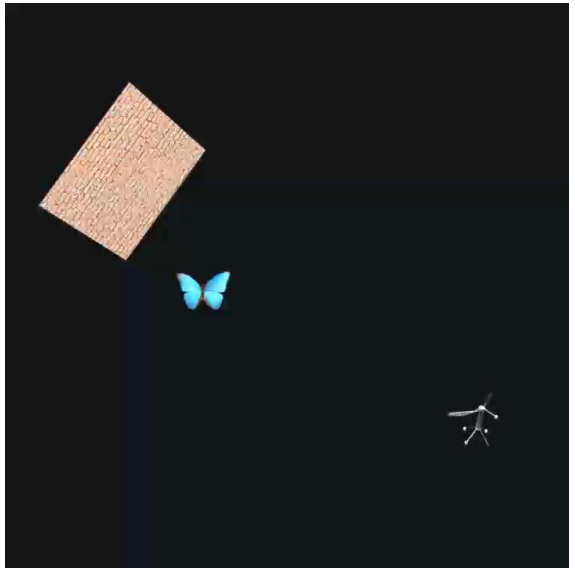
Camera View



Neuromorphic Camera



World View



Neuromorphic Motion Detection

- Pixel intensity can be recovered by integrating the event rate

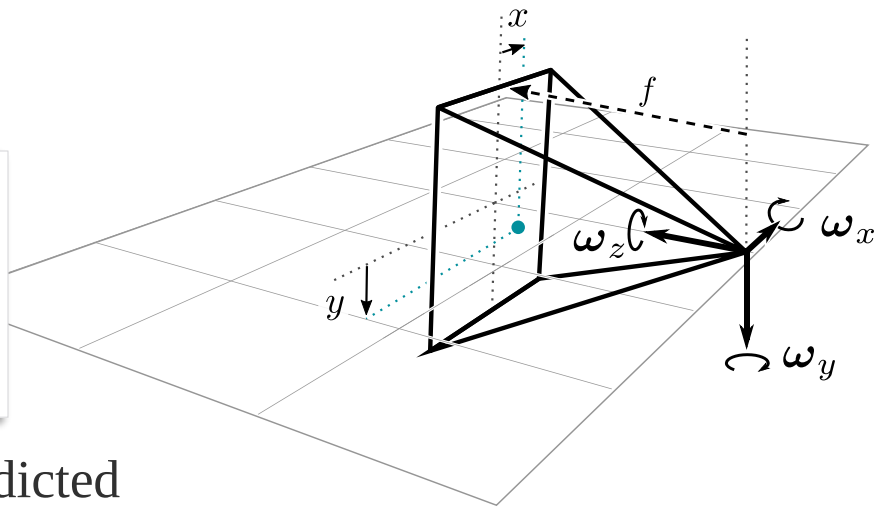
$$I(x,y,t) = \exp\left(\theta \int_0^t f(x,y,\tau)d\tau\right) + I(x,y,0)$$

- By previous assumptions, future pixel intensity can be predicted if image-plane motion field (v_x, v_y) is known

- Motion field due to camera rotation is

$$v_x(x,y) = -\frac{x^2}{f}\omega_y(t) + \frac{xy}{f}\omega_x(t) - f\omega_y(t) + y\omega_z(t)$$

$$v_y(x,y) = -\frac{xy}{f}\omega_y(t) + \frac{y^2}{f}\omega_x(t) - x\omega_z(t) + f\omega_x(t)$$



- Pixel intensity after a short time Δt is predicted from motion field:

$$\tilde{I}(x,y,t) = I(x - v_x\Delta t, y - v_y\Delta t, t - \Delta t)$$

Neuromorphic Motion Detection Results

1. Compute difference between predicted and measured intensity

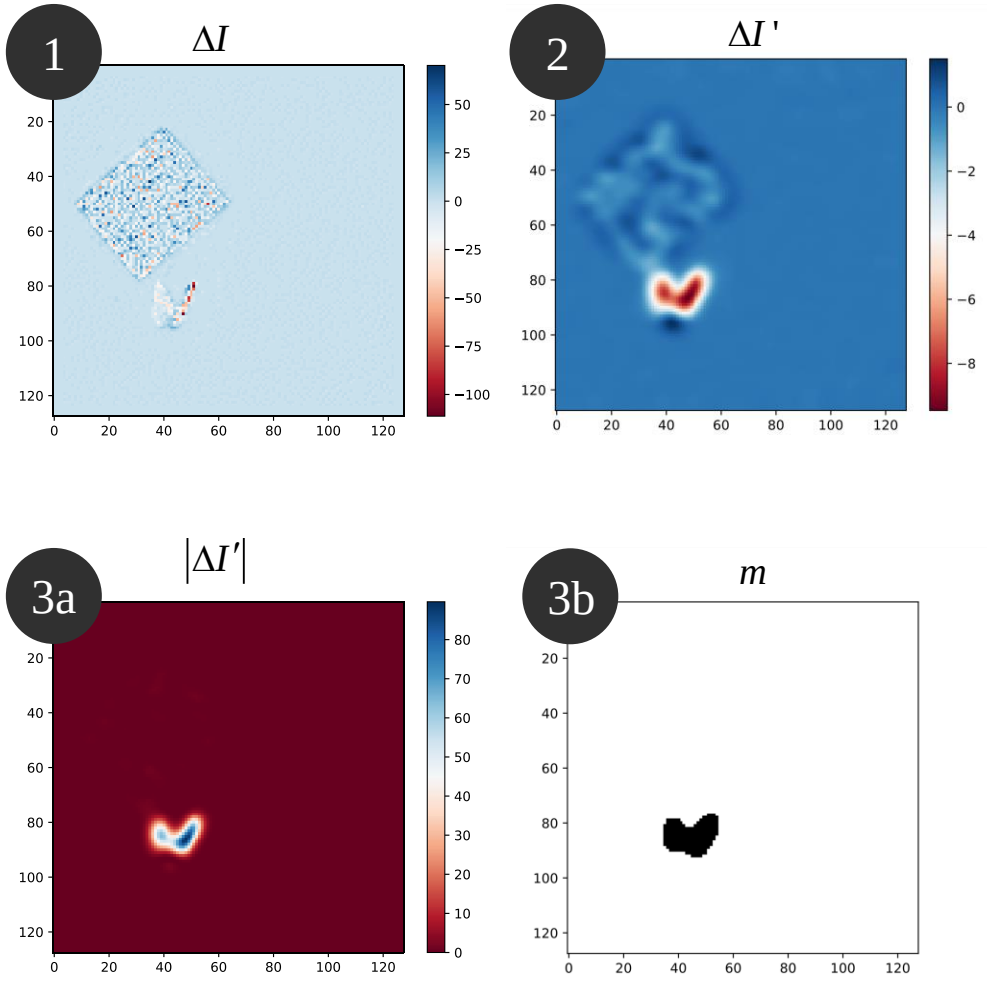
$$\Delta I(x,y,t) = I(x,y,t) - \tilde{I}(x,y,t)$$

2. Denoise by convolving with a multivariate Gaussian kernel K_Σ with covariance Σ

$$\Delta I'(x,y,t) = K_\Sigma(x,y,t) * I(x,y,t)$$

3. Detect motion by comparing smoothed intensity difference with a threshold γ

$$m(x,y,t) = \begin{cases} 1, & \text{if } |\Delta I'(x,y,t)| > \gamma. \\ 0, & \text{otherwise.} \end{cases}$$



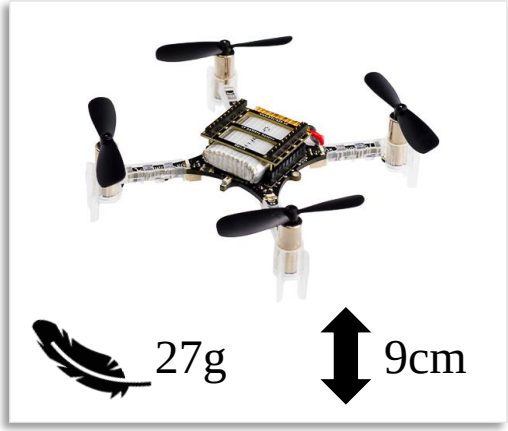
Summary

Innovation

- Neuromorphic sensing and control algorithms for intelligent, energy-efficient autonomy

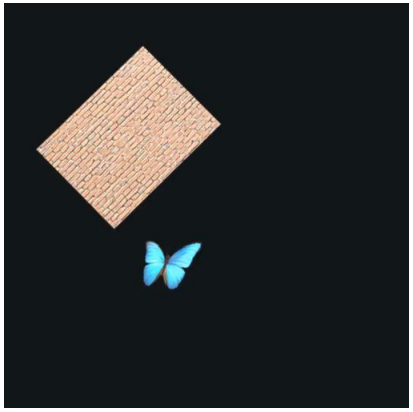
Potential Applications

- Autonomous flight for small-scale UAVs
  - Real-time target detection
  - Obstacle avoidance

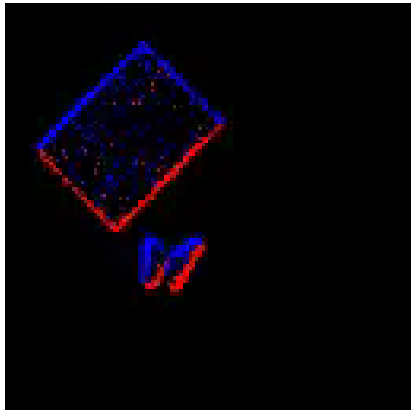


Crazyflie 2.0 (https://www.bitcraze.io/crazyflie-2/)

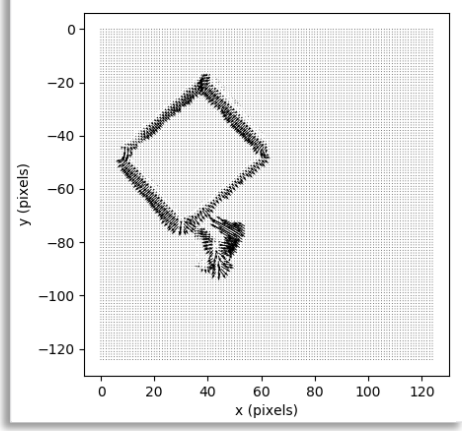
Standard Camera



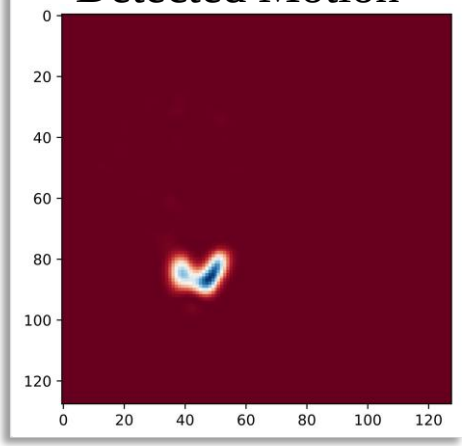
Neuromorphic Camera



Direction of Motion



Detected Motion



LISC
LABORATORY FOR INTELLIGENT SYSTEMS AND CONTROLS

Neuromorphic Sensing and Control of Autonomous Micro-Aerial Vehicles

Taylor Clawson

June 12, 2018

Related Work

T. S. Clawson, S. Ferrari, S. B. Fuller, R. J. Wood, "Spiking Neural Network (SNN) Control of a Flapping Insect-scale Robot," *Proc. of the IEEE Conference on Decision and Control*, Las Vegas, NV, pp. 3381-3388, December 2016.

T. S. Clawson, S. B. Fuller, R. J. Wood, S. Ferrari "A Blade Element Approach to Modeling Aerodynamic Flight of an Insect-scale Robot," *American Control Conference (ACC),* Seattle, WA, May 2017.

T. S. Clawson, T. C. Stewart, C. Eliasmith, S. Ferrari "An Adaptive Spiking Neural Controller for Flapping Insect-scale Robots," *IEEE Symposium Series on Computational Intelligence (SSCI)*, Honolulu, HI, December 2017.