Neuromorphic Sensing and Control of Bio-Inspired Robots

Taylor S. Clawson[1]

Advisor: Silvia Ferrari, Professor[1]

Collaborators: Robert J. Wood, Professor[2], and Rebecca Steinmeyer[2]

[1]Department of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY
[2]John A. Paulson School of Engineering and Applied Sciences and the Wyss Institute for Biologically Inspired Engineering, Harvard University, Cambridge, MA

Biocene
Cleveland, OH
August 15, 2018

Neuromorphic Systems Background

Neuromorphic Vision Sensors

- Produce asynchronous events based on pixel-level changes in brightness
- Pixels are sensitive to log of brightness, resulting in high dynamic range (~120 dB)
- Eliminates redundant data, allowing high temporal resolution (1 μs precision) and low latency (at most 1 ms)
- Sparse output: reduced computational cost
- Low power consumption (~3 mW)
- Does not measure absolute brightness

Brandli, C., et al. (2014). IEEE J. Solid-State Circ. (2014)
Lichtsteiner, P., et al. (2008). IEEE J. Solid-State Circ. (2008)
M. Mahowald, Springer Science & Business Media. (1994)
E. Culurciello and A. G. Andreou, Analog Integrated Circuits and Signal Processing (2006)
K. A. Zaghloul and K. Boahen, IEEE Transactions on Biomedical Engineering (2004)

Neuromorphic Vision Sensors

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

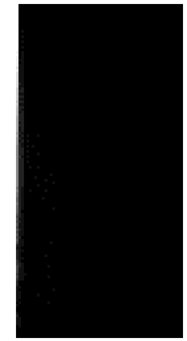
$$p_i = \begin{cases} 1, & \text{if } \ln(I(x,y,t_{i-1})) - \ln(I(x,y,t_i)) = -\theta \\ -1, & \text{if } \ln(I(x,y,t_{i-1})) - \ln(I(x,y,t_i)) = \theta \end{cases}$$

- The ith event $\mathbf{e}_i$ is described by the tuple $\mathbf{e}_i = (x, y, t, p)_i$

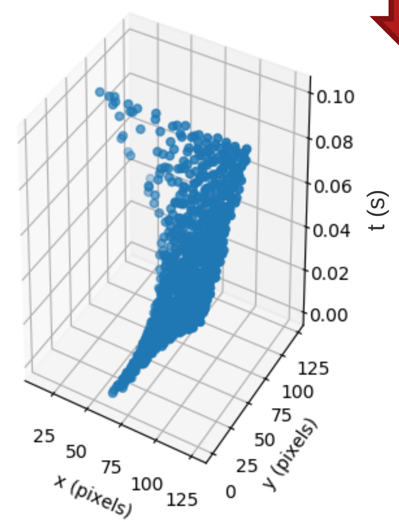
$$x, y \in \mathbb{N}^+ \qquad t \in \mathbb{R}^+ \qquad p \in \{-1, 1\}$$

- The set of all events is

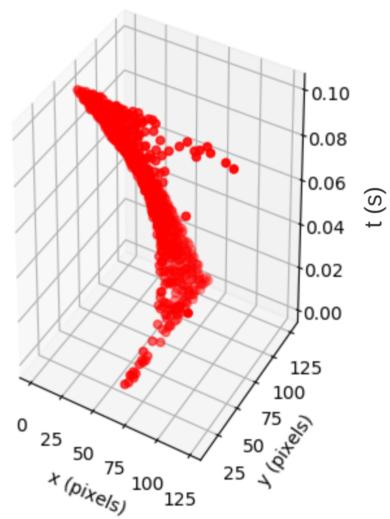
$$\mathcal{E} = \{\mathbf{e}_i \mid i = 1, \ldots, N\}$$



Source



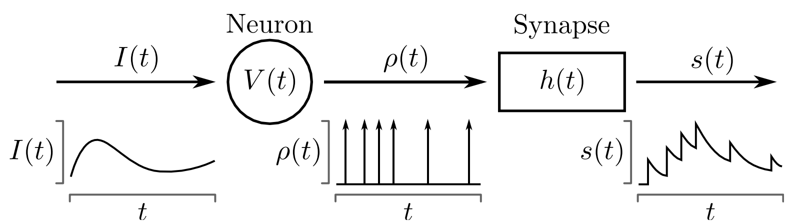
"On" Events
($p_i = 1$)



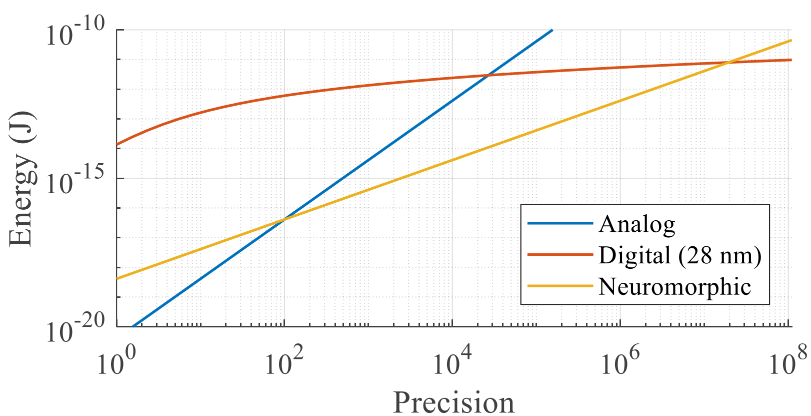
"Off" Events
($p_i = -1$)

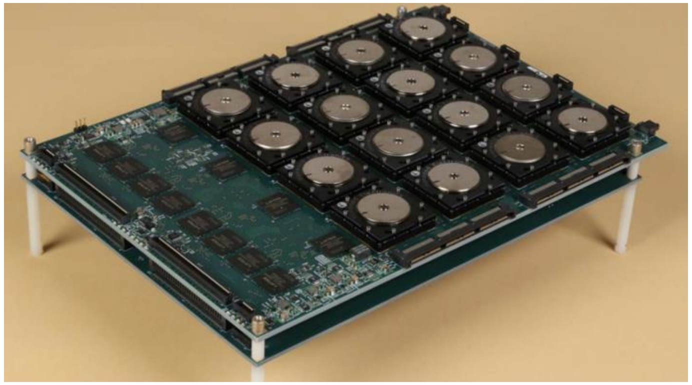
Neuromorphic Chips: Digital and Analog

- Neuromorphic chips compute with networks of spiking neurons connected through synapses

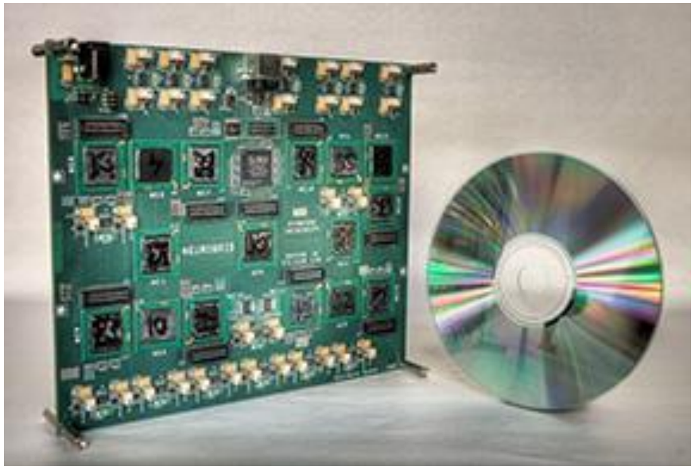


- Many neuromorphic chip designs are more power efficient than alternatives across 5 decades of precision [Boahen, 2017]





DARPA SyNAPSE Board with IBM TrueNorth Chips

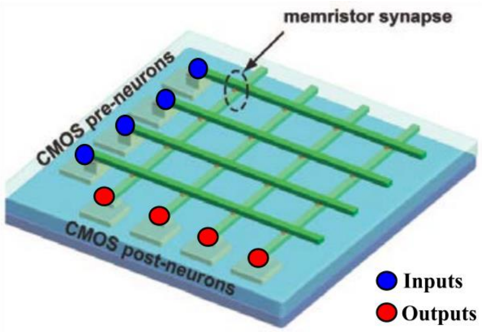


Neurogrid [Benjamin, 2014]

Images: K. Boahen https://web.stanford.edu/group/brainsinsilicon/neurogrid.html#Choices; DARPA http://www.darpa.mil/NewsEvents/Releases/2014/08/07.aspx

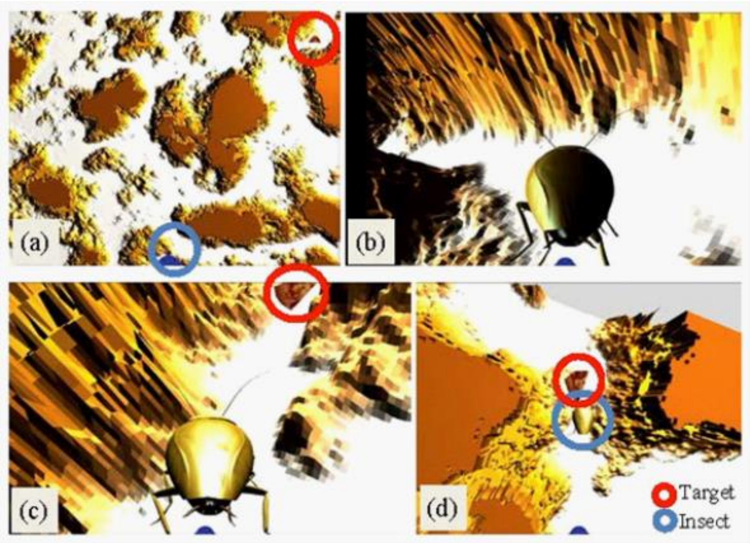
Neuromorphic Chips: Memristors

- Memristors (memory + resistor) are devices whose resistance can be modulated by the charge passing through them



- Memristors mimic Spike-Timing Dependent Plasticity (STDP) in biological synapses, which can be used to train neural networks [Hu, 2014]

- Memristor-based neuromorphic chips have been used to demonstrate control of a walking insect on simulated hardware [Mazumder, 2016]



D. Hu, X. Zhang, Z. Xu, S. Ferrari, and P. Mazumder, "Digital implementation of a spiking neural network (SNN) capable of spike-timing-dependent plasticity (STDP) learning," *Proc. of the IEEE 14th International Conference on Nanotechnology (IEEE NANO)*, 2014.

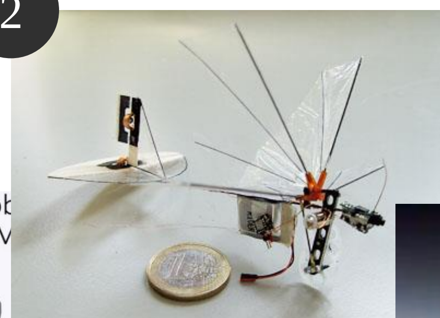
P. Mazumder, D. Hu, I.Ebong, X. Zhang, Z. Xu, and S. Ferrari, "Digital implementation of a virtual insect trained by spike-timing dependent plasticity," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2016.

Small Autonomous Robots

Small Bio-inspired Autonomous Robots

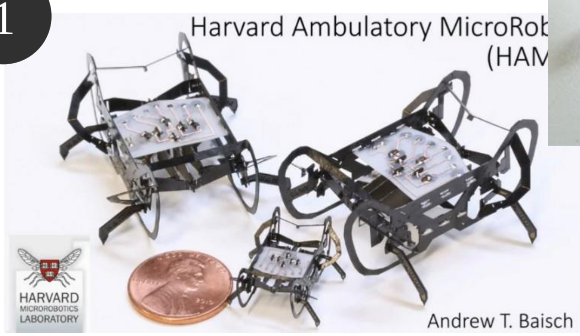
- Safer: weigh a few grams or less and thus are safe to operate near humans
- Smaller: can access narrow or confined spaces inaccessible to other vehicles
- Efficient: flapping flight is more efficient at this scale than in larger vehicles

2



1. HAMR [Baisch, 2014]
2. DelFly [deCroon, 2012]
3. RoboBee [Ma, 2013]

1

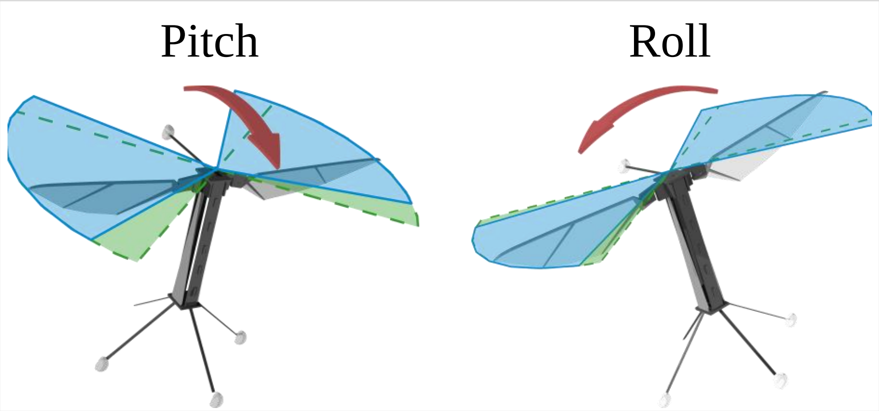


3

- Available sensors limited by total vehicle weight of less than a few grams
- Sensitive to environmental disturbances and physical parameter variations

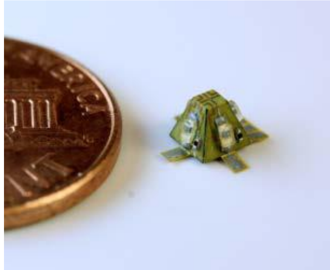
The RoboBee

- Thrust and body torques controlled by modulating wing stroke angle
- Several biologically-inspired sensors have been developed for autonomous flight

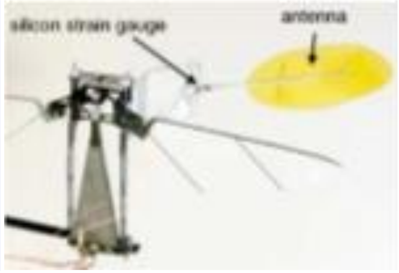


Available Sensors

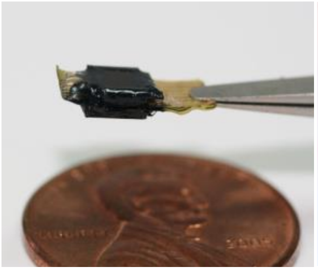
Ocelli



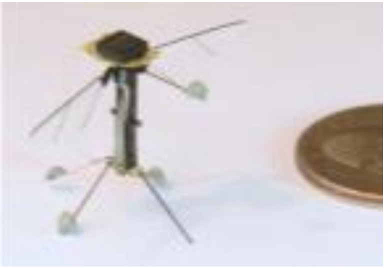
Antenna



Optical Flow



Six-axis IMU



Video and images courtesy of the Harvard Microrobotics Lab

Neuromorphic Sensing and Control

Single-layer SNN Controller

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

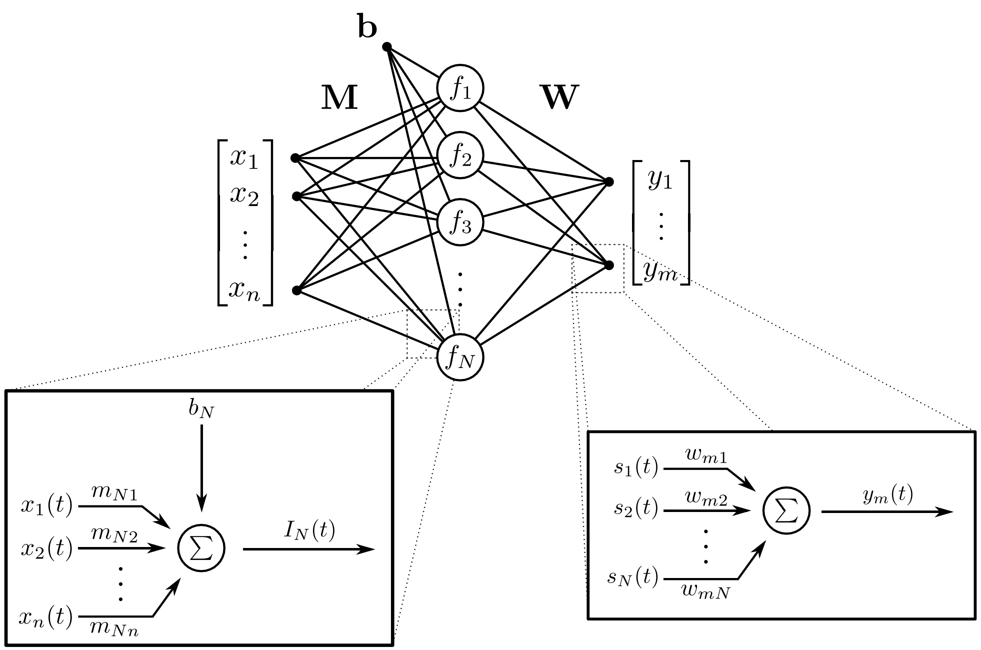
$$\mathbf{y}(t) = \mathbf{W}\mathbf{s}(t) = \mathbf{W}F(\mathbf{M}\mathbf{x}(t) + \mathbf{b})$$

- 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 PIF controller)

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



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

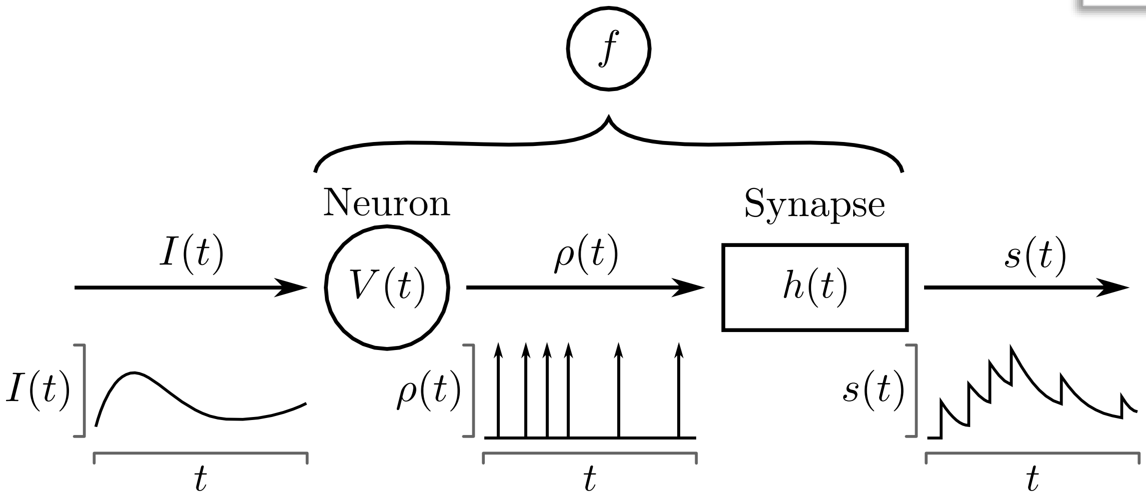
SNN Control Model

- Neurons generate spike trains *ρ(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)*

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

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

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



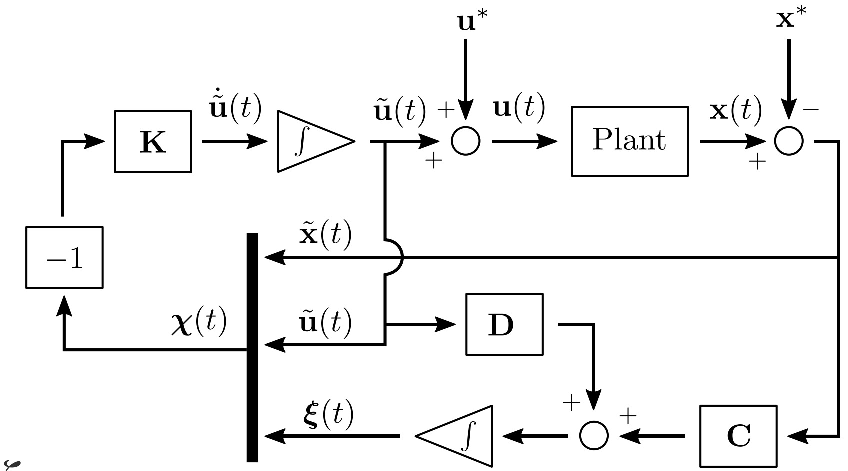
δ	Dirac delta
t_k	Time of kth spike
M	Spike count
τ_s	Synaptic time constant

PIF Compensator

- PIF control law is used as the target function for training an SNN offline

- Optimal linear controller guaranteed to stabilize linear system

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$$
$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$$



- Control law proportional to error in state $\mathbf{x}$, control $\mathbf{u}$, and integral of output ξ

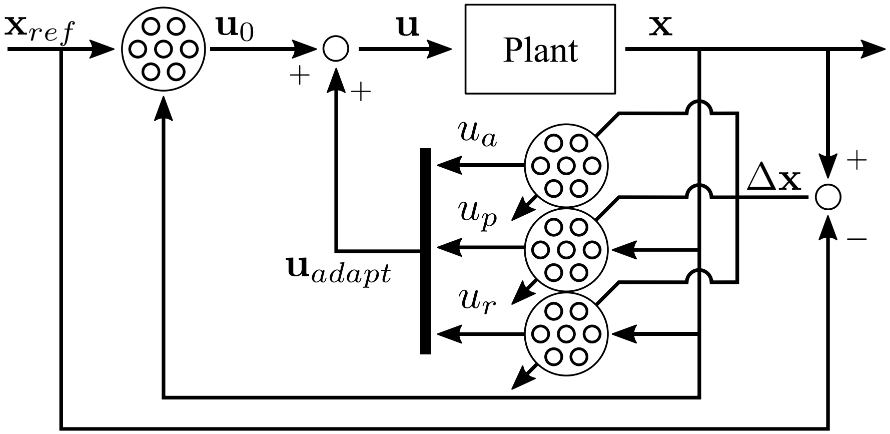
$$\mathbf{v}(t) \triangleq \dot{\mathbf{u}}(t) = -\mathbf{K}\chi(t)$$
$$= -\mathbf{K}_1\mathbf{x}(t) - \mathbf{K}_2\mathbf{u}(t) - \mathbf{K}_3\xi(t)$$

$\mathbf{x}^*$	State set point
$\mathbf{u}^*$	Control set point
χ	Augmented state vector
$\mathbf{v}$	Control rate of change

- The control law minimizes the quadratic cost J

$$J = \lim_{t_f \to \infty} \frac{1}{2t_f} \int_0^{t_f} \{\chi^T(t)\mathbf{Q}'\chi(t) + \mathbf{v}^T(t)\mathbf{R}'\mathbf{v}(t)\} dt$$

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 PIF 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]

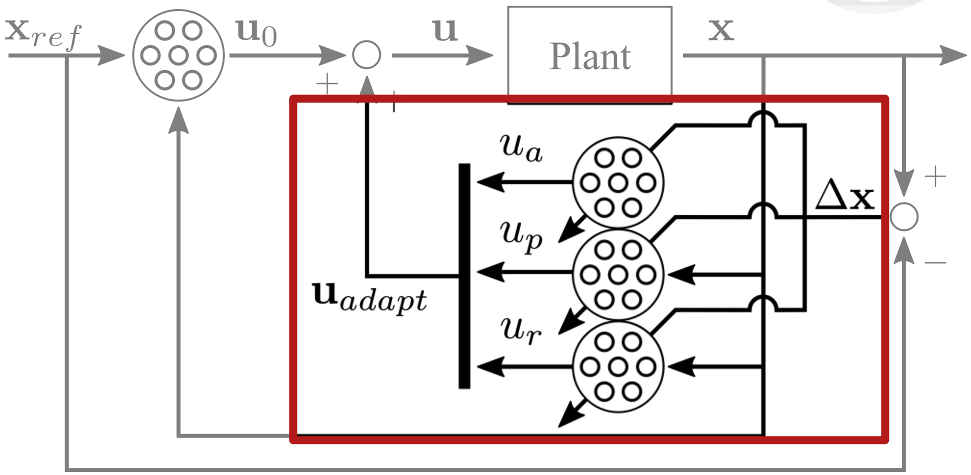
Adaptive SNN Controller (Hovering Only)

- 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

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

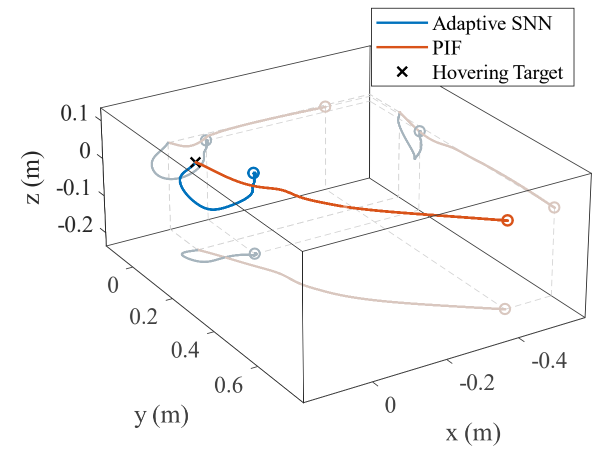
$\mathbf{\Lambda}$	Error weight matrix
Δx	State error
$\mathbf{W}_a$	Output connection weights
$\mathbf{s}_a$	Post-synaptic current
γ	Learning rate

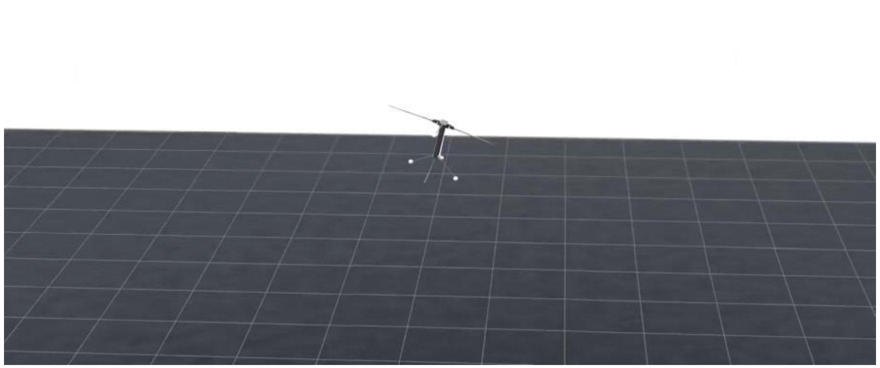
[T. S. Clawson, T. C. Stewart, C. Eliasmith, S. Ferrari "An Adaptive Spiking Neural Controller for Flapping Insect-scale Robots," *IEEE Symposium Series on Computational Intelligence (SSCI)*, Honolulu, HI, December 2017]

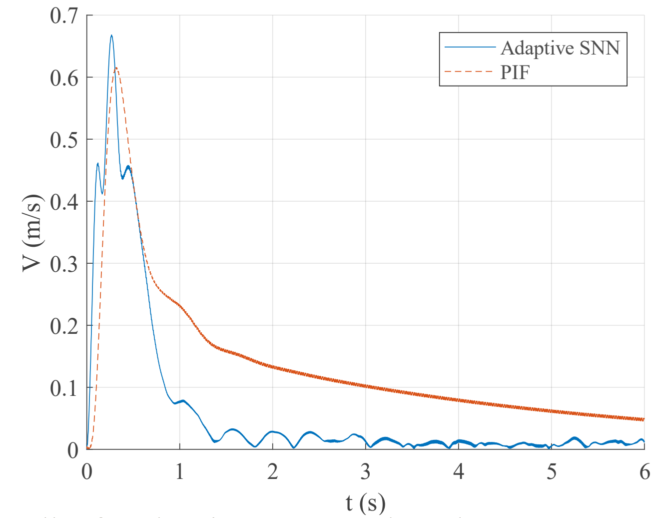
Adaptive SNN Controller (Hovering Only)

Comparison:

- SNN initialized with PIF
- SNN controller quickly adapts to asymmetries in the wings to stabilize hovering flight
- PIF compensator maintains stability, but drifts significantly from the origin







[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 PIF
- PIF 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

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

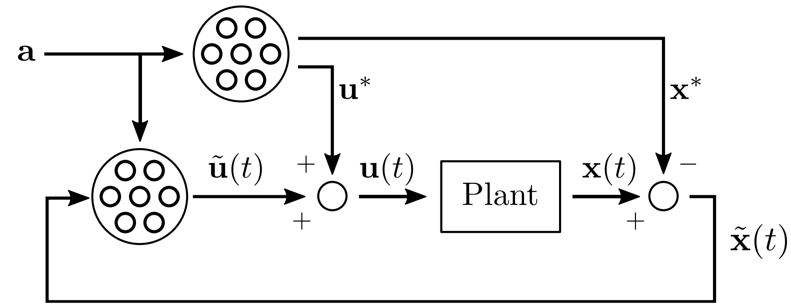
$$\mathbf{G}(0) = -\mathbf{K}(\mathbf{a})_2^{-1}\mathbf{K}_1(\mathbf{a}) \triangleq \mathbf{K}_{ss}(\mathbf{a})$$

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

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

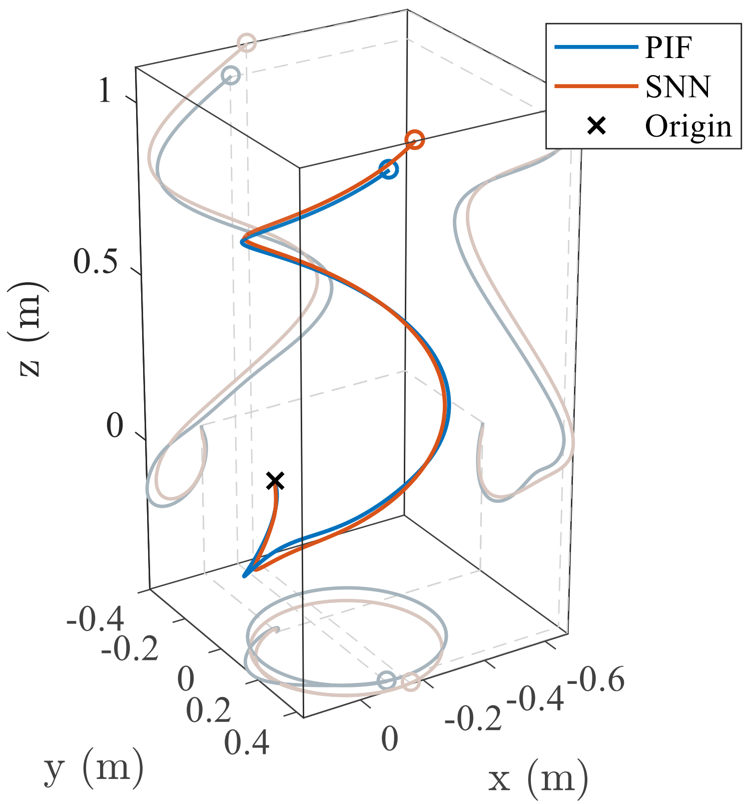
- SNN Control input is a linear transformation of post-synaptic current

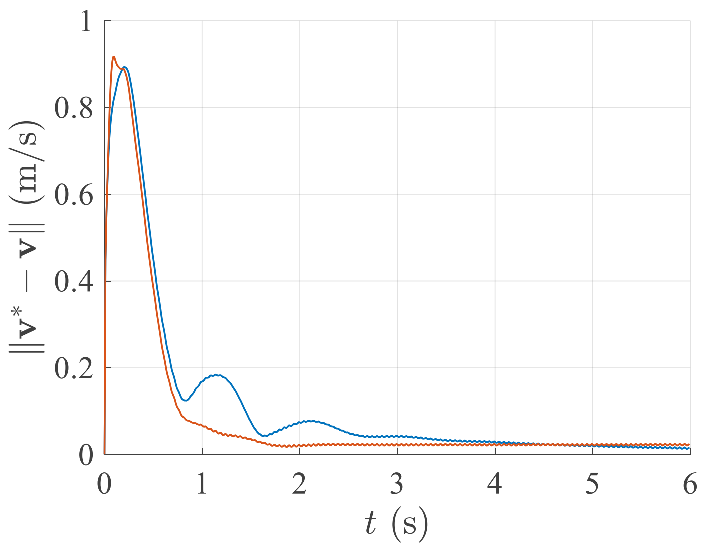
$$\mathbf{u}(t) = \mathbf{W}(\mathbf{a})\mathbf{s}(t)$$

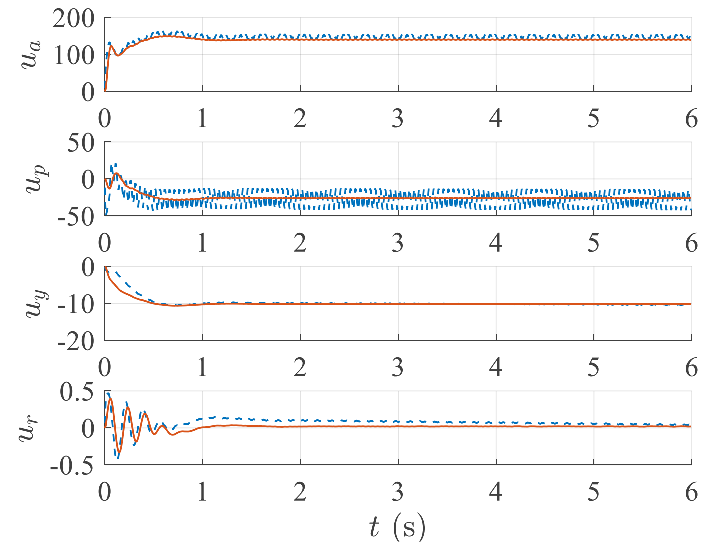


$\mathbf{K}_i$	PIF gain matrices	$\mathbf{x}$	State deviation
$\mathbf{u}$	Control deviation	ξ	Integral of output error
$\mathbf{a}$	Scheduling variables	$\mathbf{G}(s)$	Transfer function
s	Laplace variable	$\mathbf{K}_{ss}$	Steady-state gain matrix
$\mathbf{W}$	Output connection weights	$\mathbf{s}$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

SNN Control – Climbing Turn







Exteroceptive Sensing

Exteroceptive Sensing Motivation

- Onboard exteroceptive sensors required for full flight autonomy
- Fast dominant time scales of insect-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)

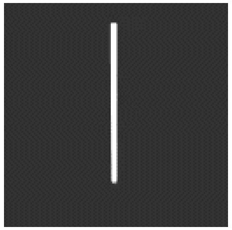
The Optical Flow Problem

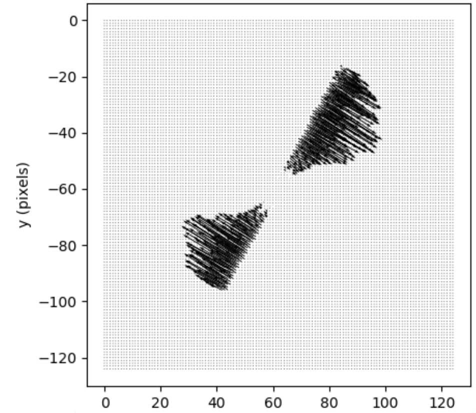
Standard Optical Flow Problem

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

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

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



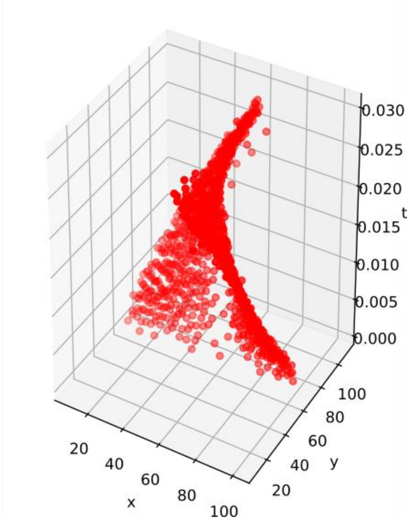


Neuromorphic Optical Flow

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

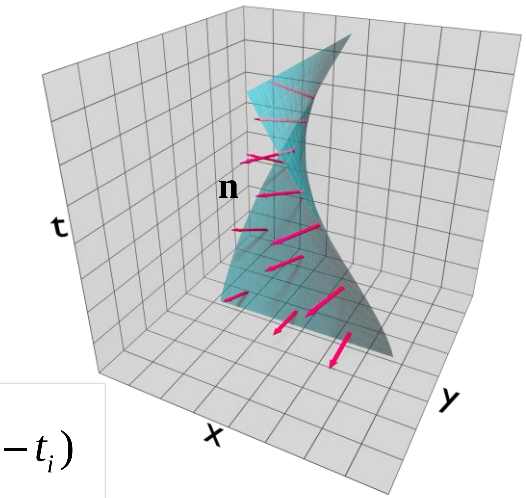
$$\begin{bmatrix} r_x(t_2) - r_x(t_1) \\ r_y(t_2) - r_y(t_1) \end{bmatrix} = \int_{t_1}^{t_2} \mathbf{v}(\tau) d\tau \approx \begin{bmatrix} v_x dt \\ v_y dt \end{bmatrix}, \qquad \mathbf{v}(\tau) = \begin{bmatrix} v_x(\tau) \\ v_y(\tau) \end{bmatrix}$$

- 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



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

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

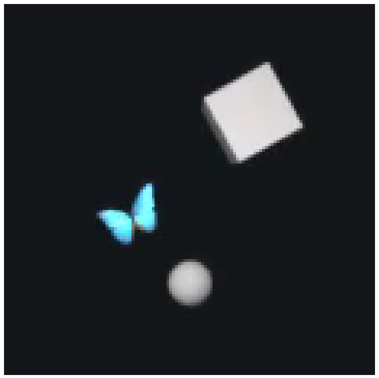
- Assume gradient **n** of event rate is normal to the motion of points in the scene
- Speed of the motion is inversely proportional to magnitude of gradient
- Optical flow is written directly in terms of the event rate gradient

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

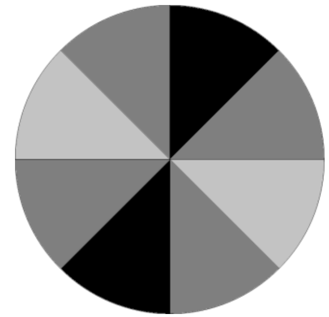
$$\mathbf{w} = \begin{bmatrix} \frac{\partial t}{\partial x} & \frac{\partial t}{\partial y} \end{bmatrix}^T = \begin{bmatrix} -\frac{a}{c} & -\frac{b}{c} \end{bmatrix}^T$$

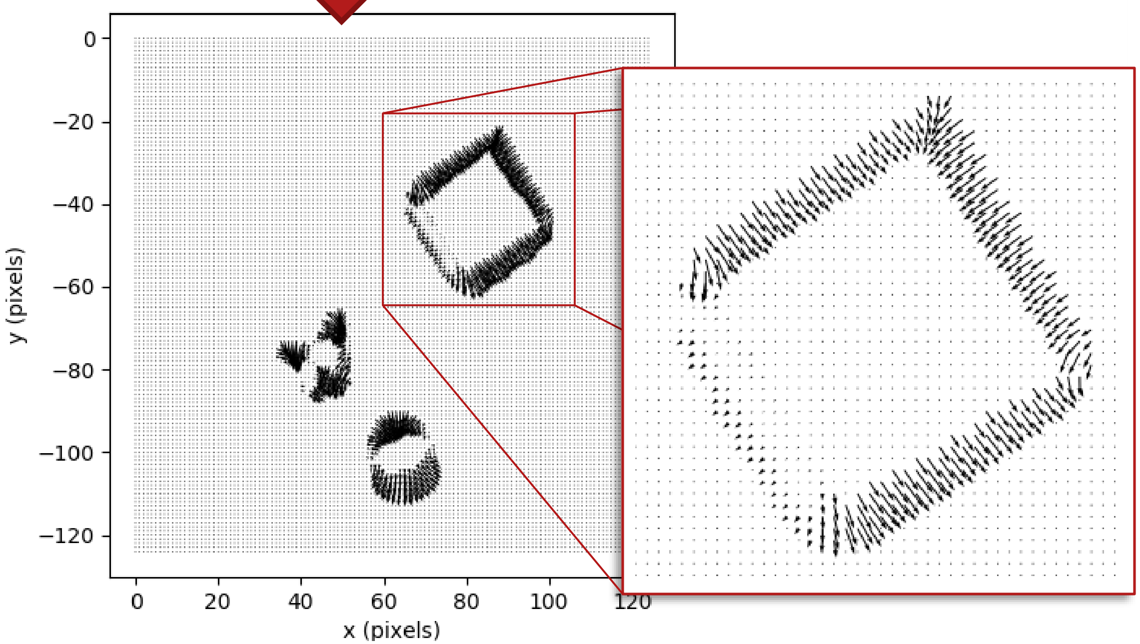
$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \left(\frac{1}{\|\mathbf{w}\|} \right) \frac{\mathbf{w}}{\|\mathbf{w}\|} = -\frac{c}{a^2 + b^2} \begin{bmatrix} a \\ b \end{bmatrix}$$

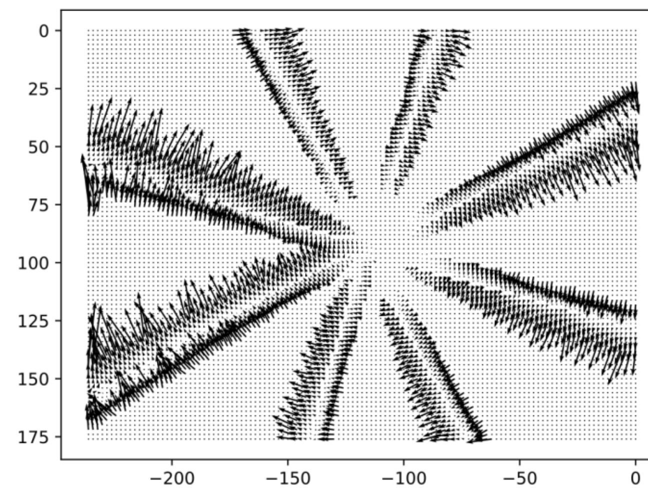
Neuromorphic Optical Flow Results



Mean processing time per event: 0.7 μs







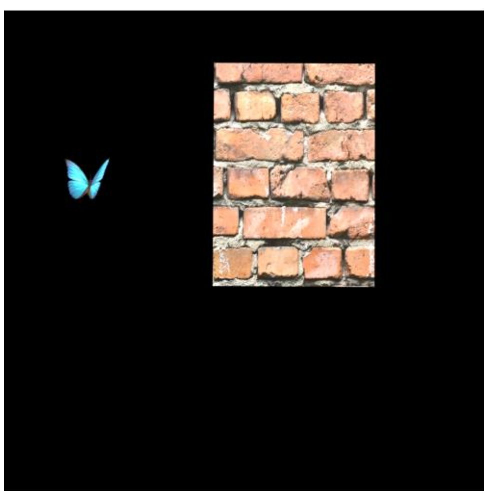
Independent Motion Detection

Detect motion relative to the environment using a rotating and translating neuromorphic camera

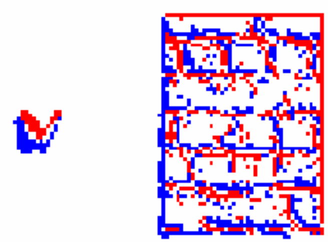
Assumptions

- Known camera motion
- Total derivative of pixel intensity is zero

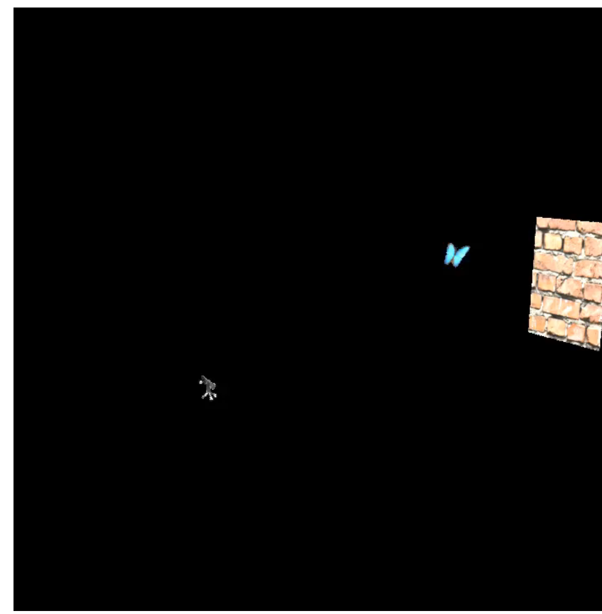
Camera View



Neuromorphic Camera



World View



Motion Detection: Known Camera Motion

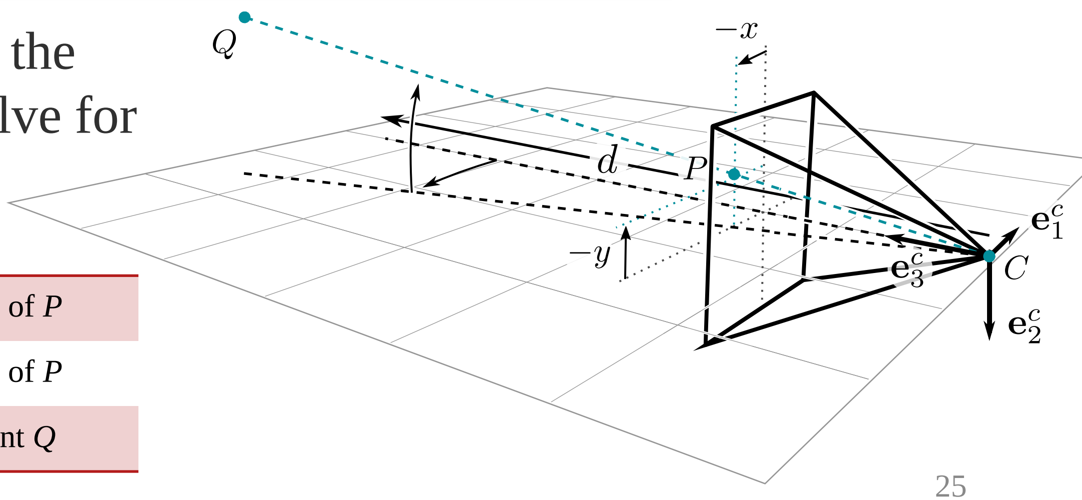
- By previous assumptions, depth of the point can be estimated if image-plane motion field (v_x, v_y) is known

- For some stationary point in the environment Q, its projection P onto the image plane will move with velocity $\dot{\mathbf{p}} = \dot{p}_x \mathbf{e}_1^c + \dot{p}_y \mathbf{e}_2^c$ where

$$\dot{p}_x(x,y,t) = \frac{1}{d}(\lambda v_x(t) - x v_z(t)) + y\omega_z(t) - \lambda\omega_y(t) - \frac{1}{\lambda}(x^2\omega_y(t) + xy\omega_x(t))$$

$$\dot{p}_y(x,y,t) = \frac{1}{d}(\lambda v_y(t) - y v_z(t)) - x\omega_z(t) + \lambda\omega_x(t) + \frac{1}{\lambda}(y^2\omega_x(t) - xy\omega_y(t))$$

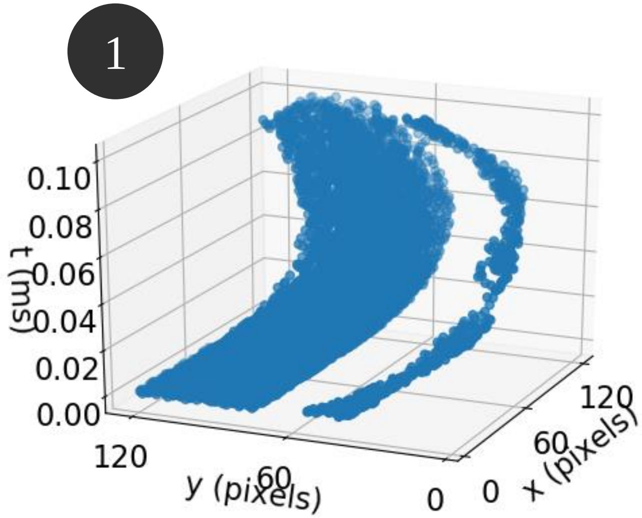
- With an additional constraint on the velocity of P, it is possible to solve for the depth d

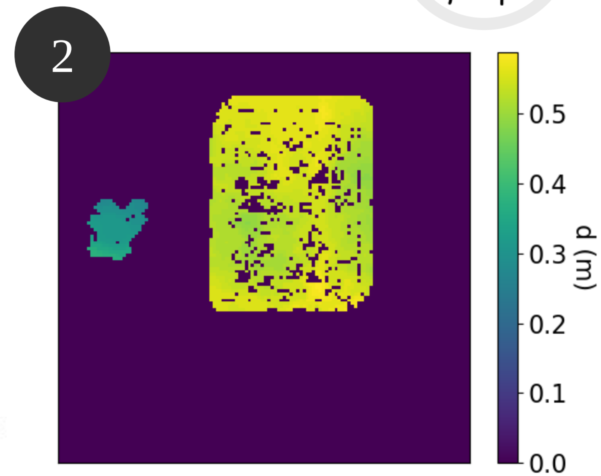


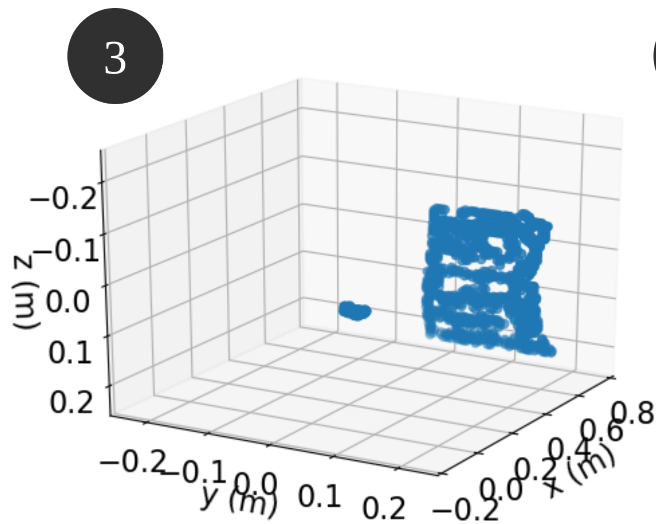
λ	Focal length	x	x-coordinate of P
$\boldsymbol{\omega}$	Camera angular rate	y	y-coordinate of P
$\mathbf{v}$	Camera velocity	d	Depth of point Q

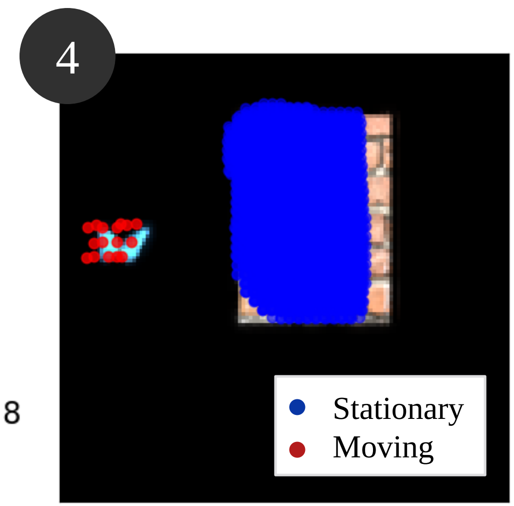
Motion Detection and Depth Estimation

1. Events can be visualized by their spatio-temporal coordinates
2. Depth d is computed from the events using known camera motion
3. Events are transformed into the world frame using estimated depth
4. A temporal low-pass filter separates moving and stationary objects





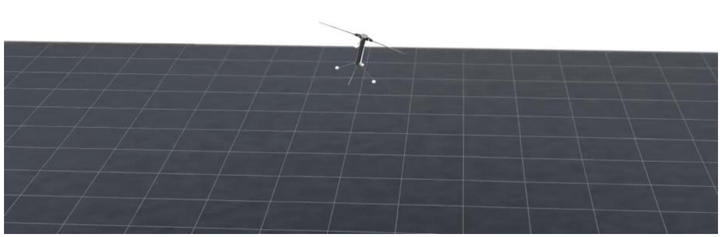




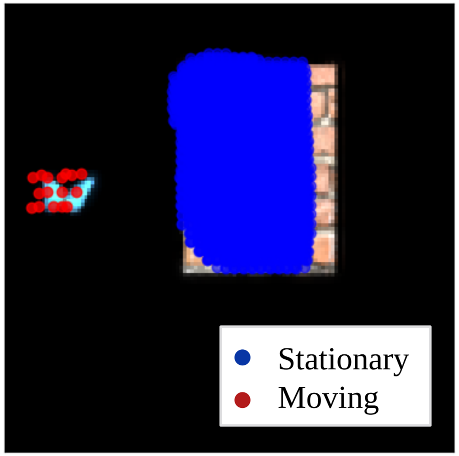
Conclusion and Future Work

Conclusions

- Small-scale robots provide many potential benefits, including safety, access to confined spaces, and power efficiency
- Enabling autonomy in small-scale robots requires high frequency sensing and control loops that operate on only a few milliwatts of power
- Adaptive spiking neural networks can control complex systems and can be directly implemented on power efficient hardware

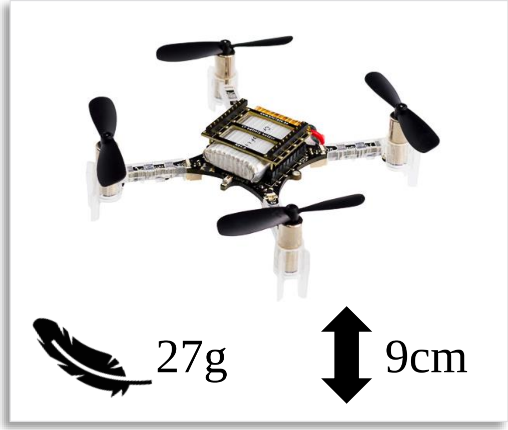


- Neuromorphic cameras can effectively detect motion and depth from moving platforms to enable efficient autonomous flight



Future Work

- Demonstrate control of maneuvers on the physical RoboBee using adaptive SNN controllers
- Use neuromorphic depth estimation and motion detection to perform obstacle avoidance and target tracking with the RoboBee in simulation
- Verify motion detection and depth estimation results with experiments on small quadcopters in flight
- Hardware-in-the-loop experiments with the physical RoboBee and simulated neuromorphic cameras to demonstrate autonomous target tracking and obstacle avoidance
- Control quadcopter flight using adaptive SNN flight control implemented on neuromorphic chips



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

LABORATORY FOR INTELLIGENT SYSTEMS AND CONTROLS

Neuromorphic Sensing and Control of Autonomous Micro-Aerial Vehicles

Taylor S. Clawson

Advisor: Silvia Ferrari, Professor

Collaborators: Robert J. Wood, Professor, and Rebecca Steinmeyer

Funding Source: ONR – n000141712614

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, 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.