LABORATORY FOR INTELLIGENT SYSTEMS AND CONTROLS

ONR Unmanned Maritime Systems Technology (UMST) Program Review, Miramar Beach, FL
January 29, 2019

Event-based Sensorimotor Planning and Control

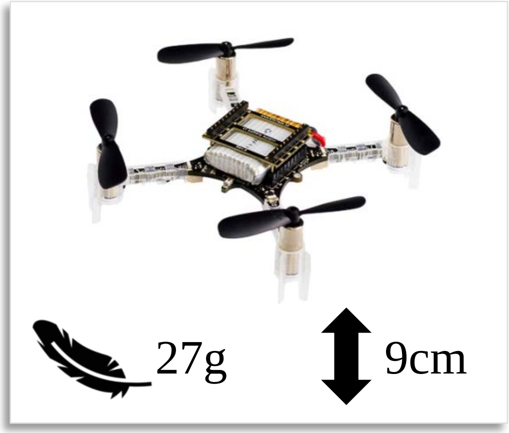
PIs: Silvia Ferrari♣ and Robert Wood♠

♣John Brancaccio Professor of Mechanical and Aerospace Engineering, Cornell
♠ Charles River Professor of Engineering and Applied Sciences, Harvard

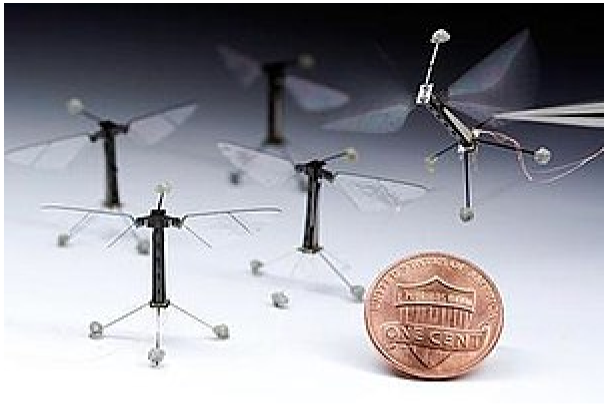
Ph.D. Student: Taylor S. Clawson

Motivation: Insect-Scale Autonomous Flight

- Safer: weigh less than 1 pound and thus are safe to operate near humans
- Smaller and covert: can access narrow or unfriendly spaces inaccessible to other vehicles
- Autonomous flight expands the capability of a single operator to monitor previously inaccessible spaces
  - More effective search and rescue
  - Surveillance in complex environments
  - Security in densely populated, sensitive regions



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



RoboBee [Ma, 2013]

Challenges: Insect-scale Sensorimotor Control

- Size, weight and power constraints
  - RoboBee power budget: ~21mW
  - Only ~2mW available for sensing and control
- Fast dynamics
  - Dominant timescales on the order of a few hundred milliseconds
- Physical parameter variations
  - Small wing asymmetries result in undesired torque during flight
- Highly susceptible to external disturbances such as wind gusts

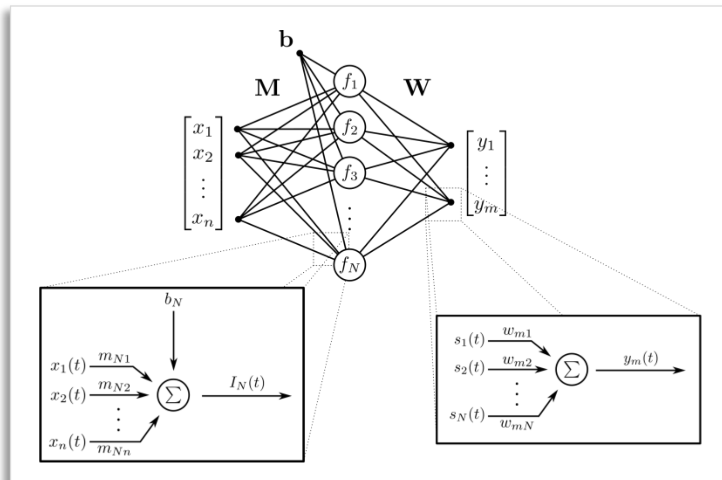
Open Loop Flight

Funded by the ONR Grant # N00014-17-1-2614

Neuromorphic Sensing and Control

Emerging Technologies

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



1

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

inivation (https://inivation.com/)

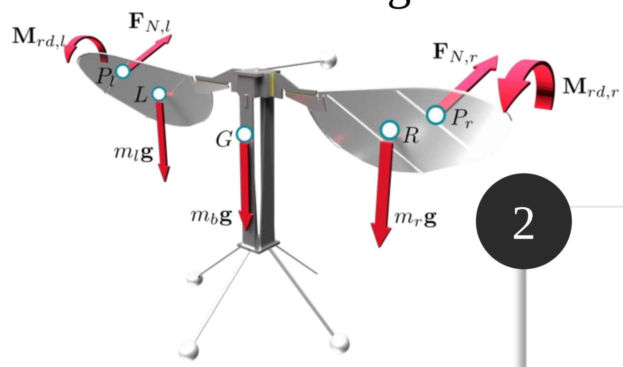
2

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

Research Goals

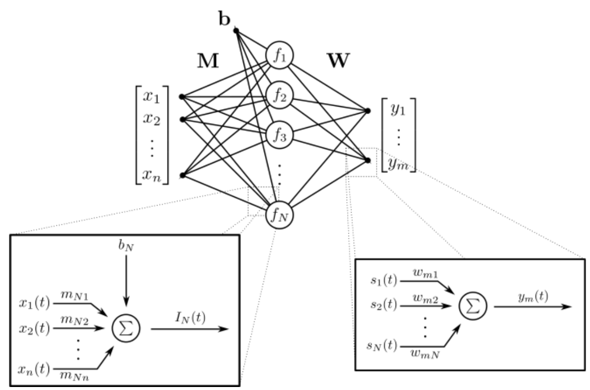
1

Modeling



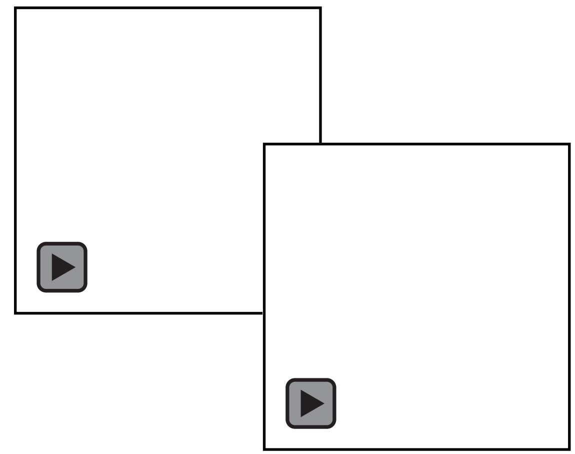
2

Adaptive Flight Control



3

Exteroceptive Sensing



1. Model the RoboBee flight dynamics, validate with experimental data
2. Develop adaptive flight controllers which account for physical variations
3. Develop sensing algorithms to perform target tracking and obstacle avoidance

Wing Modeling

Assumptions:

- Rigid wings with passive pitching dynamics
- No stroke-plane deviation

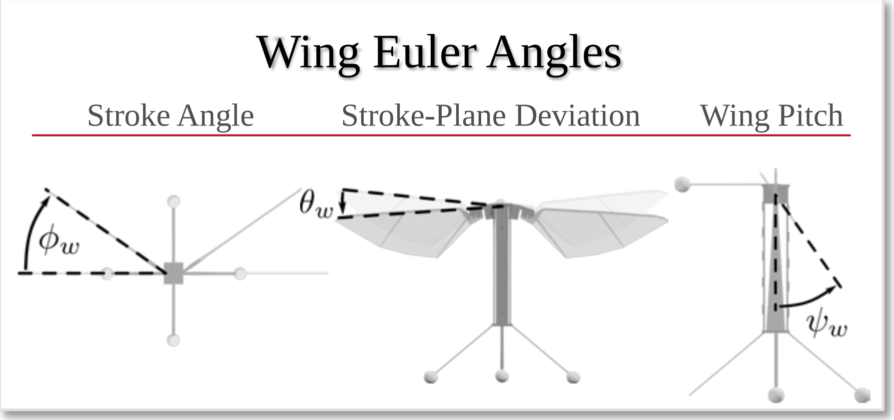
$$\theta_w = 0$$

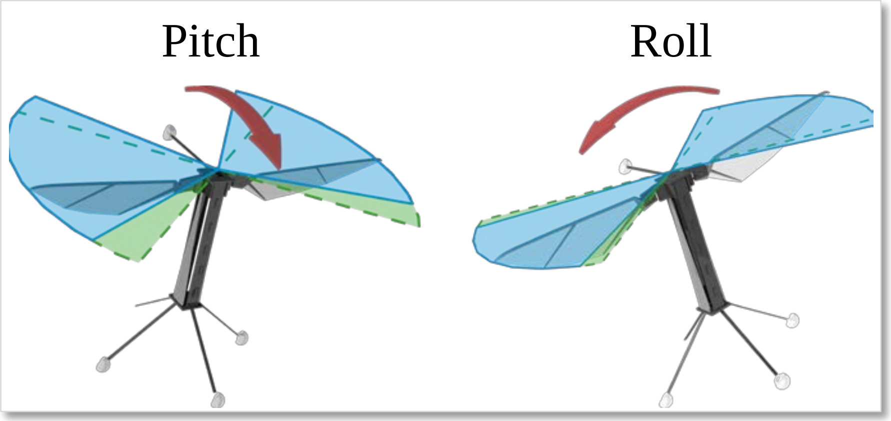
- Control inputs **u** affect stroke angle

$$\mathbf{u} = \begin{bmatrix} u_a & u_p & u_r \end{bmatrix}$$

- Stroke angle modeled by second order system

$$\ddot{\phi}_w(t) + 2\zeta\omega_n\dot{\phi}_w(t) + \omega_n^2\phi_w(t) = \frac{u_a \pm u_r}{2}\sin(\omega_f t) + u_p$$



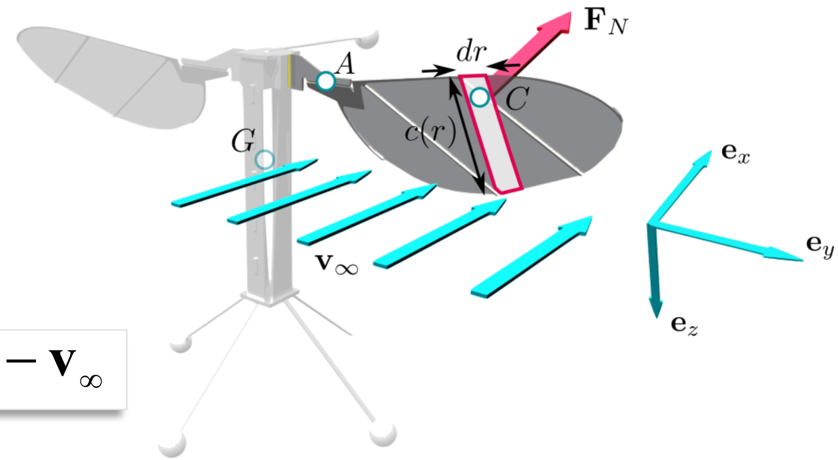


ζ	Effective damping ratio	ω_f	Effective natural frequency
ω_n	Forcing frequency	u_a	Flapping amplitude input
u_p	Pitch input	u_r	Roll input

Aerodynamic Forces and Moments

- Aerodynamic forces on wing caused by translational motion
- Locally, lift and drag are proportional to the square of the incident velocity $\mathbf{v}_C$

$$F_L(\alpha) = \frac{1}{2}\rho \int_0^R C_L(\alpha)\mathbf{v}_C^T \mathbf{v}_C c(r) dr$$



- Where

$$\mathbf{v}_C = \mathbf{v}_G + \boldsymbol{\omega}_b \times \mathbf{r}_{A/G} + \boldsymbol{\omega}_r \times \mathbf{r}_{C/A} - \mathbf{v}_\infty$$

$$C_L(\alpha) = C_{L_{max}} \sin(2\alpha)$$

- Rotational damping $\mathbf{M}_{rd}$ caused by span-wise rotation of wing

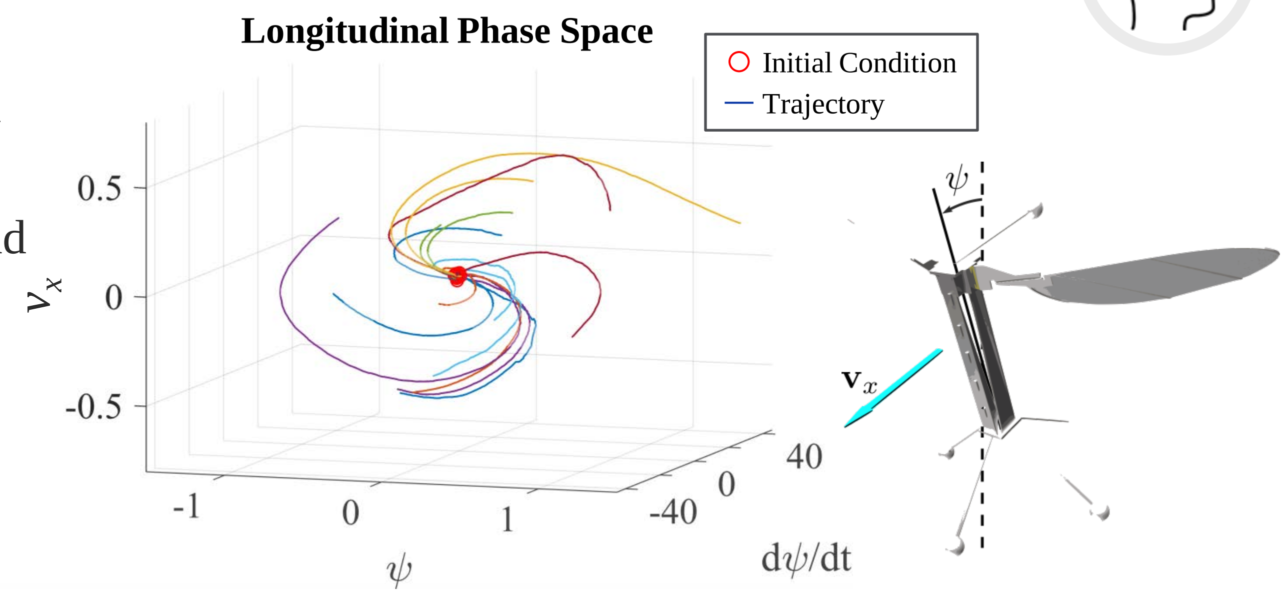
$$\mathbf{M}_{rd} = -\frac{1}{2}\rho C_{rd} \int_0^R \int_{z_0}^{z_1} (\boldsymbol{\omega}_y{}^2 z^2) |z| dz dr$$

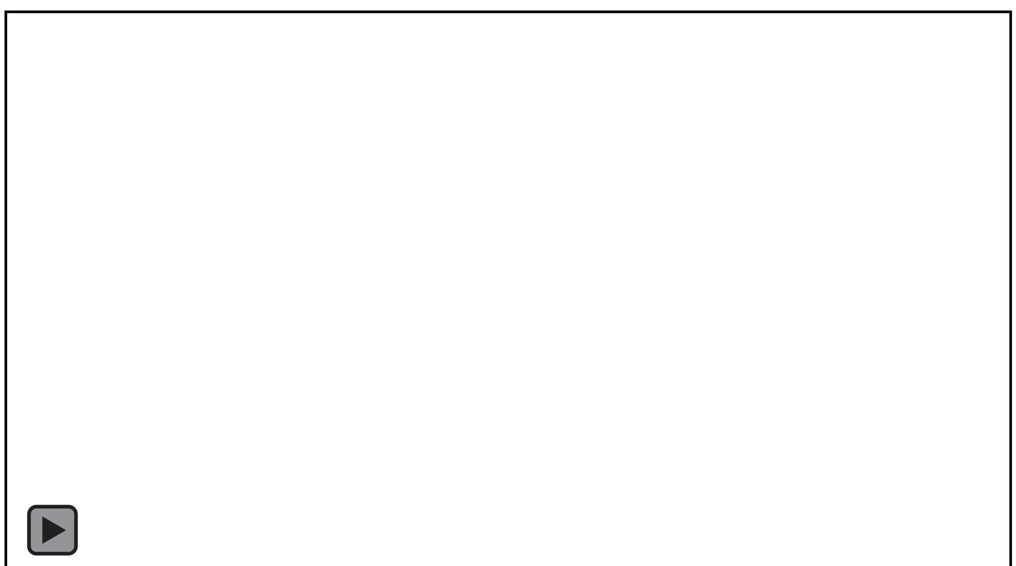
F_L	Lift force	α	Angle of attack
$\mathbf{r}_{A/G}$	Position of hinge relative to body CG	$\mathbf{r}_{C/A}$	Position of blade element relative to hinge
$\mathbf{v}_C$	Velocity of element	$\mathbf{F}_N$	Aerodynamic normal force
$\boldsymbol{\omega}_b$	Body angular rate	$\boldsymbol{\omega}_w$	Wing angular rate
$\mathbf{v}_\infty$	Free stream velocity	$c(r)$	Chord length
$\mathbf{M}_{rd}$	Rotational damping moment	C_L	Lift coefficient
C_{rd}	Rotational coefficient		

[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.]

Model Validation

- Validate model with open loop flight tests
- Dominant longitudinal and lateral modes visible in experimental data
- Model predicts the same dominant modes



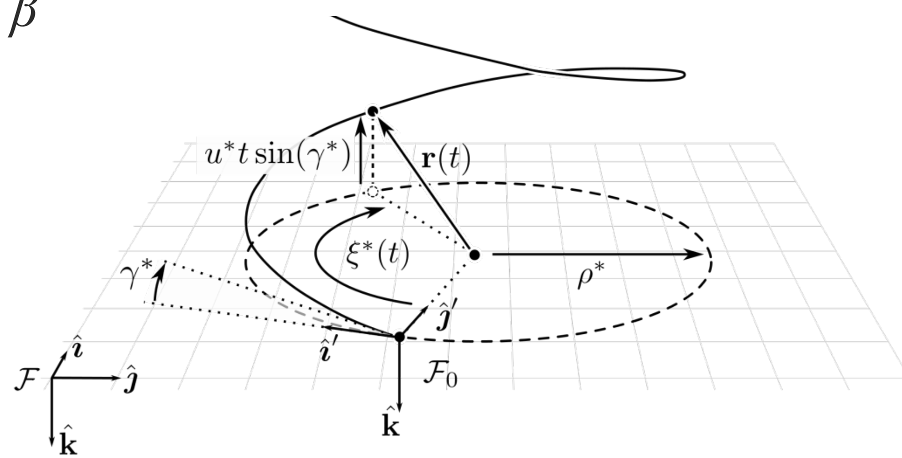


Steady Maneuvers

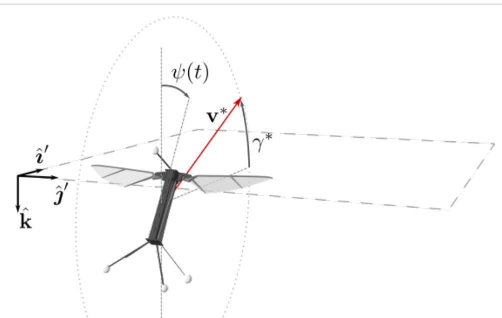
- *Steady* maneuvers are trajectories with minimum period equal to the flapping period T and constant control inputs
- Command input $\mathbf{y}^*$ defines maneuvers in terms of commanded speed u^*, climb angle γ^*, turn rate $\dot{\xi}^*$, and sideslip angle β^*

$$\mathbf{y}^* = \begin{bmatrix} u^* & \gamma^* & \dot{\xi}^* & \beta^* \end{bmatrix}$$

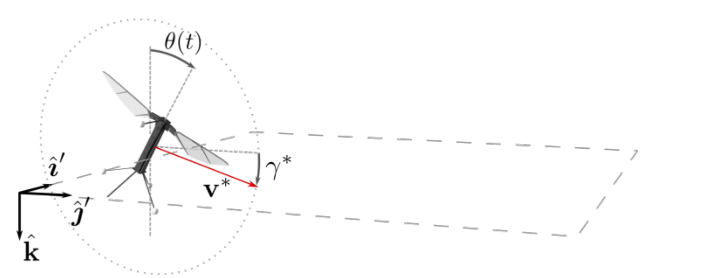
- The most general steady maneuver is the coordinated turn



- Other steady maneuvers include:



Longitudinal Flight



Lateral Flight

Steady Forward – Lateral Mode

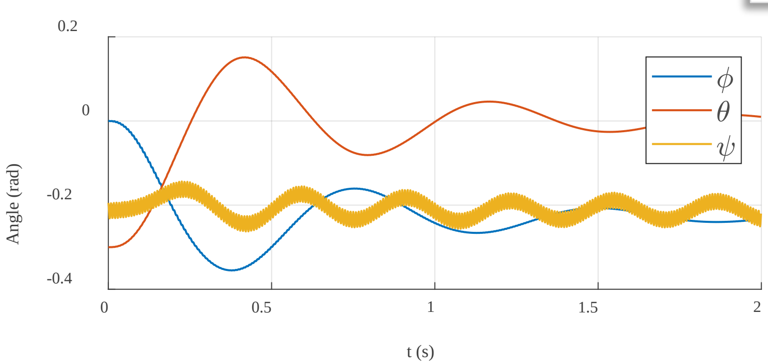
Lateral mode becomes stable in forward flight

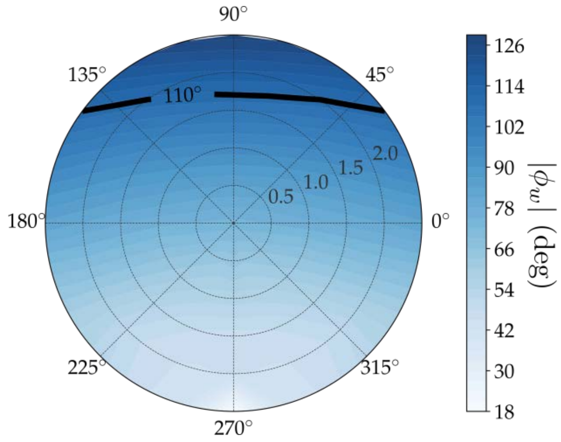
Lateral mode shape **v** shows coupling between yaw and roll

$$\mathbf{x} = \begin{bmatrix} \theta \\ \psi \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ v_x \\ v_y \\ v_z \end{bmatrix} \qquad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{bmatrix} = \begin{bmatrix} 0.044+0.001i \\ -0.002-0.001i \\ -0.070+0.039i \\ -0.045-0.322i \\ 0.002-0.022i \\ 0 \\ 0.027-0.035i \\ -0.002-0.002i \end{bmatrix}$$

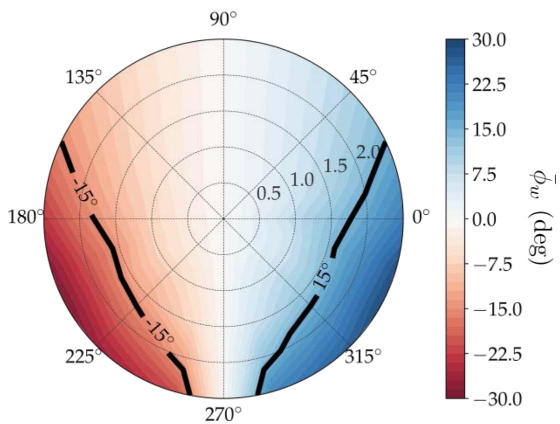
Time constant τ and frequency f of longitudinal mode:

$$\tau = 0.62\text{s}$$
$$f = 1.38\text{Hz}$$

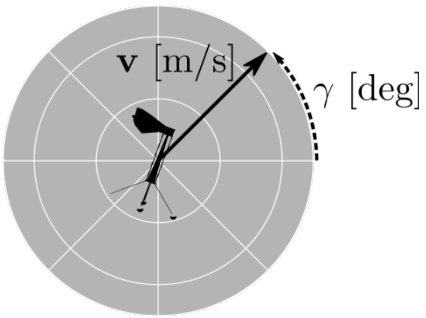


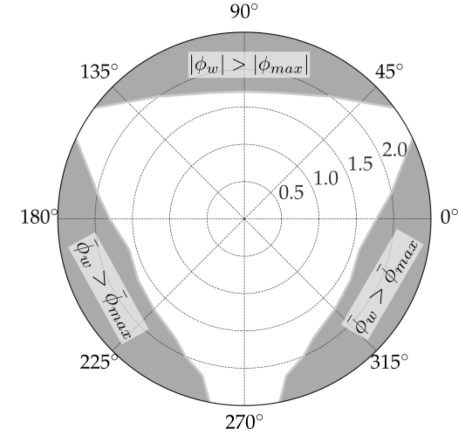


Peak-to-peak stroke amplitude



Mean stroke angle





Flight Envelope

Longitudinal Flight Envelope

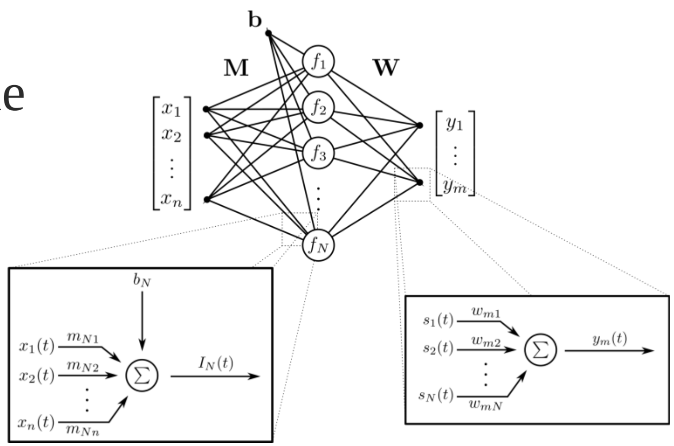
Peak-to-peak stroke amplitude and mean stroke angle as a function of speed and climb angle

Flapping Wing Flight Control

- Fixed-gain controllers require hand calibration for each robot [Ma, ’13], [Dickson, ’08]
- Adaptive controller for wind gust disturbance rejection only stabilizes about hovering [Chiraratananon, P. ’17]
- Hovering control of simplified 2D model with SNN [Clawson, T.S. ’16]

Accomplished Research Goal:

- Develop a full envelope flight controller, which can adapt online to physical parameter variations
- Spiking neural networks (SNNs) can adapt online and can be implemented in power-efficient neuromorphic chips



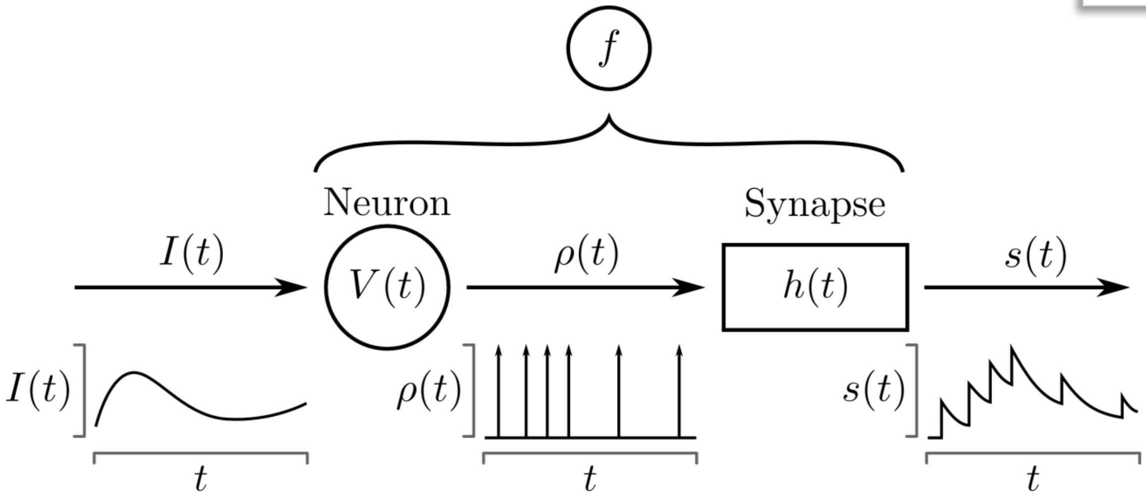
Event-based SNN Control Model

- 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 k[th] spike
M	Spike count
τ_s	Synaptic time constant

SNN Controller – Full Flight Envelope

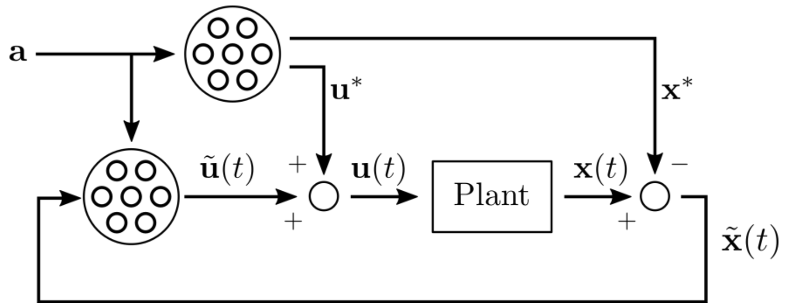
- SNN trained to approximate steady-state gain of gain-scheduled PIF
- PIF Gain matrices dependent on scheduling variables **a**

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

- 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 $\mathbf{K}_{ss}$

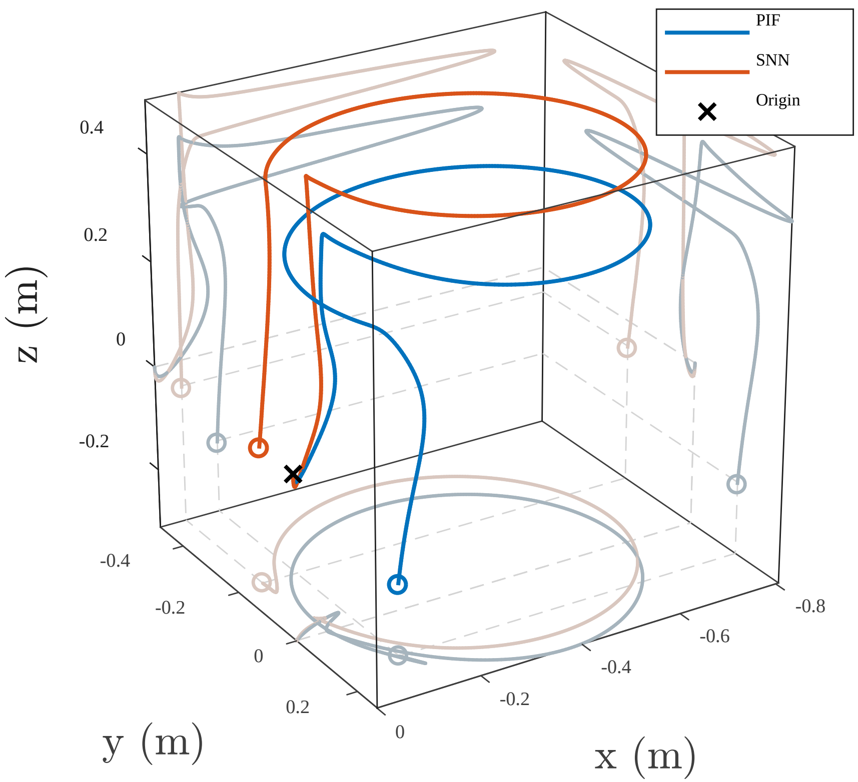
$$\mathbf{W} = \underset{\mathbf{V}}{\operatorname{argmin}} \sum_j \left\| \mathbf{K}_{ss}(\mathbf{a}) - \mathbf{V}F(\mathbf{M}\mathbf{a}_j + \mathbf{b}) \right\|^2$$

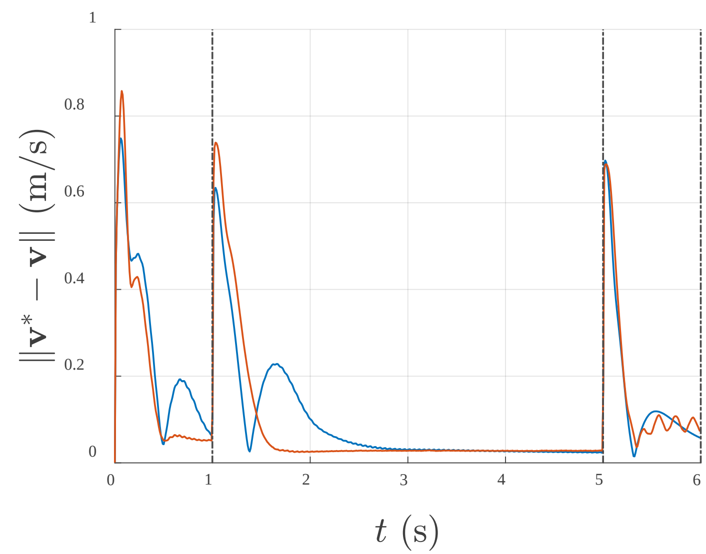
- SNN Control input is a linear transformation of post-synaptic current

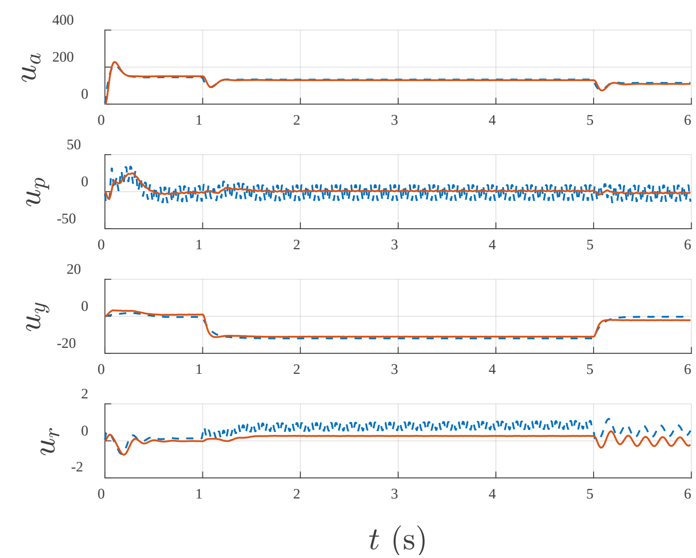
$$\tilde{\mathbf{u}}(t) = \mathbf{W}(\mathbf{a})\mathbf{s}(t)$$

$\mathbf{K}_i$	PIF gain matrices	$\tilde{\mathbf{x}}$	State deviation
$\tilde{\mathbf{u}}$	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 – Complete Turn







Neuromorphic Vision Sensors

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

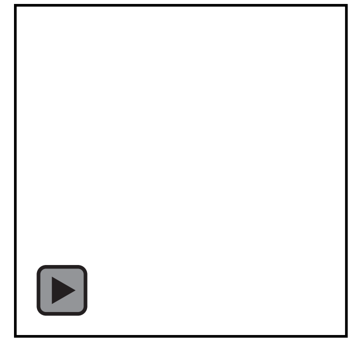
$$p_i = \begin{cases} 1, & \text{if } \ln(I(x,y,t_{i-1})) - \ln(I(x,y,t_i)) \geq -\theta \\ -1, & \text{if } \ln(I(x,y,t_{i-1})) - \ln(I(x,y,t_i)) \leq \theta \end{cases}$$

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

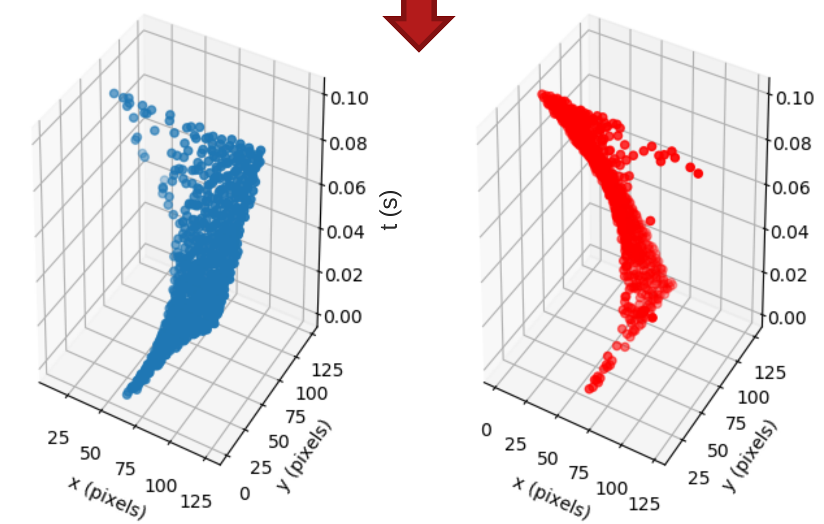
$$x, y \in \mathbb{N}^+ \qquad t \in \mathbb{R}^+ \qquad p \in \{-1, 1\}$$

- The set of all events is

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



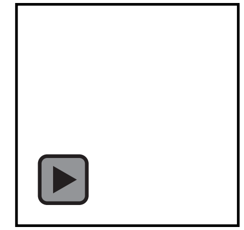
Source



"On" Events

"Off" Events

Neuromorphic Optical Flow

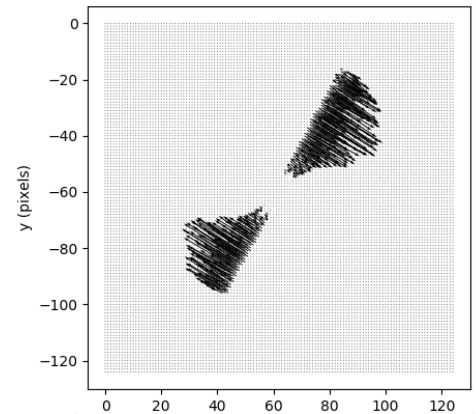


Standard Optical Flow Problem

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

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

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



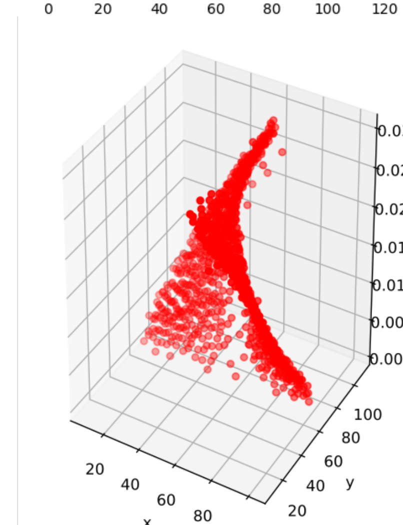
Neuromorphic Optical Flow

- 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 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 Results

1. Compute difference between predicted and measured intensity

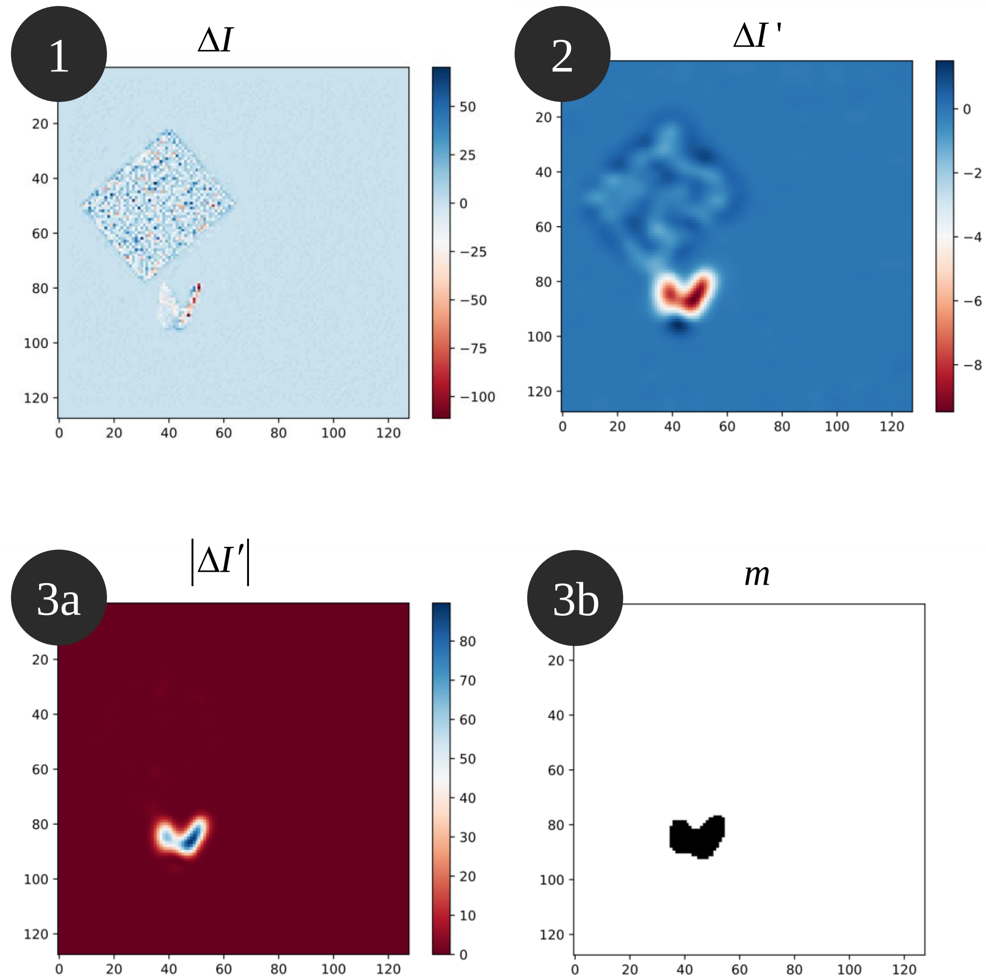
$$\Delta I(x,y,t) = I(x,y,t) - \tilde{I}(x,y,t)$$

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

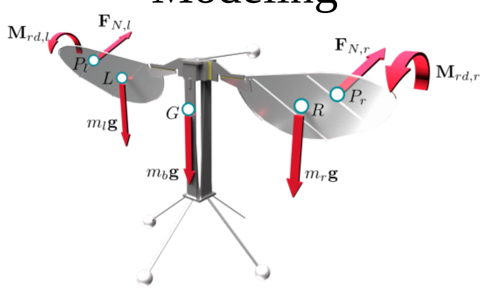
$$m(x,y,t) = \begin{cases} 1, & \text{if } |\Delta I'(x,y,t)| > \gamma. \\ 0, & \text{otherwise.} \end{cases}$$



Summary of Research Accomplishments

1

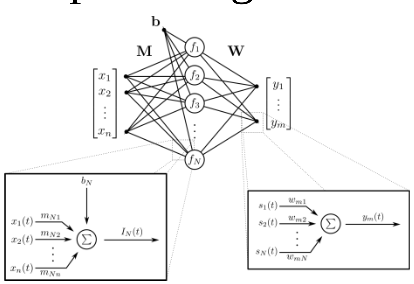
Modeling



- Flight model captures dominant modes
- Set points for steady maneuvers were computed
- Model predicts that forward flight becomes stable with increasing speed

2

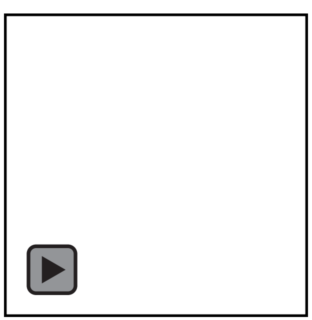
Adaptive Flight Control



- Adaptive SNN Controller can adapt to unmodeled parameter variations
- SNN can provide control for full flight envelope

3

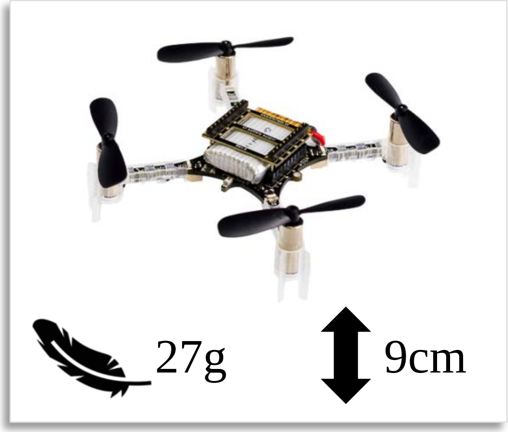
Exteroceptive Sensing



- Optical flow can be efficiently computed from neuromorphic cameras
- Target motion can be detected from a rotating neuromorphic camera

Future Work

- With the model:
  - Analyze stability at additional set flight set points to show bifurcations as a function of flight speed and other variables
- On the physical RoboBee:
  - Integrate SNN controller with the current hardware setup
  - Demonstrate basic maneuvers with the SNN Controller
- In Simulation:
  - Using neuromorphic cameras, track a moving target while avoiding obstacles in an unknown environment
- On a small quadcopter:
  - Attach regular camera and use frames to simulate neuromorphic camera in real time
  - Track a moving target while avoiding obstacles in an unknown environment



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

Funded by the ONR Grant # N00014-17-1-2614

Event-based Sensorimotor Planning and Control

PIs: Silvia Ferrari♣ and Robert Wood♠

Ph.D. Student: Taylor S. Clawson

Published 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.

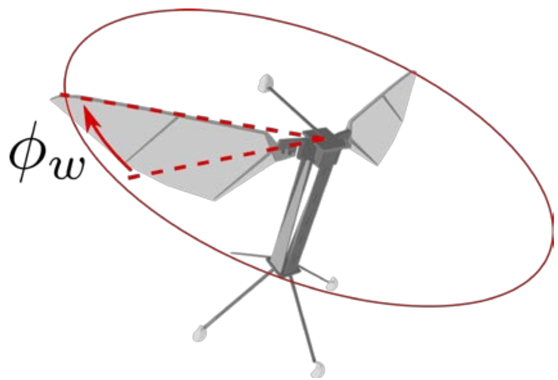
T. S. Clawson, S. B. Fuller, R. J. Wood, S. Ferrari “A Blade Element Approach to Modeling Aerodynamic Flight of an Insect-scale Robot,” *American Control Conference (ACC)*, Seattle, WA, May 2017.

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.

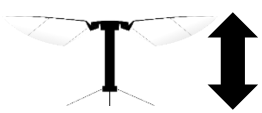
Back-up Slides

Background: RoboBee Actuators

- Wing stroke angle ϕ_w controlled independently for each wing
- Thrust and body torques controlled by modulating stroke angle commands

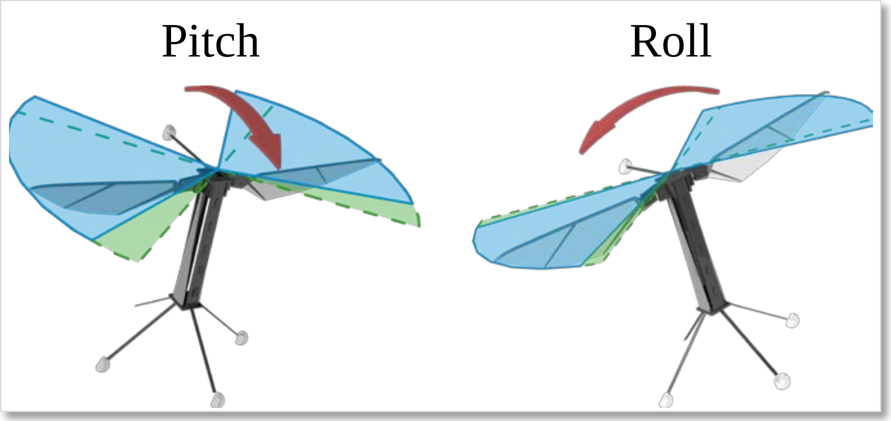


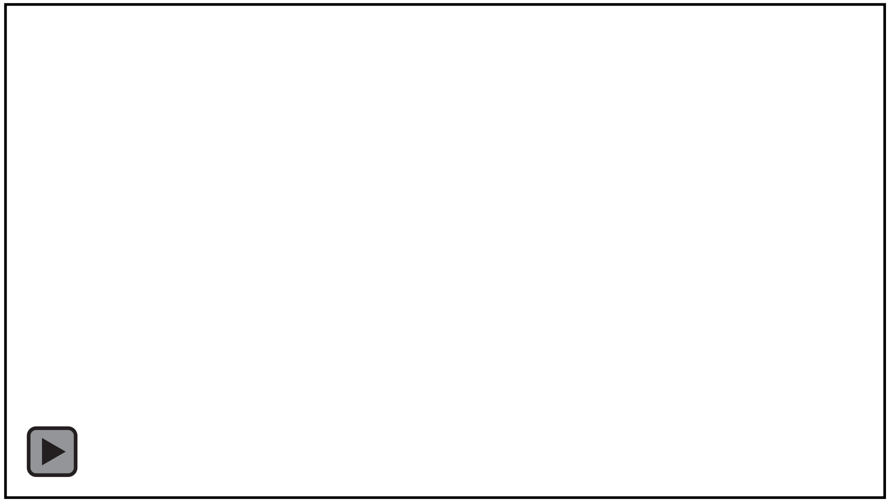
100 mg



14 mm

120 Hz

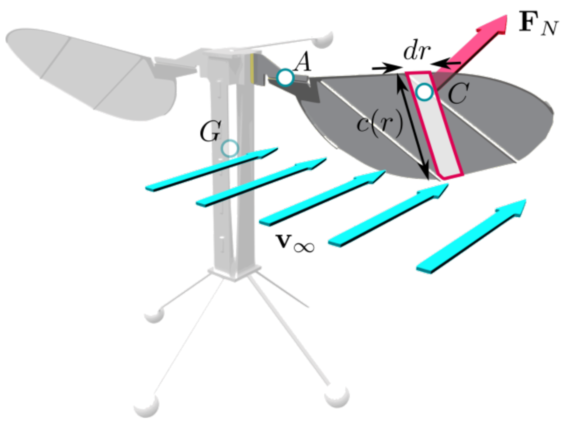




Video of RoboBee test flight courtesy of the Harvard Microrobotics Lab

Modeling Flapping Wing Flight

- Aerodynamic forces in flapping flight differ from classic airfoil models
- Modeling aerodynamic effects on flapping wings
  - Computationally expensive CFD models [Liu, ’98], [Sun, ’02]
  - Simplified models can accurately predict stroke-averaged forces [Whitney, ’10], [Dickinson, ’99], [Wang, ’04]
- Modeling flight dynamics of the insect or robot body
  - Simple 2D models [Ristroph ’13]
  - Stroke-averaged models [Chirarattananon, ’16]
  - Kinematically-constrained wing trajectories
    - Limited wing pitch [Oppenheimer, ’10]
    - Kinematic models from experimental data [Wang, ’16], [Dickson, ’08]
- Finding hovering set point and analyzing modes of motion and stability [Wu, ’12]



Solving for Maneuver Set Points

- To find set points corresponding to steady maneuvers, solve equations of motion subject to maneuver constraints $\mathbf{c}_m = \mathbf{0}$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \qquad \mathbf{c}_m \triangleq \mathbf{x}^*(T) - \mathbf{x}(T)$$

- Discretize ODE and write dynamics as constraints using Hermite-Simpson rule

$$\mathbf{0} = \bar{\mathbf{x}}_{k+1} - \frac{1}{2}(\mathbf{x}_{k+1} + \mathbf{x}_k) - \frac{\Delta t}{8}(\mathbf{f}_k - \mathbf{f}_{k+1}) \qquad \mathbf{0} = \mathbf{x}_{k+1} - \mathbf{x}_k - \frac{\Delta t}{6}(\mathbf{f}_{k+1} + 4\bar{\mathbf{f}}_{k+1} + \mathbf{f}_k)$$

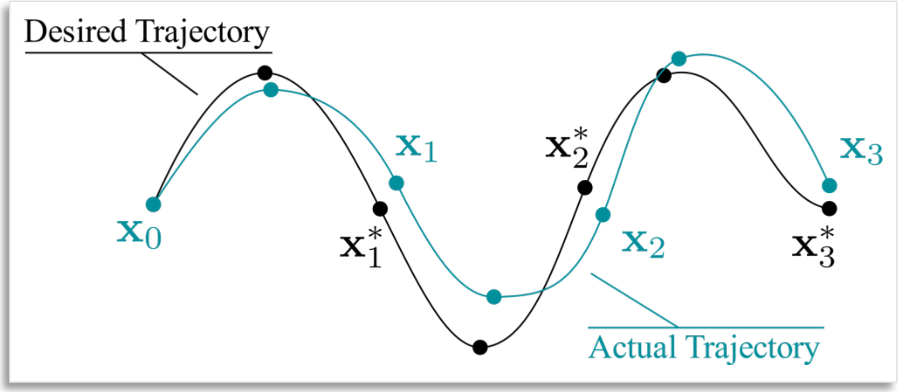
- Dynamics constraints can be written in terms of constant matrices **A**, **B**:

$$\mathbf{c}_d(\mathbf{x}, \mathbf{u}) \triangleq \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{q}(\mathbf{x}, \mathbf{u})$$

$$\mathbf{q}(\mathbf{x}, \mathbf{u}) \triangleq \Delta t \begin{bmatrix} \mathbf{f}_1 & \bar{\mathbf{f}}_2 & \mathbf{f}_2 & \bar{\mathbf{f}}_3 & \dots & \mathbf{f}_M \end{bmatrix}^T$$

- Use nonlinear program to numerically solve:

$$\begin{bmatrix} \mathbf{c}_m(\mathbf{x}) \\ \mathbf{c}_d(\mathbf{x}, \mathbf{u}) \end{bmatrix} = \mathbf{0}$$



Single-layer SNN Controller

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

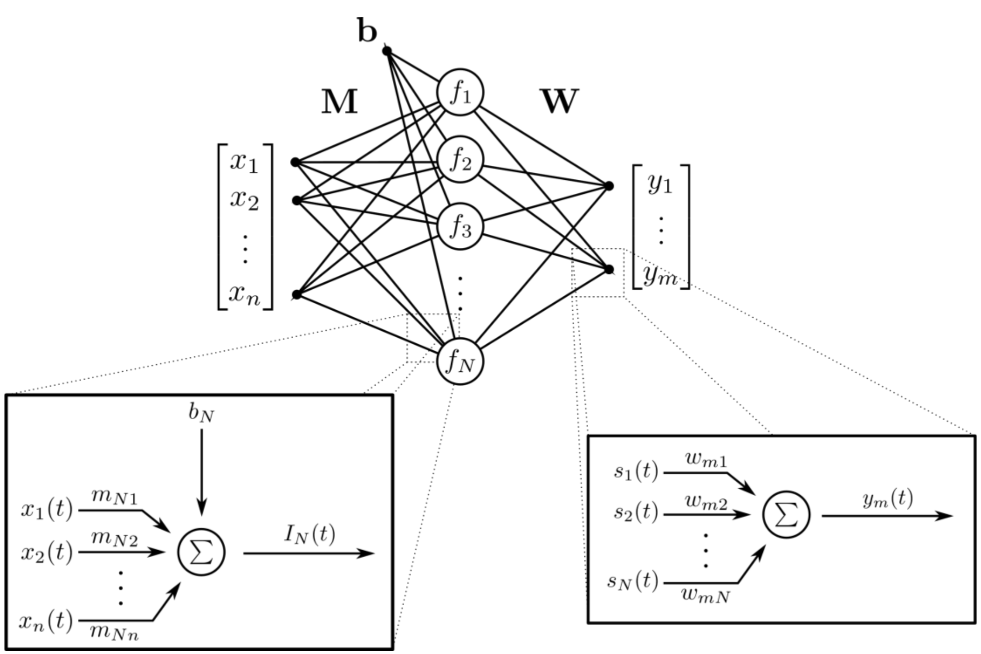
$$\mathbf{y}(t) = \mathbf{W}\mathbf{s}(t) = \mathbf{W}F(\mathbf{M}\mathbf{x}(t) + \mathbf{b})$$

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

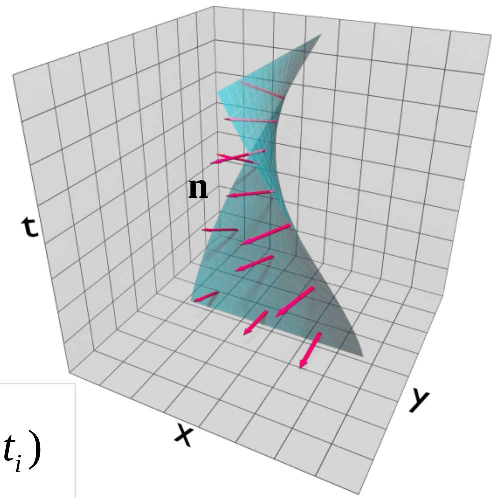
Exteroceptive Sensing Motivation

- 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)

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

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

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

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