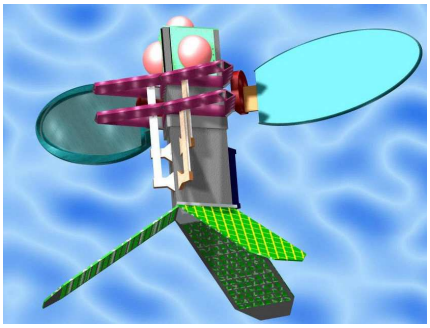
# MFI's Flight Control: a Hierarchical and Biologically Inspired Design



*Luca Schenato, Xinyan Deng, Wei Chung Wu*
*Shankar Sastry*
Department of EECS
University of California, Berkeley

# MFI Project: Overview

- **Objective**: Development of a micromechanical flying insect (MFI), a 10-25mm (wingtip-to-wingtip) device eventually capable of sustained autonomous flight
- **Length**: 5 years (kick off August 1998)
- **Investigators**: Prof. R. Fearing, EE
  Prof. M. Dickinson, Integrative Biology
  Prof. K. Pister, EE
  Prof. T. Sands, Material Science
  Prof. A. Majumdar , ME
  Prof. D. Liepmann , ME

## Project goals:

- Study of aerodynamics in microscale.
- Study of robotics in microscale.
- Integration on a single chip of sensors, communication devices and power supply.
- Study of real insect flight.
- New control techniques for complex motor tasks.
- Scouting and surveillance

## MFI : main problems

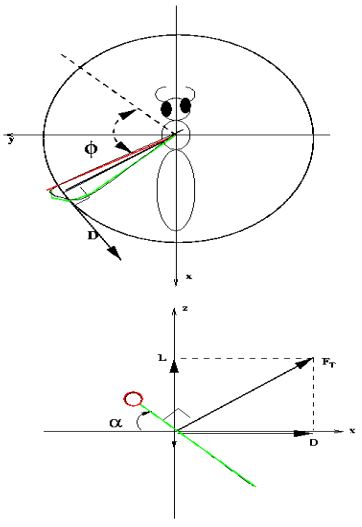
- Small dimension, but large displacements.
- Fast movements
- Small power dissipation.
- Lack of a quantitative model for the aerodynamics of insect flight.
- Partial knowledge about flight control in real flying insects.
- Large perturbations on the MFI's body dynamics.

# Talk summary:

- **Real insect flight analysis**:
  - minimum lift/power requirements
  - body's attitude control mechanisms
  - neuromotor system
- **Virtual Insect Flight Simulator (VIFS)**
- **Control unit:**
  - Hierarchical architecture of controllers
  - Switching control for hovering stabilization
  - Reinforcement learning based controllers

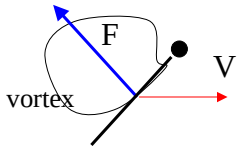
# Flight analysis:

- The mean aerodynamic forces over a whole wingbeat are sufficient to **sustain** the insect weight.
- **Fundamental** kinematics parameters:
  - wingbeat frequency, ***n***
  - stroke amplitude, $\Phi$
  - mean angle of attack, $\overline{\alpha}$

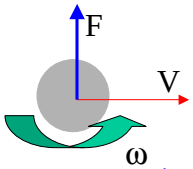


# Insect Aerodynamic :

- Aerodynamic machanisms:
  - Delayed stall:

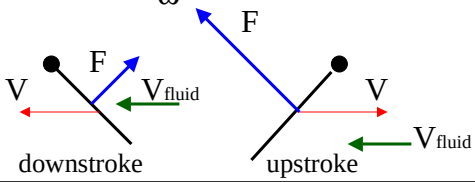



  - Magnus effect




  - Wake capture

$V_{rel}=V_{wing}+V_{fluid}$

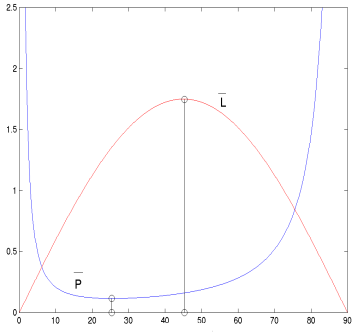



# Aerodynamic model:

- **Mean lift:**

$$\bar{L}=\frac{1}{4}R^3\bar{c}\hat{r}_2^2(S)\rho u_{m.s.}^2 C_N n^2\Phi^2 sin(2\alpha)$$

(aerodyn.; morphology; kinematics)




ANGLE OF ATTACK

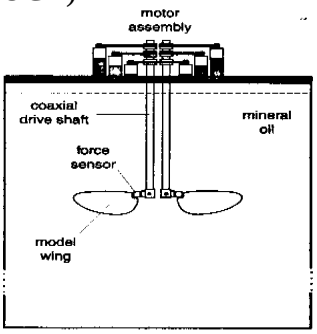
- **Mean aerodynamic power:**

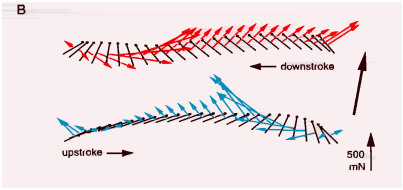
$$\bar{P}=\sqrt{\frac{\bar{L}^3}{\rho R\bar{c}}}\left(\frac{u_{m.c.}}{u_{m.s.}}\right)^3\left(\frac{r_3(S)}{r_2(S)}\right)^3\frac{C_D(\alpha)}{[C_L(\alpha)]^{\frac{3}{2}}}$$

# Experimental instrumentation:

M.H. Dickinson (UCB)

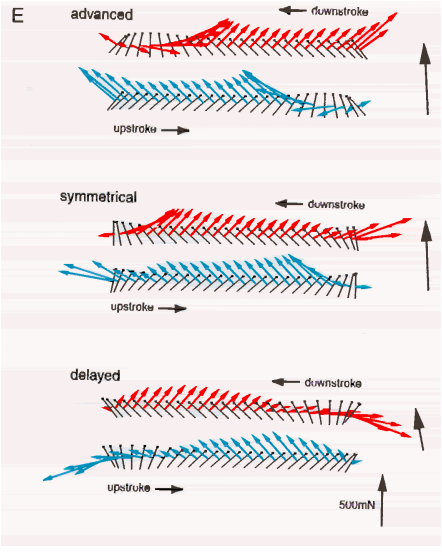
- **Apparatus:** dynamically scaled model of fruit-fly:
  - 25cm wings provided of force sensors
  - mineral oil tank
  - 1-5 N aerodynamic forces
  - 2-3 seconds per wingbeat
- **Objective**: can measure instantaneous forces, impossible on real flying insects.



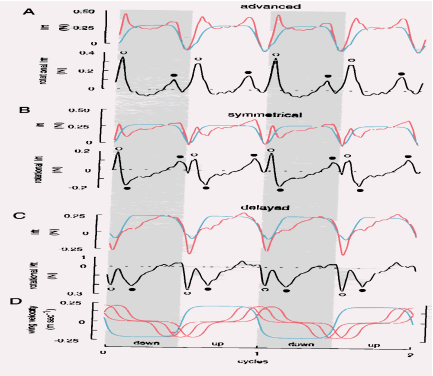



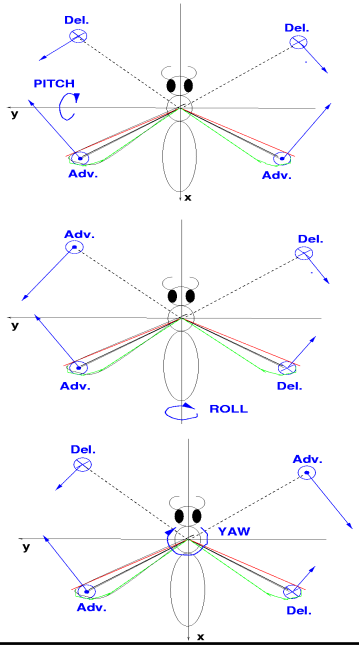

# Distribution of instantaneous forces along a wingbeat cycle:

- Important to stabilise the flight.
- **<u>Fundamental</u>** kinematic parameters**:**
  - phase of rotation, $t_{rot.}$
  - mean stroke, $\overline{\phi}$
  - angle of attack, $\alpha$
- Large and impulsive forces at the wing reversals.



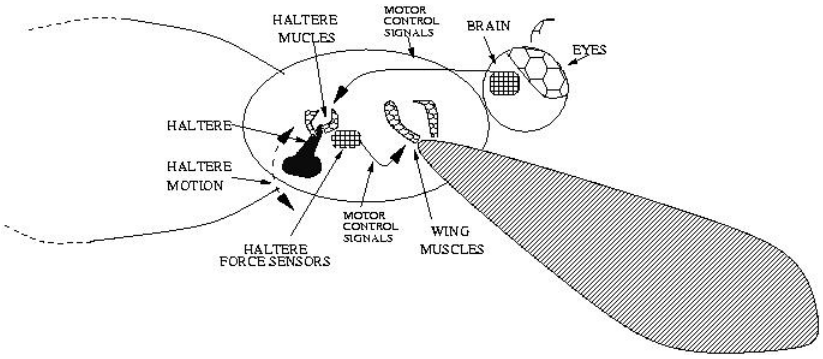
# Flight stabilization mechanisms



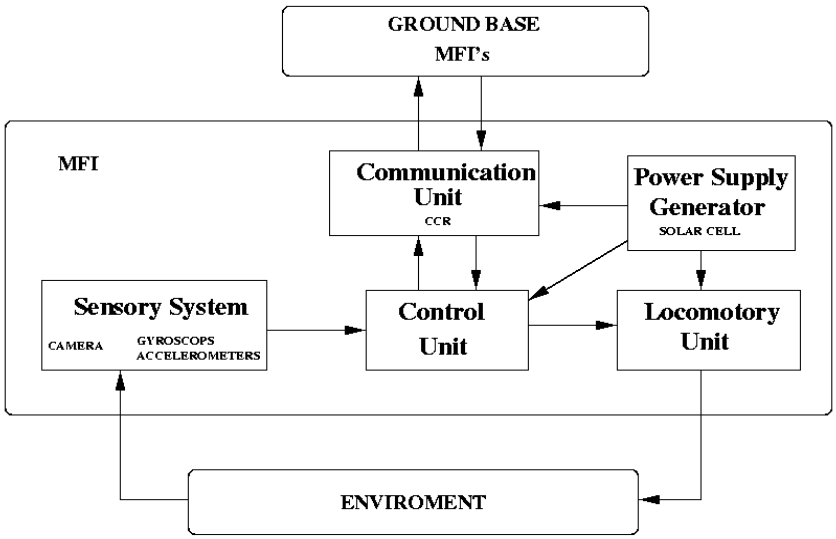

- It is possible to obtain torques if the wing kinematics is asymmetric.

# Neuromotor Control of Insect Flight

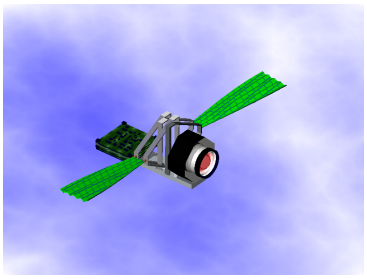


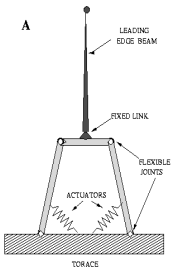

- Halteres are the insect **inertial** navigation system
- Halteres **passively** control **wing** kinematics
- Visual system **voluntarily** controls **halteres** kinematics, thus **indirectly** modifying wing kinematics

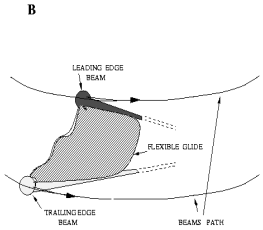
# MFI structure:




# Locomotory Unit:



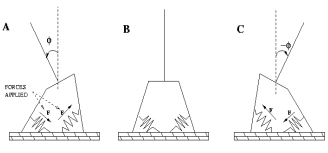


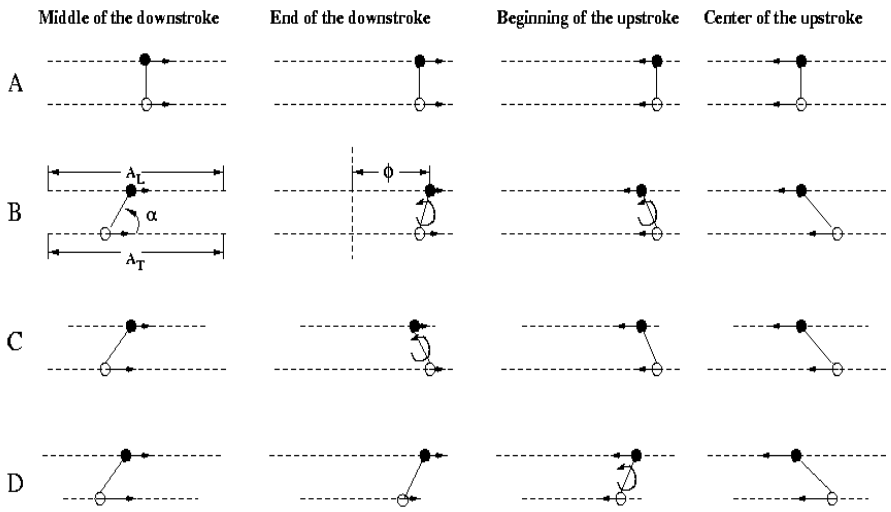

## Mathematical model:

$$I_w\ddot{\phi} + \gamma\dot{\phi}\left|\dot{\phi}\right| + \kappa\phi = T\sin(\omega t)$$

$$\gamma = -\frac{1}{2}\rho C_D(\alpha)\hat{r}_3^3(S)R^4\overline{c}$$



# Control of Wing Kinematics




$$x_L = A_L \sin(2\pi nt + \delta) + A_{DC} \qquad x_T = A_T \sin(2\pi nt) + B_{DC}$$

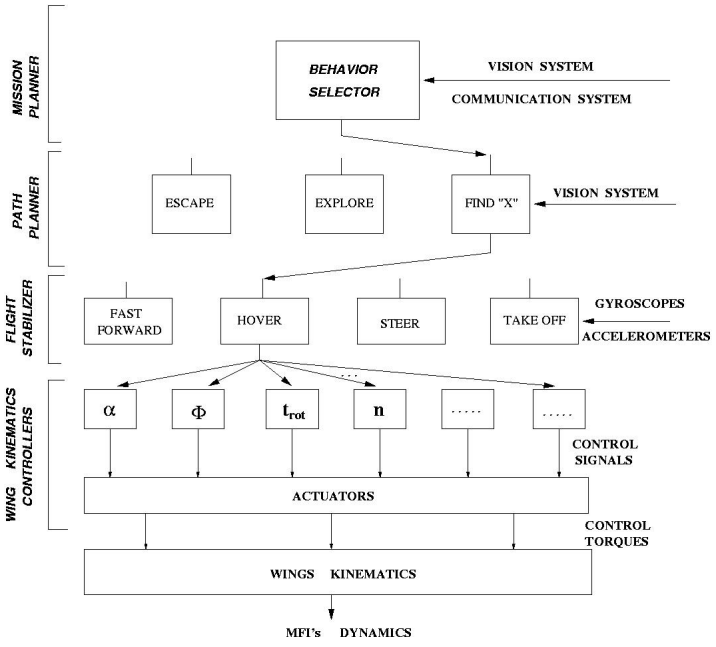
# Flight control issues:

- Lack of **precise** mathematical model for the aerodynamics .
- Limited knowledge about the neuromotor control system in real flying insects.
- Sudden and strong external perturbations on the MFI's body dynamics.
- Uncertainty on the exact dimensions of the electromechanical devices.

# Proposed Solution:

- **Hierarchical structure for the control unit:**
  - reduces the original complex problem to the coordination among simpler control systems.
  - allows the possibility to increase the complexity and the variety of possible tasks in the future.
- **Controllers based on "reinforcement learning":**
  - do not require the knowledge of a mathematical model for the aerodynamics.
  - robust to external perturbations.
  - robust to uncertainty about exact electromechanical parameters

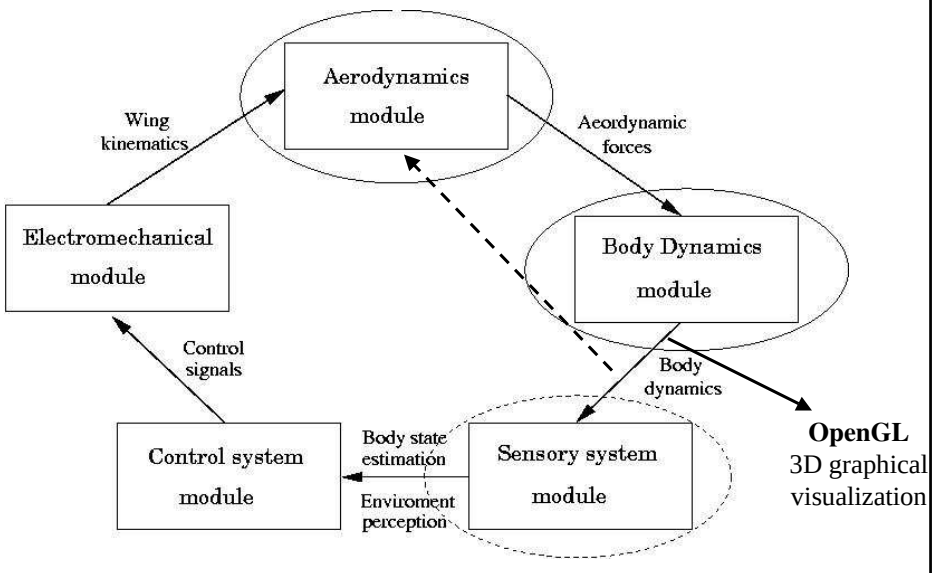
# Control Unit: Architecture

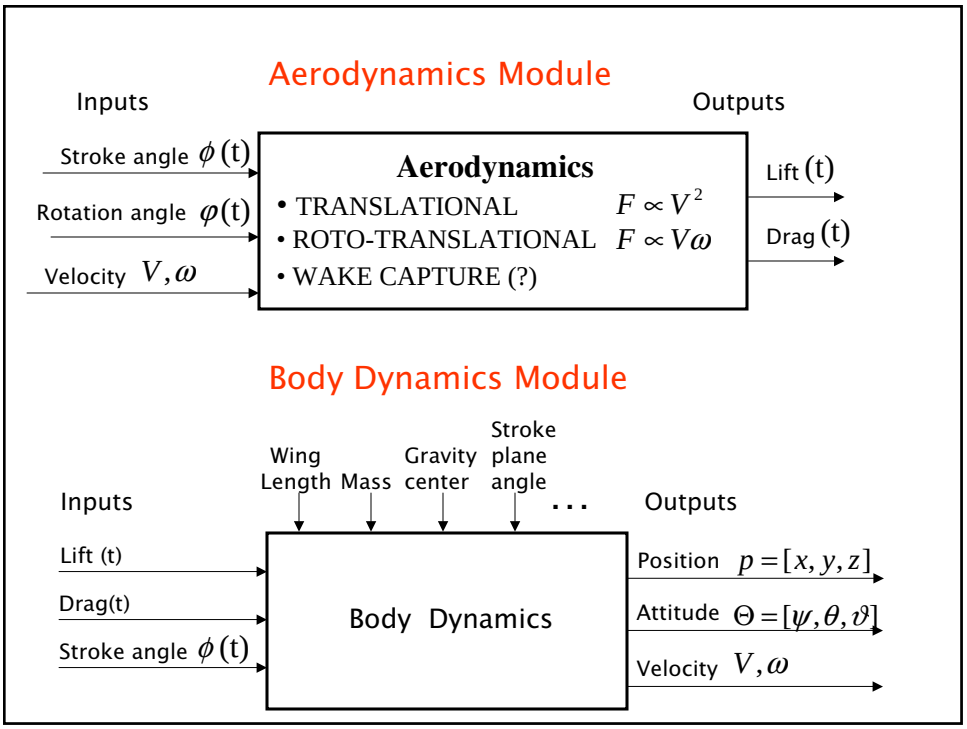


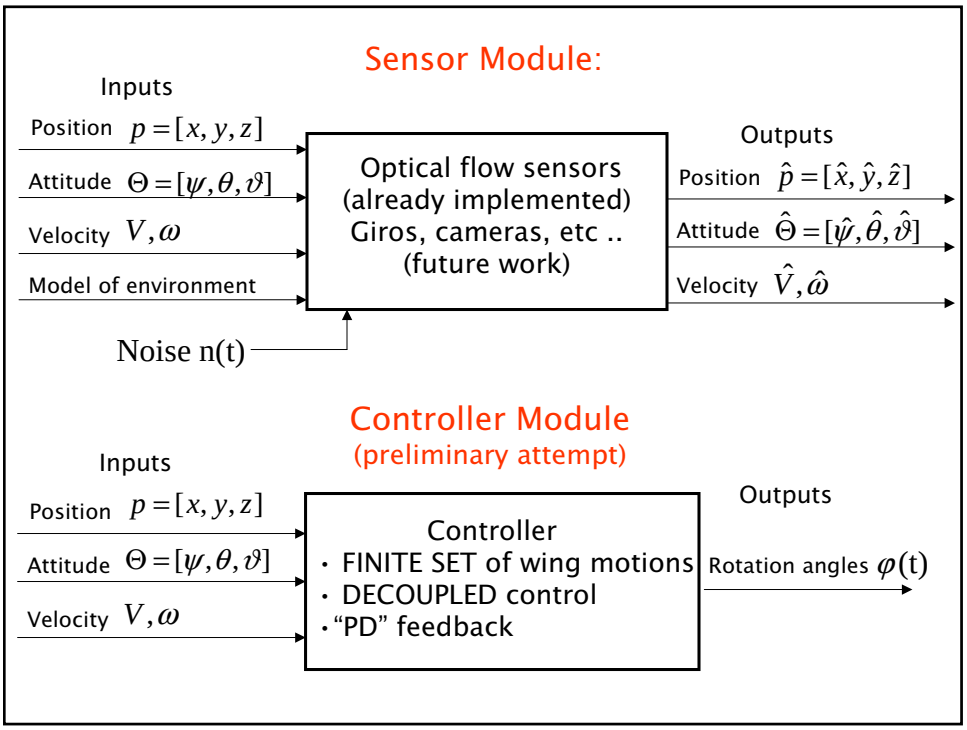
# Virtual Insect Flight Simulator (VIFS)

- **Fast simulation** of insect flight on any workstation
- Allows **evaluation** of steering maneuvers and/or **control algorithms**
- Appropriate for **visualization** purposes

## Simulator architecture

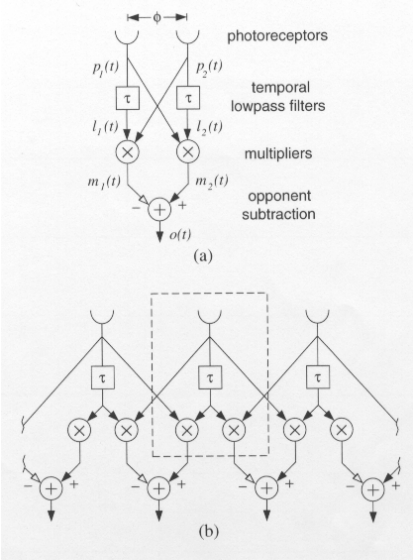



Aerodynamics Module
Inputs
Outputs
Stroke angle $\phi$(t)
Rotation angle $\varphi$(t)
Velocity $V, \omega$
Aerodynamics
• TRANSLATIONAL $F \propto V^2$
• ROTO-TRANSLATIONAL $F \propto V\omega$
• WAKE CAPTURE (?)
Lift (t)
Drag (t)
Body Dynamics Module
Wing Length
Mass
Gravity center
Stroke plane angle
...
Inputs
Outputs
Lift (t)
Drag(t)
Stroke angle $\phi$(t)
Body Dynamics
Position $p = [x, y, z]$
Attitude $\Theta = [\psi, \theta, \vartheta]$
Velocity $V, \omega$


Sensor Module:
Inputs
Position $p = [x, y, z]$
Attitude $\Theta = [\psi, \theta, \vartheta]$
Velocity $V, \omega$
Model of environment
Noise n(t)
Optical flow sensors (already implemented) Giros, cameras, etc .. (future work)
Outputs
Position $\hat{p} = [\hat{x}, \hat{y}, \hat{z}]$
Attitude $\hat{\Theta} = [\hat{\psi}, \hat{\theta}, \hat{\vartheta}]$
Velocity $\hat{V}, \hat{\omega}$
Controller Module
(preliminary attempt)
Inputs
Position $p = [x, y, z]$
Attitude $\Theta = [\psi, \theta, \vartheta]$
Velocity $V, \omega$
Controller
• FINITE SET of wing motions
• DECOUPLED control
• "PD" feedback
Outputs
Rotation angles $\varphi$(t)

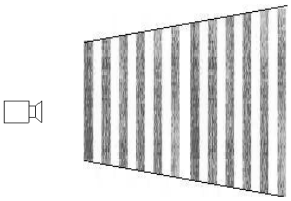
## OPTICAL FLOW SENSOR MODELING:

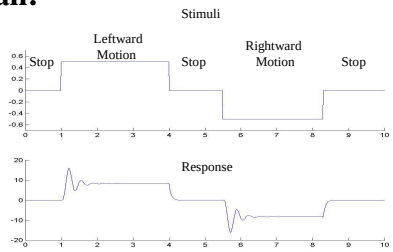
- Based on **analog spatial-temporal correlation** algorithm (Reichardt)
- Models a sensor developed at Caltech (R. Harrison, C. Koch)
- **20 photoreceptors** with 35° field of view
- **parameters** for filters extracted from Harrison's Ph.D. Thesis 2000
- **1mW** power consumption



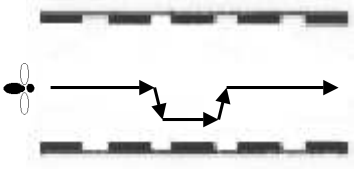

## SENSOR SIMULATION:

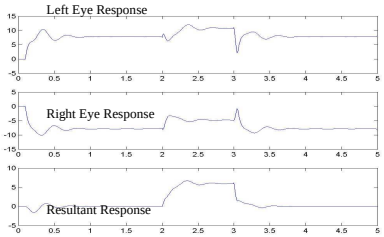
### Striped wall:






### Striped tunnel (2 sensors):





## AERODYNAMICS MODULE SIMULATIONS:

Stroke angle

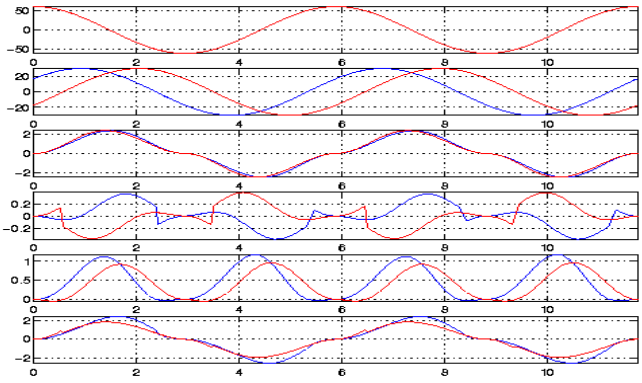
Rotation angle

Pressure force: Translational

Pressure force: Roto-translational

Lift

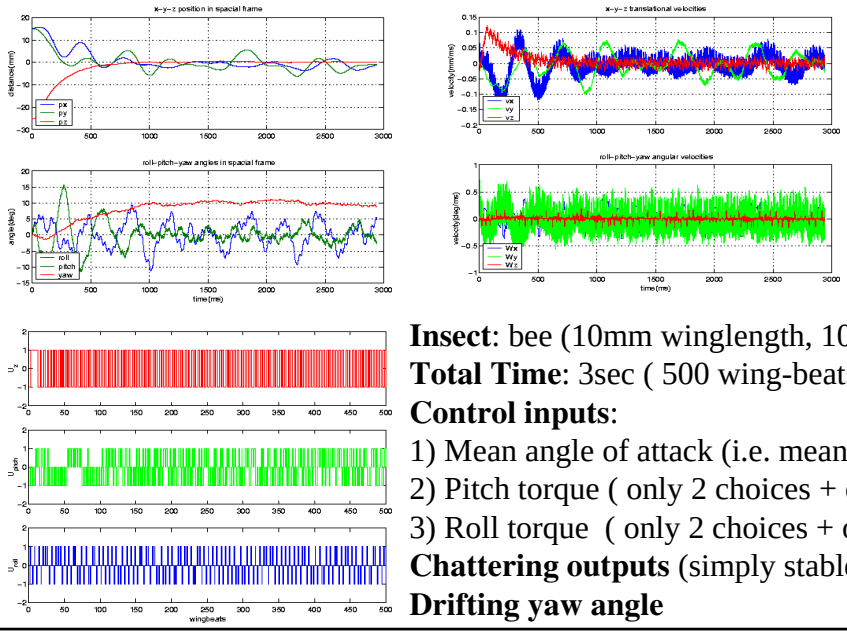
Wing thrust (drag)



Wrench map

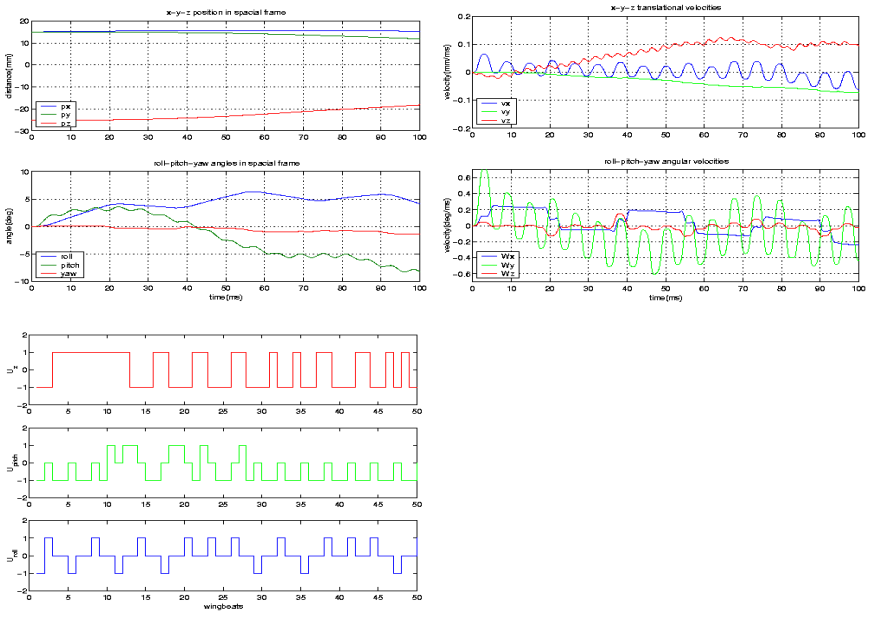
| left (degs) | right (degs) | Fx (mN) | Fy ($m_N$) | Fz (mN) | $\tau$ roll | $\tau$ pitch | $\tau$ yaw |
|---|---|---|---|---|---|---|---|
| AA 45 | DD 45 | -0.0262 | -0.1863 | 1.0486 | 0.0703 | -0.1136 | -0.055 |
| DD 45 | AA 45 | -0.0262 | 0.1863 | 1.0486 | -0.0703 | -0.1136 | 0.055 |
| AD 45 | DA 45 | -0.0215 | -0.0015 | 1.2072 | 0.0208 | -0.0939 | -0.7202 |
| DA 45 | AD 45 | -0.0215 | 0.0015 | 1.2072 | -0.0208 | -0.0939 | 0.7202 |
| AD 45 | AD 45 | -0.1384 | 0 | 1.2049 | 0 | -0.7186 | 0 |
| DA 45 | DA 45 | -0.0965 | 0 | 1.2117 | 0 | 0.5298 | 0 |
| SS 45 | SS 45 | -0.0226 | 0 | 1.2324 | 0 | -0.1009 | 0 |
| SS 60 | SS 60 | -0.0479 | 0 | 0.7939 | 0 | -0.195 | 0 |
| SS 45 | SS 60 | -0.035 | -0.0403 | 1.0135 | -0.7452 | -0.147 | 0.0652 |
| SS 60 | SS 45 | -0.035 | 0.0403 | 1.0135 | 0.7452 | -0.147 | 0.0652 |

## CONTROLLER SIMULATION: HOVERING




**Insect**: bee (10mm winglength, 100mg)
**Total Time**: 3sec ( 500 wing-beats)
**Control inputs**:
1) Mean angle of attack (i.e. mean lift)
2) Pitch torque ( only 2 choices + or - )
3) Roll torque ( only 2 choices + or - )
**Chattering outputs** (simply stable)
**Drifting yaw angle**

## CONTROLLER SIMULATION: ZOOM




# Aerodynamic model: possible improvements

- Find a **map** for *mean* forces and torque:
  the wing kinematic parameters are updated only every one (or few) wingstrokes

$$\overline{L}, \overline{D}, \overline{\tau}_{roll}, \overline{\tau}_{pitch}, \overline{\tau}_{yaw}$$

- Find a **map** from kinematics to *instantaneous* aerodynamic forces using Robofly data:

$$L(t), D(t), \tau_{roll}(t), \tau_{pitch}(t), \tau_{yaw}(t)$$

# Reinforcement learning:

- Learning algorithms for problems in a Markov Decision Process (MDP) framework:

  state $\mathbf{s} \in S = [p, R, u, \omega; \mathbf{v}_{t-1}]$

  action $\mathbf{v} \in V(\mathbf{s}) = [n, \Phi, \bar{\phi}, \alpha, t_{rot}]$

  transition probability $p(s_i, v, s_j) = p_{i,j}(v)$

  transition cost(reward) $c(s_i, v, s_j) = c_{i,j}(v)$

- These algorithms **learn** the *minimum expected **total** cost-to-go* or *value* for each state, $J(\mathbf{s})$, and this can determine the best *policy* (choice of action in each state, $v^*(s)$ )

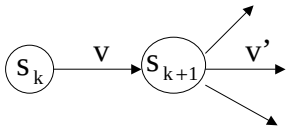
# Q-learning

$Q_t(s_k, v) \equiv$ The expected cost-to-go at time instant $t$ when action $\mathbf{v}$ is chosen in the state $s_k$ , and optimal actions are chosen thereafter.

**Optimal Q function**, $Q^*(s_k, v)$, is a fixed point for the Bellman equation: $$Q^*(s_k, v) = E\left[c_{k,k+1}(v) + \min_{v'} Q^*(s_{k+1}, v')\right]$$

**Optimal action:**

$$v^*(s_k) = \arg\min_{v'} Q^*(s_k, v')$$




**Update:**

$$Q_{t+1}(s_k, v) = (1-\alpha) Q_t(s_k, v) + \alpha\left[c_{k,k+1}(v) + \gamma \min_{v'} Q_t(s_{k+1}, v')\right]$$

$$Q_t(s_k, v) \xrightarrow[t\to\infty]{} Q^*(s_k, v)$$

# Implementation of flight mode: HOVER

**Transition cost** :

$$c_{i,j}(s) = \int_t^{t+1} e^{-\lambda(\tau - t)} \left( \|p(\tau) - p_d\| + \|R(\tau) - R_d\| + \|u(\tau)\| + \|\omega(\tau)\| \right) d\tau$$

**Value function:**

$$Q(s,v) = E\left[ \int_t^{\infty} e^{-\lambda\tau} \left( \|p(\tau) - p_d\| + \|R(\tau) - R_d\| + \|u(\tau)\| + \|\omega(\tau)\| \right) d\tau \right]$$

- Different basic maneuvers can be implemented analogously

# Problems:

- Large **state space**: 12 + 2x2 = 16 dimensions:

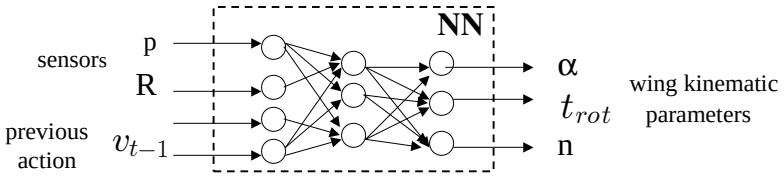
intractable in practice
example: uniform discretization: 4 cells per dimension requires approximately $4^{16}$ 4 billions cells !

- Large **action space**: 7x2 = 14 parameters to choose

even harder to deal with then the previous one

# Proposed solutions: (1)

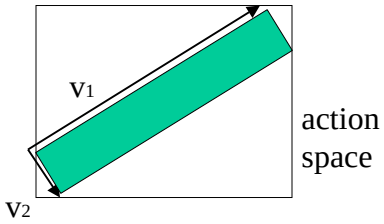
- State Space:
  - *Adaptive discretization* and *function approximators:* cells have small size only where necessary
  - *Direct policy search:* use function approximator (i.e. small number of unknowns) to parameterize policies. Search/optimize function approximator parameters to find a good policy.




# Proposed solutions: (2)

**Action Space:**

- Exploit *prior* knowledge about flight mechanisms of real flying insects:
  - coupling of actions that give rise to the same aerodynamic effect



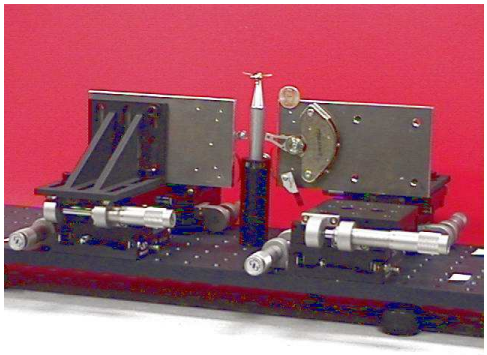

- *Initialize* Value Function or NN using "*learning by watching*": faster convergence

# Conclusions:

- Identified the biokinematic parameters necessary to sustain insect weight and to stabilize body's attitude
- Hierarchical architecture simplifies controller design
- Reinforcement learning controllers do not require knowledge of aerodynamic model
- Implementation of Virtual Insect Flight Simulator
- First attempt of hovering control

# Future work:

- **Improve** VIFS ( aerodynamics, sensors, …)
- Test **nonlinear** and **robust controllers**
- Evaluate **reinforcement learning**-based controllers
- Implementation of a full set of **flight mode** controllers such as 'steer' and 'forward'.
- Implementation of the controllers for the **path planning**.

# Mock-up of the MFI

http://robotics.eecs.berkeley.edu/~ronf/mfi.html