Mining Hybrid Models from Data

Claire J. Tomlin
Hamsa Balakrishnan, Inseok Hwang

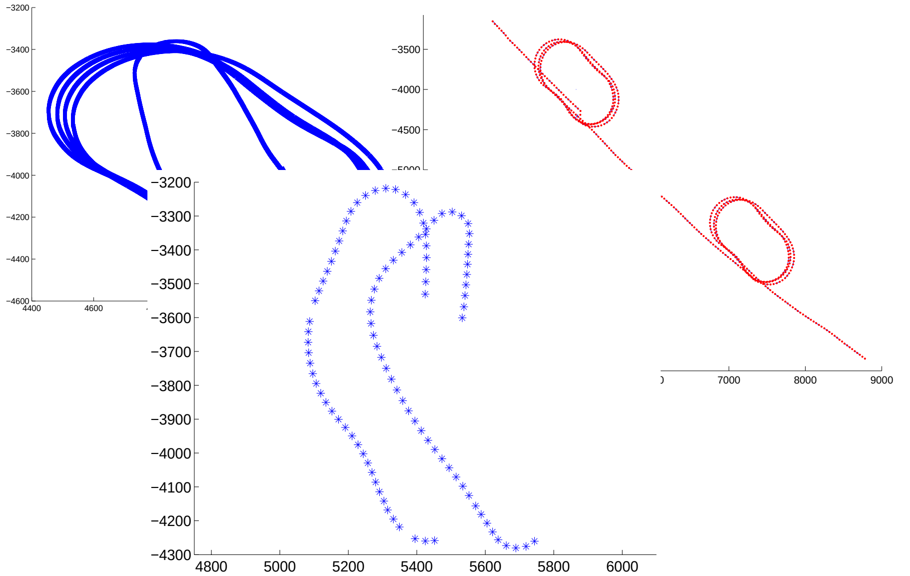
Department of Aeronautics and Astronautics
Department of Electrical Engineering
Stanford University

May 2005

Background and Outline

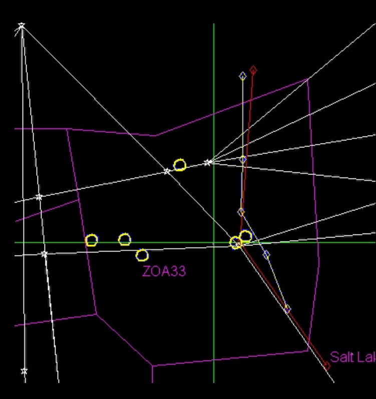
- Tools for the analysis and control of hybrid systems
  - Reachable set calculations
  - Approximation algorithms for trajectory optimization in hybrid systems
- This talk: identifying hybrid models from data

ETMS/UAV flight data



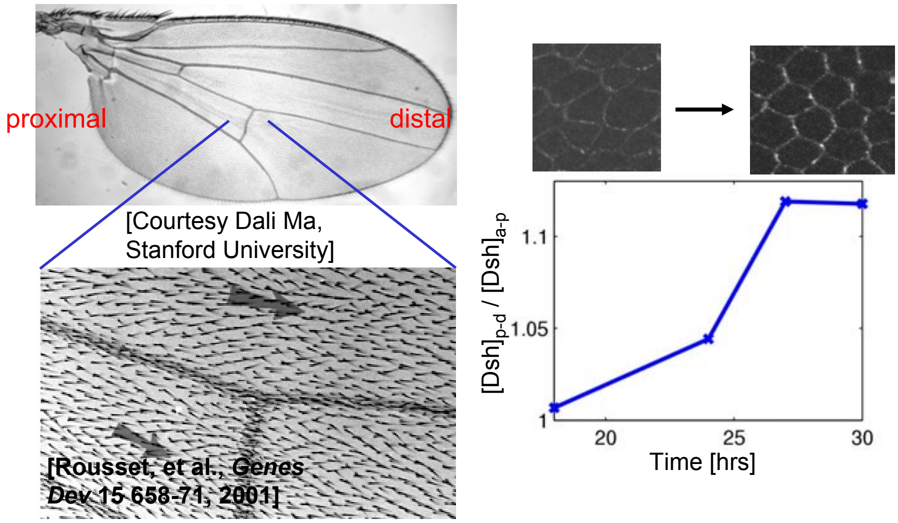
[UAV data with Teo and Jang; ETMS data from McNally (Ames)]

ETMS data with synthesized flight



[with Alex Bayen, Pascal Grieder]

Drosophila wing epithelium: Dsh protein



[Amonlirdviman, Khare, Tree, Chen, Axelrod, Tomlin **Science** ’05]

Hybrid System Model

$H = \{S, Init, In, f, Dom, R\}$

where

$S = Q \cup \Re^n$
state space

$Init \subseteq S$
initial states

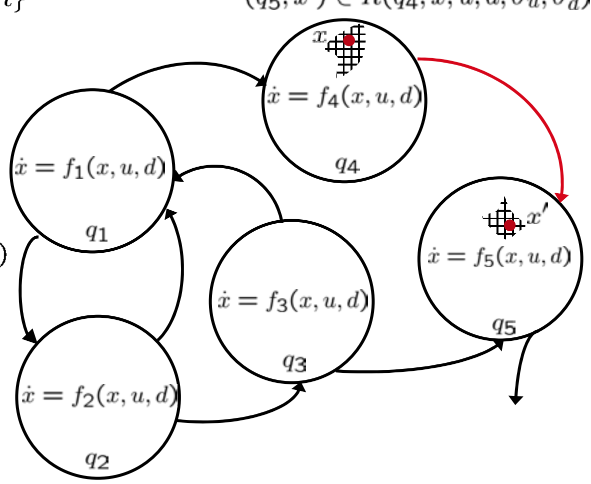
$In = (U \cup D) \cup (\Sigma_u \cup \Sigma_d)$
inputs

$f : S \times In \rightarrow S$
vector field

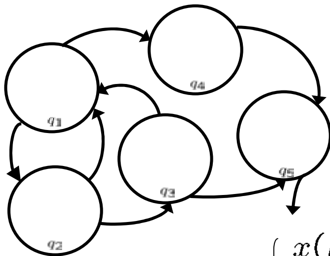
$Dom \subseteq S$
domain

$R : S \times In \rightarrow 2^S$
transitions

$(q_5, x') \in R(q_4, x, u, d, \sigma_u, \sigma_d)$



Stochastic Linear Hybrid System



Mode q_i
$$\begin{cases} x(k+1) &= A_i x(k) + w_i(k) \\ y(k) &= C x(k) + v_i(k) \\ w(k) &\sim \mathcal{N}(0, Q_i) \\ v(k) &\sim \mathcal{N}(0, R_i) \end{cases}$$

Continuous state parameters: $\theta_i = \{A_i, Q_i, R_i\} \Rightarrow \Theta$

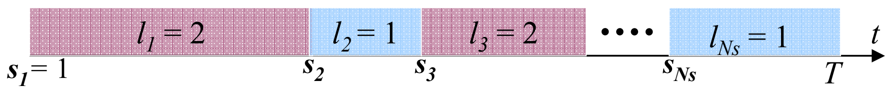
Transition matrix: $M_{ij} = \text{prob}(l_k = j | l_{k-1} = i)$

Assumptions on system behavior

We **assume** that

- ❑ Measurement matrices *C* are known
- ❑ System has a minimum (known) dwell time, T_d in each mode
- ❑ Typical system behavior is manifested in the available data sets
- ❑ Mode transitions are independent of the continuous state
- ❑ Mode transitions are probabilistic and Markovian

Maximum Likelihood Hybrid Model



- Data sequence $Y_{1:T} = [y(1), \ldots, y(T)]$
- N discrete modes
- N_S segments

Switching points	S	$=$	$\{s_1, s_2, \cdots s_{N_S}\}$
Labels (modes)	L	$=$	$\{l_1, l_2, \cdots l_{N_S}\}$

Continuous model Θ; Discrete model $D = \{S, L, M\}$

Maximum Likelihood Model

Given the continuous output of the system $Y_{1:T}$ we would like to compute the maximum likelihood hybrid system model.

$$\{\overline{D}, \overline{\Theta}\} = \arg \max_{\{D,\Theta\}} \Lambda(Y_{1:T} | D, \Theta)$$

where Λ is the likelihood function we would like to maximize

Parameter Inference Algorithm

Assume an initial continuous model ($\Theta^{(0)}$) and an initial discrete model ($D^{(0)}$).

iterate

Step 1: Find the **globally optimal segmentation** points (*S*) and their labels (*L*) assuming the model parameters of the current iteration (*k*). Update switching matrix, M. This gives us the maximum likelihood model $D^{(k+1)}$.

Step 2: Fit **maximum likelihood models** into the segmented time sequences, *i.e.*, for the computed {*S*,*L*}, fit the best $\Theta^{(k+1)}$.

until convergence to a local maximum

Likelihood function

$$\log \mathcal{L}(Y_{1:t}|\theta_i) = -\frac{1}{2}\sum_{k=1}^{t} \log \det \Sigma_{ki} - \frac{1}{2}\sum_{k=1}^{t} (r_i(k))'\Sigma_{ki}^{-1} r_i(k)$$

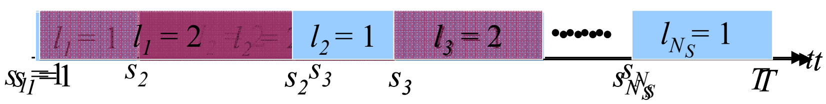
where $r_i(k) = y(k) - C_i x(k|k-1)$,

$$\Sigma_{ki} = C_i P(k|k-1) C_i' + R_i.$$

- Reflects how well model tracks the continuous state
- Easy to compute (Kalman filter recursions)

Optimal Segmentation

- Finding the optimal segmentation is potentially intractable

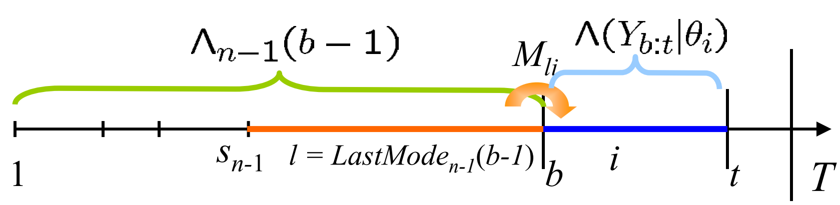


$O(N^T)$ possible segmentations!

Step 1: Optimal segmentation

Finding optimal segmentation is potentially intractable ($O(N^T)$)

Let $\Lambda_n(t) =$ max. Λ derived by dividing $Y_{1:t}$ into n parts



$$\Lambda_n(t) = \max_{i,b} \Lambda_{n-1}(b-1) M_{li} \Lambda(Y_{b:t}|\theta_i)$$

Dynamic Programming Algorithm

$$\Lambda_n(t) = \max \Lambda \text{ by dividing } Y_{1:t} \text{ into } n \text{ parts}$$
$$LastMode_n(t) = \text{label of the last segment that achieves this}$$
$$LastStart_n(t) = \text{start time of the above last segment}$$

$$\Lambda_1(t) = \max_i \Lambda(Y_{1:t}|\theta_i)$$
$$LastMode_1(t) = \arg\max \Lambda(Y_{1:t}|\theta_i)$$

$$\text{while } 1 \leq n \leq \lfloor \frac{T}{T_d} \rfloor \text{ and } nT_d \leq t \leq T$$
$$\Lambda max_n(t) = \max_{\substack{1<i<N \\ (n-1)T_d \leq b \leq t-T_d}} [\Lambda max_{n-1}(b-1)\Lambda(Y_{b:t}|\theta_i)M_{li}]$$

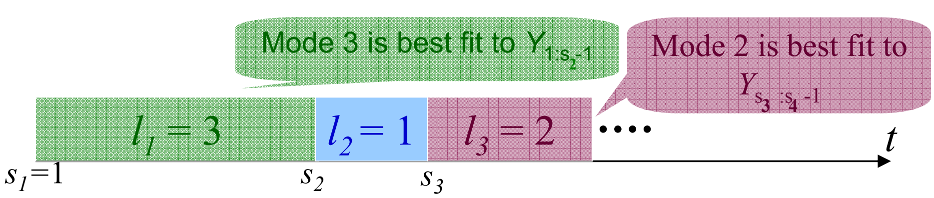
Using this, we can find the best segmentation as the one that achieves $\max_{1 \leq n \leq \lfloor \frac{T}{T_d} \rfloor} \{\Lambda max_n(T)\}$ with complexity $O(NT^3)$

Step 2: Fitting the best continuous model

- For the optimal segmentation determined in Step 1, we fit the best continuous parameters for each mode, by maximizing

$$\Gamma(\mu, \Sigma, A_i, Q_i, R_i) = \mathrm{E}\left[\log \Lambda(Z_{1:N}|\theta_i)|Y_{1:n}\right]$$

where Z is the "complete" data, *i.e.*, both the observed variables (Y) and the state variables x_k, k=1…n.

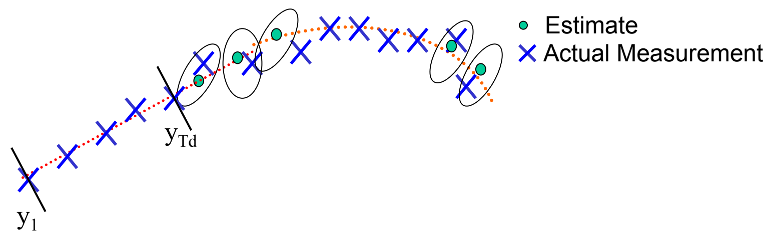


Initialization

Proposed algorithm → local maximum → need initial conditions

We know system stays in a mode for at least T_d

$Y_{1:T_d}$ ⟶ best model for mode 1

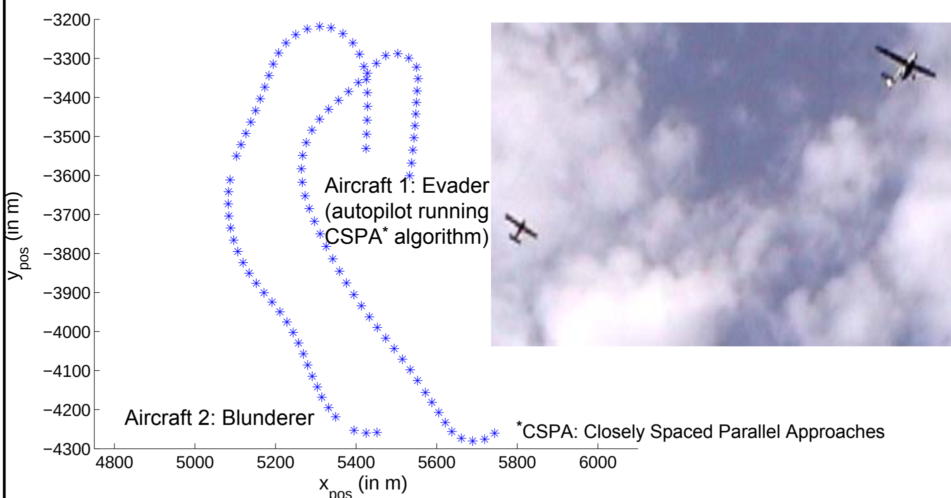


We estimate the number of discrete modes in the system (*N*), the initial segmentation, and the initial continuous model.

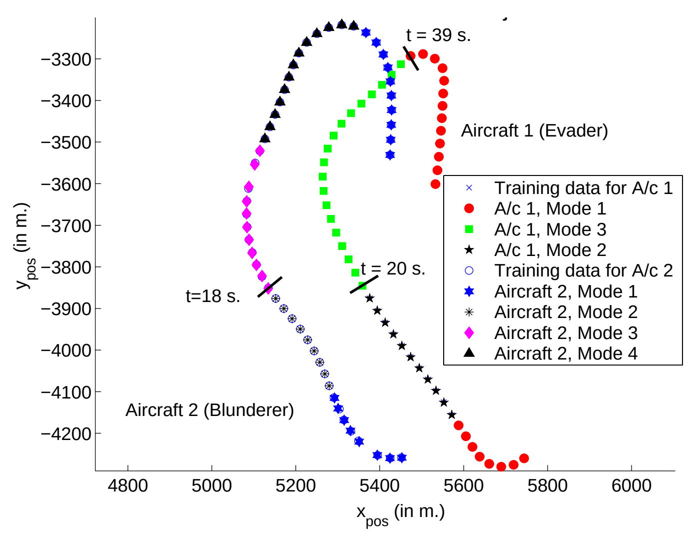
Related Work

- Models for Motion Capture (Li et al.)
- Time Series Analysis (Shumway and Stoffer)
- Hybrid Estimation Algorithms (Bar-Shalom, Blom et al.)
- Observability and Identifiability of Hybrid Systems (Vidal, Soatto, Sastry, Bemporad et al., Balluchi et al.)
- System Identification/ Subspace Identification Methods (Ljung, De Moor, Van Overschee, Vidal, Ma)
- Dynamic Textures (Soatto et al.)

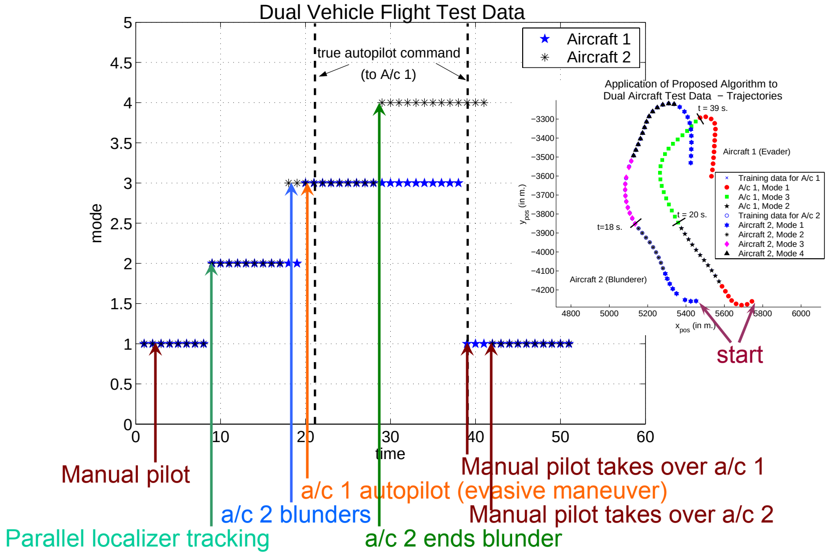
DragonFly Dual Aircraft Test



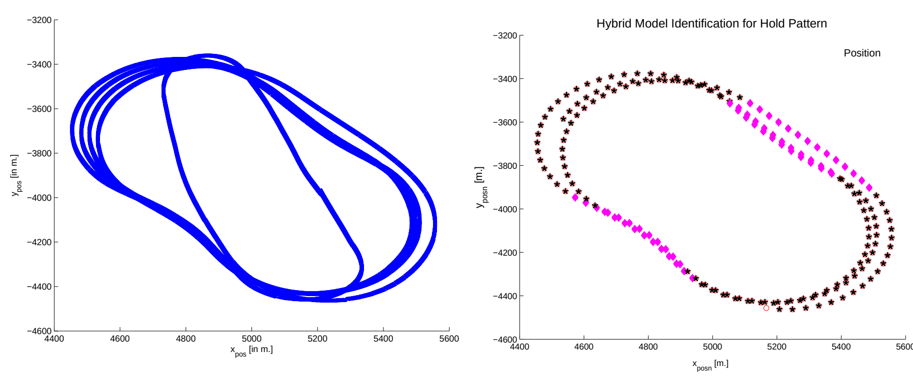
Results



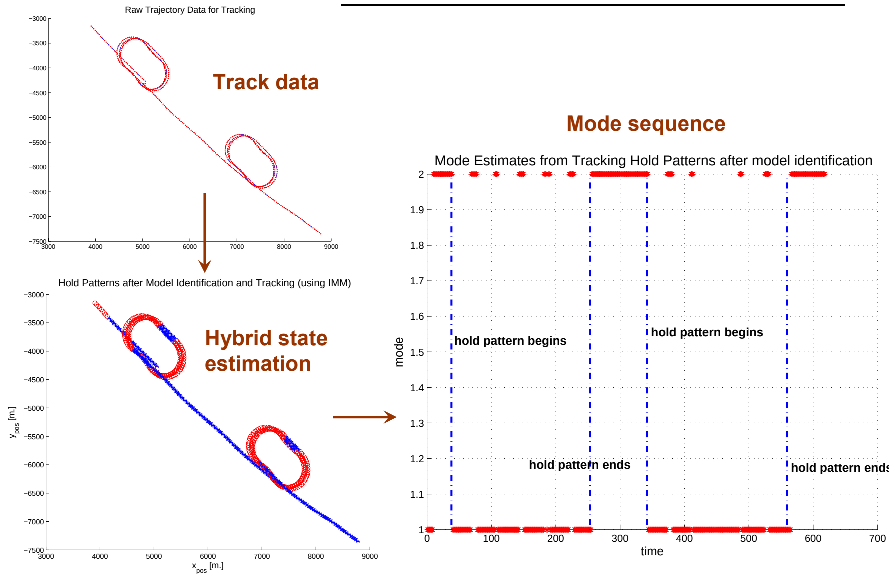
Validation



Holding Pattern Data



State Estimation for a Hold Pattern



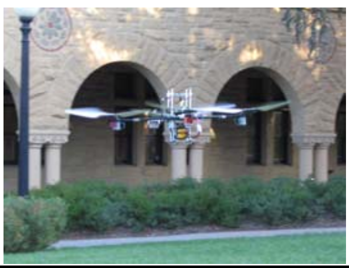
Summary and new directions

- Dynamic programming algorithm to infer stochastic linear hybrid models from time series data
- Tested on UAV and ETMS flight data
- Current work:
  - Incorporating dependence on continuous state
  - State estimation and identity management
  - Applications:
    - Time series data from *Drosophila* (with Jeff Axelrod, Stanford)
    - Flight/weather data from NASA (with Nikunj Oza, NASA Ames)
    - STARMAC project

New Testbed: STARMAC

Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC)

- Quadrotor Design
- Autonomous Control
- Wireless
- Full Onboard Sensing
  - IMU, GPS, SODAR



Ground Station

- Mobile User Interface
- Communicates with fleet: 1 to 8 vehicles
- Optional Joystick Interface

STARMAC Flight Tests

Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC)

Autonomous Flight Demonstration