Hybrid Estimation for Fault Detection and More

Lars Blackmore

April 6, 2005

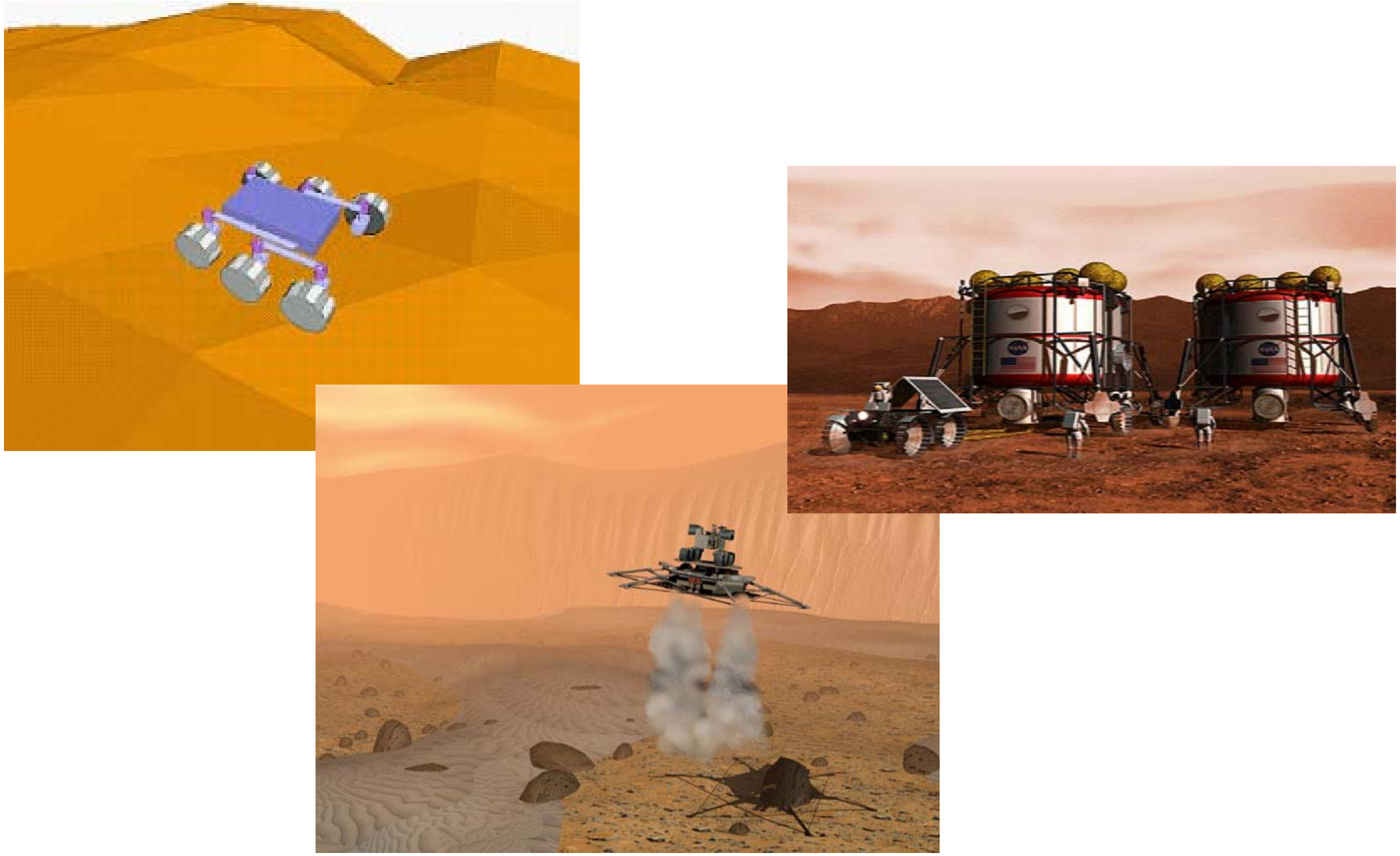
Overview

- Background
  - Kalman Filter review
  - Fault detection using Kalman Filters (Multiple Model method)
- Fault detection (and more) using Hybrid Estimation
  - Why Hybrid?
  - Modeling
  - Technical challenges
  - Current state of the art

60s-80s

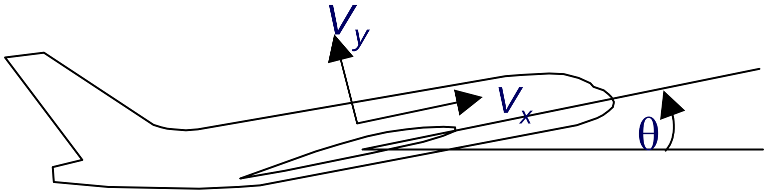
90s - present

Fault Detection



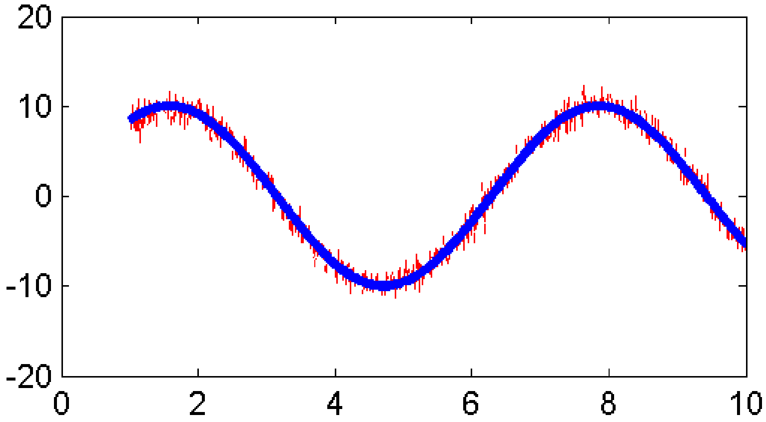
Kalman Filter Review

- Problem:
  - Given a continuous dynamic system model:



$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, t) + \upsilon(t)$$
$$\mathbf{y}_{t+1} = g(\mathbf{x}_{t+1}, \mathbf{u}_t) + \omega(t)$$

  - And a set of noisy observations:



  - Estimate the ‘hidden’ state of the system

Kalman Filter Review

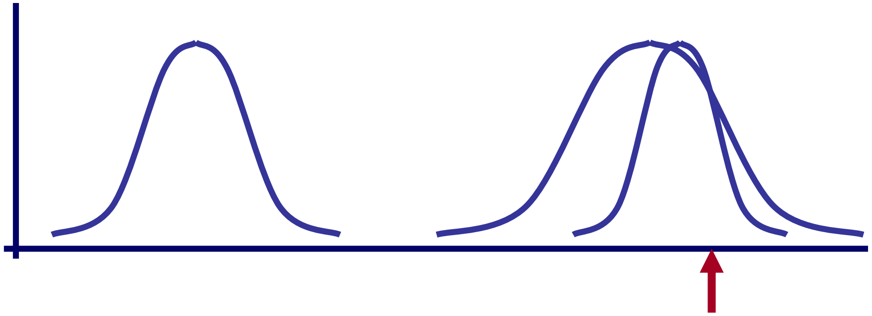
- Solution: Kalman Filters
  - Calculate **belief state** about hidden variables

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t}, \mathbf{u}_{1:t})$$

  - Approximate as Gaussian

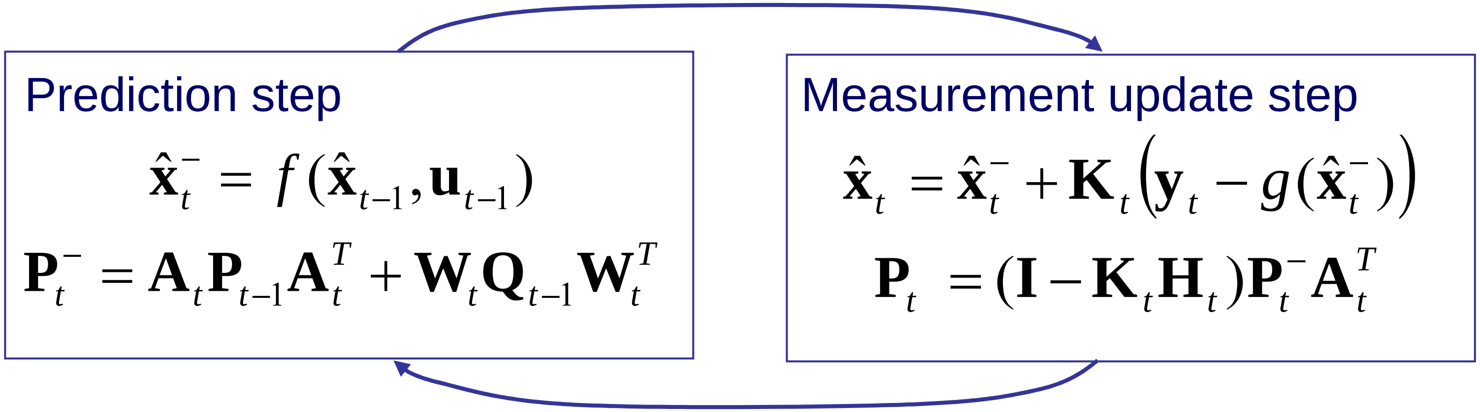
$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t}, \mathbf{u}_{1:t}) \sim N(\hat{\mathbf{x}}_t, \mathbf{P}_t)$$

  - Predict/update cycle:
    1. Start with belief state at t-1
    2. Predict belief state at t using system model
    3. Use measurement at t to adjust the belief state



Kalman Filter Review

- Details
  - KF equations:



Prediction step

$$\hat{\mathbf{x}}_t^- = f(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1})$$

$$\mathbf{P}_t^- = \mathbf{A}_t\mathbf{P}_{t-1}\mathbf{A}_t^T + \mathbf{W}_t\mathbf{Q}_{t-1}\mathbf{W}_t^T$$

Measurement update step

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \mathbf{K}_t\left(\mathbf{y}_t - g(\hat{\mathbf{x}}_t^-)\right)$$

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_t^-\mathbf{A}_t^T$$

  - Linear systems:
    - State distribution **is** Gaussian, KF is **exact**
  - Nonlinear systems:
    - Gaussian assumption is an approximation
    - Extended Kalman Filter accurate to 1st order
    - Unscented Kalman Filter accurate to 2nd order

Overview

- Background
  - Kalman Filter review
  - Fault detection using Kalman Filters (Multiple Model method)

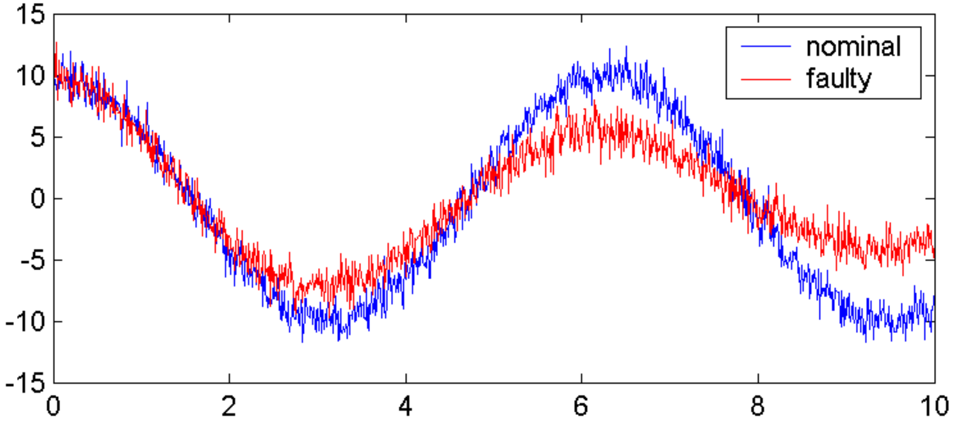
60s-80s

- Fault detection (and more) using Hybrid Estimation
  - Why Hybrid?
  - Modeling
  - Technical challenges
  - Current state of the art

90s - present

Multiple Model Fault Detection

- Fault detection:
  - Is the system operating nominally or is it faulty?

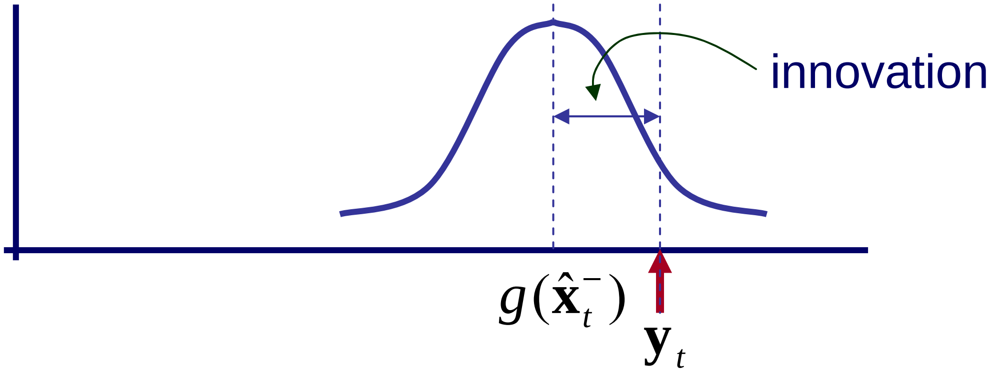


- Assume models known for both cases
  - Which model most likely given observations and inputs?
- How can we use Kalman Filters here?

Multiple Model Fault Detection

- KF predicts distribution of observation given inputs at each time step

- KF ‘innovation’ is discrepancy between expectation and actual observation



- Can use this to determine agreement between model and observations

Multiple Model Fault Detection

- Idea:
  - Use a Kalman Filter for each model
  - Small innovations → model and observations agree
  - Large innovations → model and observations disagree
  - So compare innovations from faulty and nominal KF
  - If innovations smaller for faulty KF, diagnose a fault

Multiple Model Fault Detection

- More formally
  - Let each model be denoted by H_i (e.g. H_0=nominal, H_1=faulty)
  - Assume some belief about each model at time *t-1*:

$$p_i(t-1) = p(H_i \mid \mathbf{y}_{1:t-1})$$

  - We want posterior probability at time t:

$$p_i(t) = p(H_i \mid \mathbf{y}_{1:t})$$

  - Use Bayes' Rule:

$$p_i(t) = \frac{p(\mathbf{y}_t \mid H_i, \mathbf{y}_{1:t-1})\, p_i(t-1)}{\sum_{j=1}^{n} p(\mathbf{y}_t \mid H_j, \mathbf{y}_{1:t-1})\, p_j(t-1)}$$

  - We can calculate $p(\mathbf{y}_t \mid H_i, \mathbf{y}_{1:t-1})$

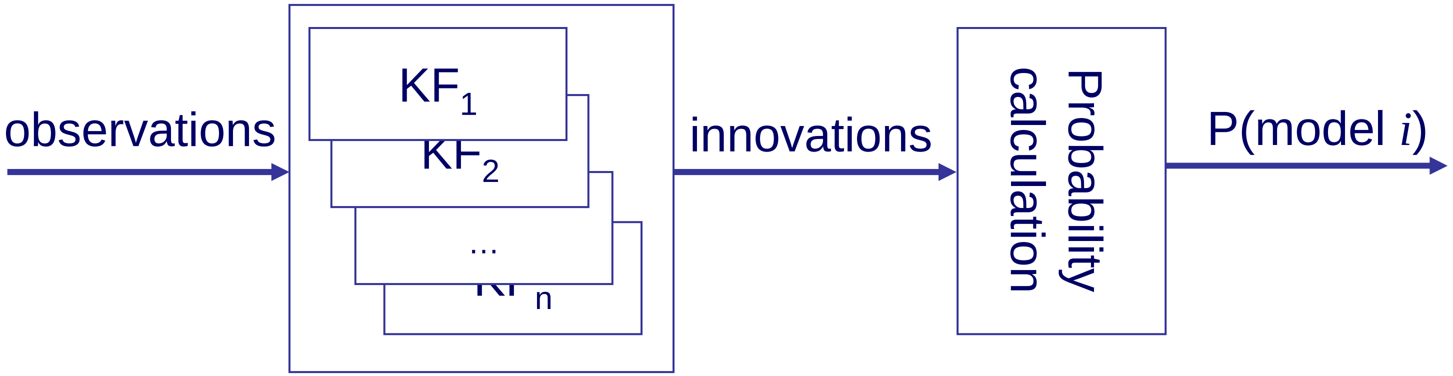
Multiple Model Fault Detection

- We can calculate $p(\mathbf{y}_t \mid H_i, \mathbf{y}_{1:t-1})$
  - using the Kalman Filter innovation

$$p(\mathbf{y}_t \mid H_i, \mathbf{y}_{1:t}) = C_t e^{-\frac{1}{2}\mathbf{i}_t^T \mathbf{V}_t \mathbf{i}_t}$$

Innovation at time t

- So by tracking n Kalman Filters we can calculate the probability of each model given the observations


observations

KF_1

KF_2

…

KF_n

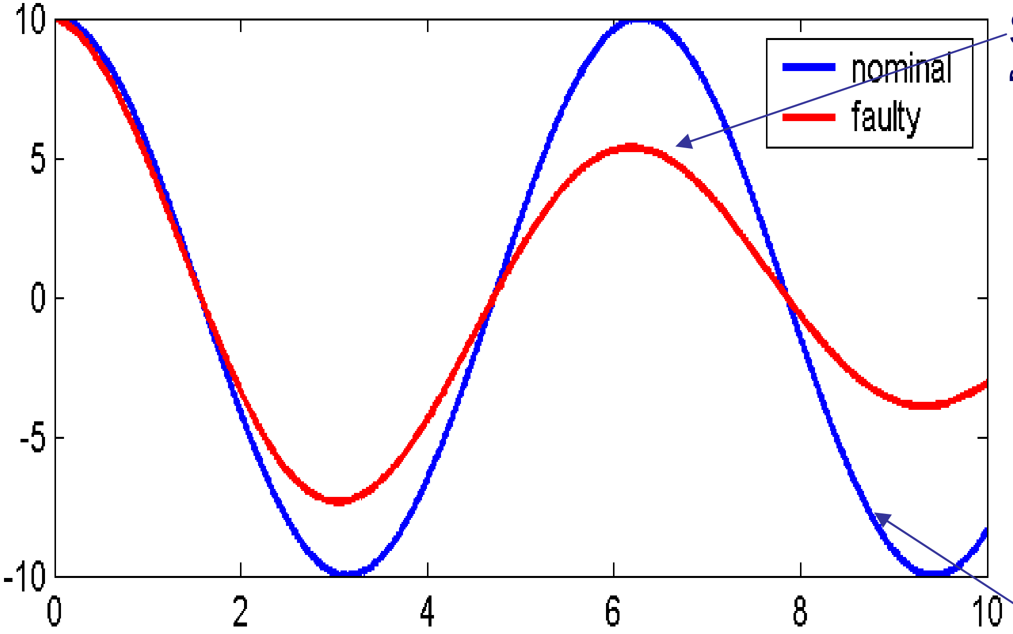
innovations

Probability calculation

P(model i)

  - Problem solved?

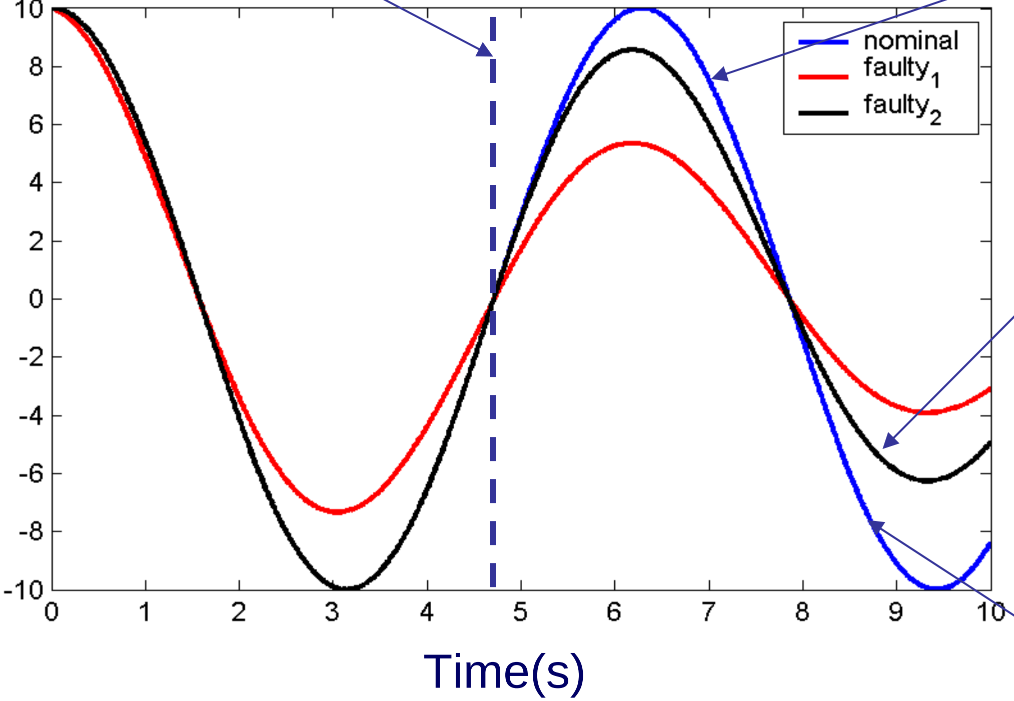
Multiple Model Fault Detection



Simulation of 'faulty' model

Simulation of 'nominal' model

Multiple Model Fault Detection



Fault occurs here

Simulation of 'faulty' model

Ground truth

Simulation of 'nominal' model

Multiple Model Fault Detection

- Main problem:
  - Our model of the failure-prone system is inadequate

- Challenges (rest of the lecture):
  - Model failure-prone systems
  - Reason about failure-prone systems (detect faults and more)

Overview

- Background
  - Kalman Filter review
  - Fault detection using Kalman Filters

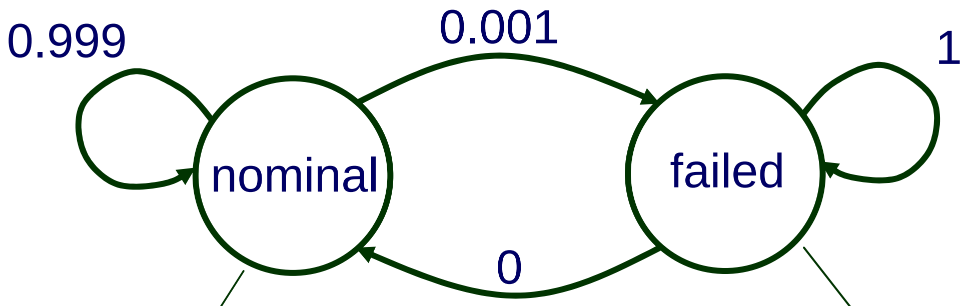
60s-70s

- **Fault detection (and more) using Hybrid Estimation**
  - Why Hybrid?
  - Modeling
  - Technical challenges
  - Current state of the art

80s - present

Hybrid System Models

- Better model for our system:
  - **Discrete modes** and transitions between them



  - **Continuous dynamics** corresponding to each mode

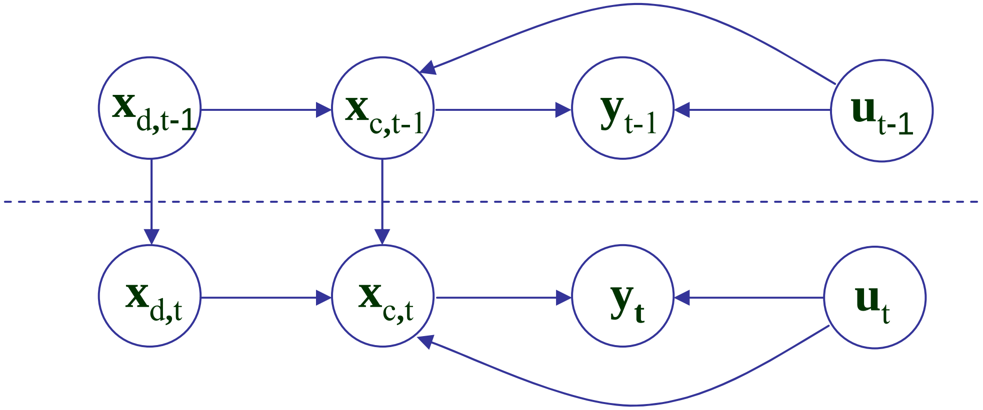
$$\mathbf{x}_{t+1} = f_{\text{nominal}}(\mathbf{x}_t, \mathbf{u}_t, t) + \upsilon_{\text{nominal}}(t)$$
$$\mathbf{y}_{t+1} = g_{\text{nominal}}(\mathbf{x}_{t+1}, \mathbf{u}_t) + \omega_{\text{nominal}}(t)$$

$$\mathbf{x}_{t+1} = f_{\text{failed}}(\mathbf{x}_t, \mathbf{u}_t, t) + \upsilon_{\text{failed}}(t)$$
$$\mathbf{y}_{t+1} = g_{\text{failed}}(\mathbf{x}_{t+1}, \mathbf{u}_t) + \omega_{\text{failed}}(t)$$

Hybrid System Models

- System now has **hybrid** state $\mathbf{x}=\{\mathbf{x}_{c,t}\ \mathbf{x}_{d,t}\}$
  - Continuous state $\mathbf{x}_{c,t}$
  - Discrete modes $\mathbf{x}_{d,t}$

- Our model is a Hidden Markov Model:

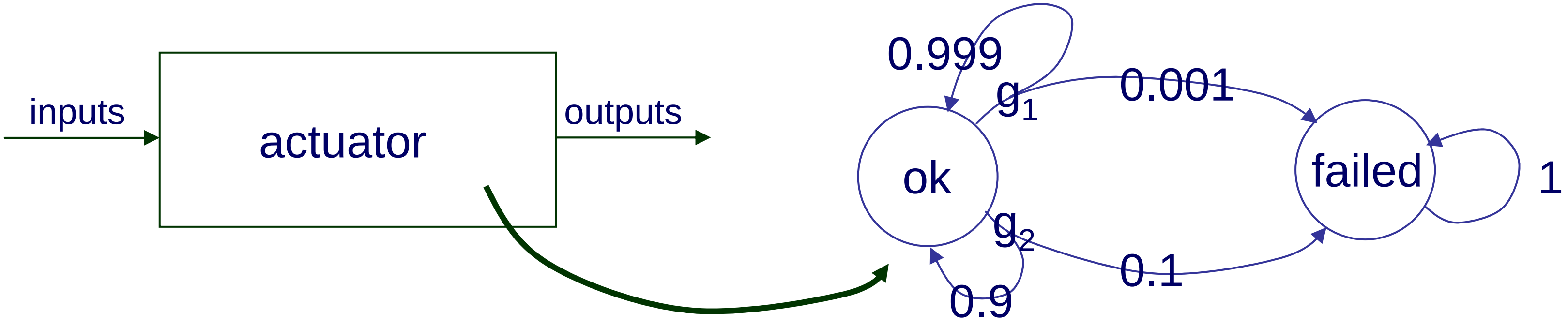


Hybrid System Models

- Hybrid models not just for failure detection
- Many systems have both discrete and continuous state even in normal operation:
  - Hardware/software controlling physical system
    - e.g. Mars rover, robot manipulator
  - Systems with valves, switches, doors
    - Lunar habitat

Hybrid System Models

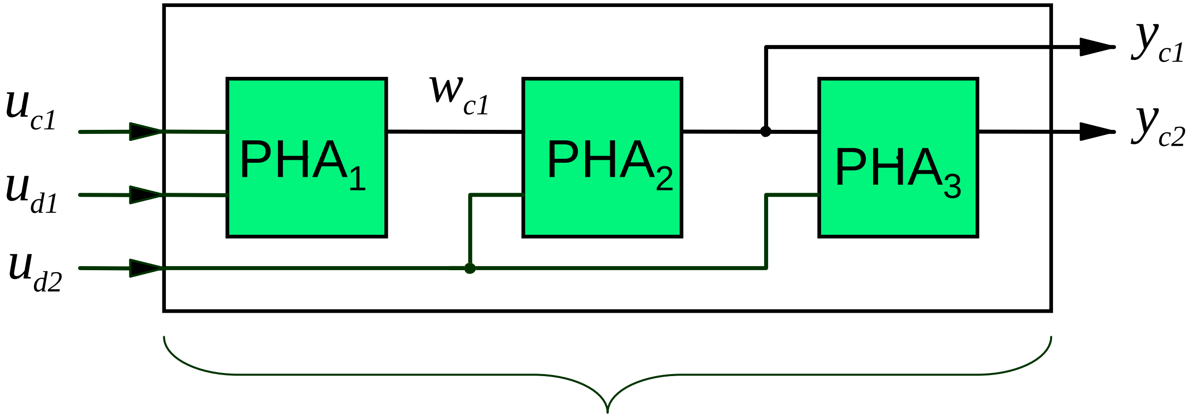
- Even better model:
  - Model each component as Probabilistic Hybrid Automaton



- Main difference: guards on discrete transitions
  - Transition probability conditioned on component **inputs** and component **continuous state**
- Example:
  - More likely to fail if temperature over safe threshold

Hybrid System Models

- ‘*Compose*’ PHA components to form overall system

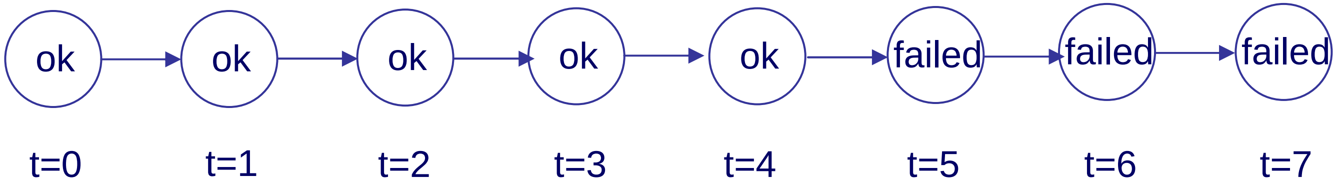


n components

- With m modes per PHA, composed PHA has $O(m^n)$ modes overall

Reasoning about Hybrid System Models

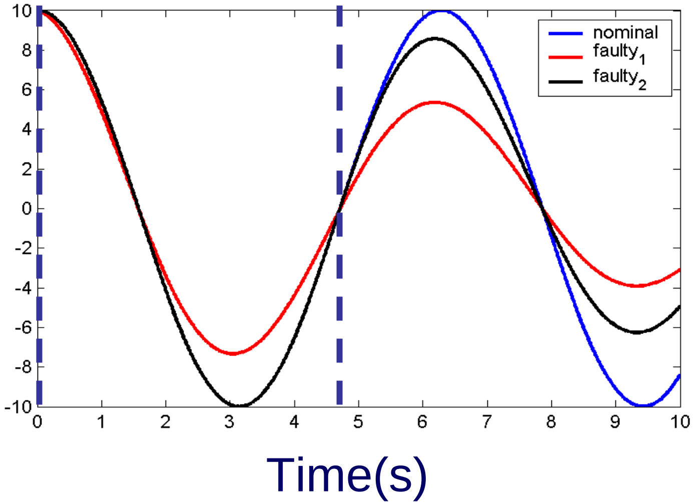
- Now we have **trajectories** of discrete modes
  - Example of a possible mode trajectory of the system:



ok → ok → ok → ok → ok → failed → failed → failed

t=0 t=1 t=2 t=3 t=4 t=5 t=6 t=7

- Known system dynamics for **each trajectory**



Hybrid Estimation

- Problem:
  - Estimate the **hybrid state** of a system

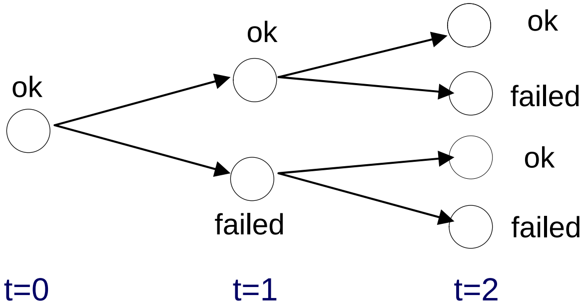
$$p(\mathbf{x}_{d,t},\mathbf{x}_{c,t}\,|\,\mathbf{y}_{1:t},\mathbf{u}_{1:t})$$

  - Given model, observations and system inputs

Hybrid Estimation

Approach:

1. Each time t, consider all possible mode trajectories



2. Calculate the distribution over **trajectories** and $\mathbf{x}_{c,t}$

$$p(\mathbf{x}_{d,1:t}, \mathbf{x}_{c,t} \mid \mathbf{y}_{1:t}, \mathbf{u}_{1:t})$$

3. Sum over the trajectories

$$p(\mathbf{x}_{d,t}, \mathbf{x}_{c,t} \mid \mathbf{y}_{1:t}, \mathbf{u}_{1:t}) = \sum_{\mathbf{x}_{d,1:t-1}} p(\mathbf{x}_{d,1:t}, \mathbf{x}_{c,t} \mid \mathbf{y}_{1:t}, \mathbf{u}_{1:t})$$

Hybrid Estimation

- How to calculate distribution for a trajectory?

$$p(\mathbf{x}_{d,1:t},\mathbf{x}_{c,t}\mid\mathbf{y}_{1:t}) = \underbrace{p(\mathbf{x}_{d,1:t}\mid\mathbf{y}_{1:t})}\,\underbrace{p(\mathbf{x}_{c,t}\mid\mathbf{x}_{d,1:t},\mathbf{y}_{1:t})}$$

Probability of trajectory
given observations

Calculate using **belief state update**

Distribution of continuous state given
mode trajectory and observations
→given by kalman filter

- Belief state update:

$$p(\mathbf{x}_{d,1:t}\mid\mathbf{y}_{1:t}) = b(\mathbf{x}_{d,1:t}) = P_O(\mathbf{y}_t)\cdot P_T(\mathbf{x}_{d,t})\cdot b(\mathbf{x}_{d,1:t-1})$$

Hybrid Estimation

- Observation function P_O

$$P_O = p(\mathbf{y}_t \mid \mathbf{x}_{d,1:t}, \mathbf{y}_{1:t-1})$$

  - Track a Kalman Filter for the trajectory $\mathbf{x}_{d,1:t}$
  - Use KF innovation (as MM method)

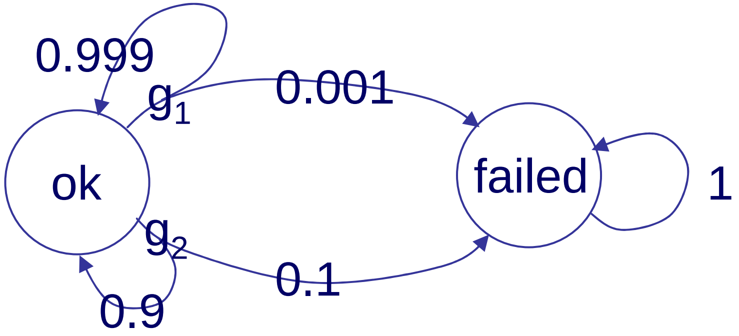
$$p(\mathbf{y}_t \mid \mathbf{x}_{d,1:t}, \mathbf{y}_{1:t-1}) = C_t e^{-\frac{1}{2}\mathbf{i}_t^T \mathbf{V}_t \mathbf{i}_t}$$

Hybrid Estimation

- Transition function P_T

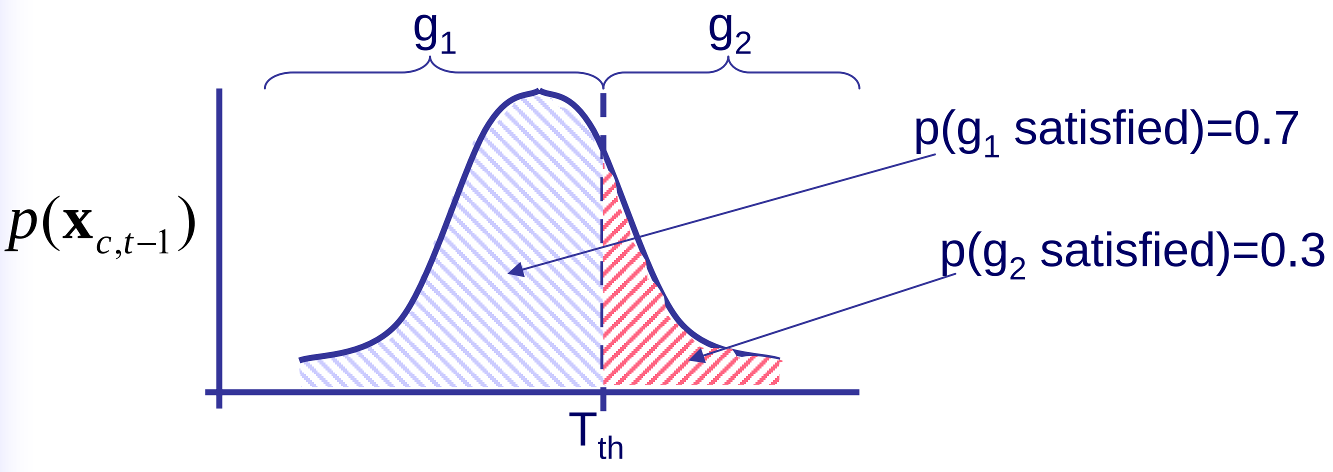
$$P_T = p(\mathbf{x}_{d,t} \mid \mathbf{x}_{d,1:t-1}, \mathbf{y}_{1:t-1})$$

  - What does the probability of transitioning from a given state at t-1 to a given state at t depend on?
  - In a PHA, P_T depends on which guard is satisfied
    - This depends on:
      - Inputs $\boldsymbol{u}_{t-1}$
      - Previous continuous state $\boldsymbol{x}_{c,t-1}$



Hybrid Estimation

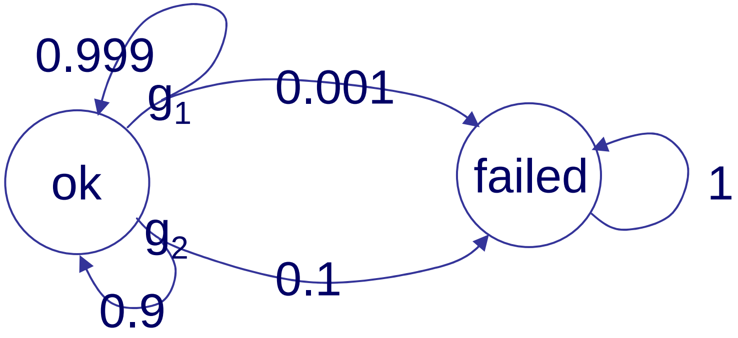
- Probability of guards being satisfied



p(g_1 satisfied)=0.7

p(g_2 satisfied)=0.3

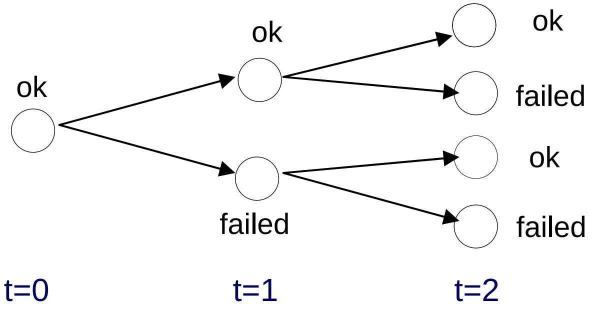
- P(ok→failed)=(0.7)(0.999)+(0.3)(0.9)



Hybrid Estimation

Approach:

1. Each time t, consider all possible mode trajectories



2. Calculate the distribution over **trajectories** and $\mathbf{x}_{c,t}$

$$p(\mathbf{x}_{d,1:t}, \mathbf{x}_{c,t} \mid \mathbf{y}_{1:t}, \mathbf{u}_{1:t})$$

3. Sum over the trajectories

$$p(\mathbf{x}_{d,t}, \mathbf{x}_{c,t} \mid \mathbf{y}_{1:t}, \mathbf{u}_{1:t}) = \sum_{\mathbf{x}_{d,1:t-1}} p(\mathbf{x}_{d,1:t}, \mathbf{x}_{c,t} \mid \mathbf{y}_{1:t}, \mathbf{u}_{1:t})$$

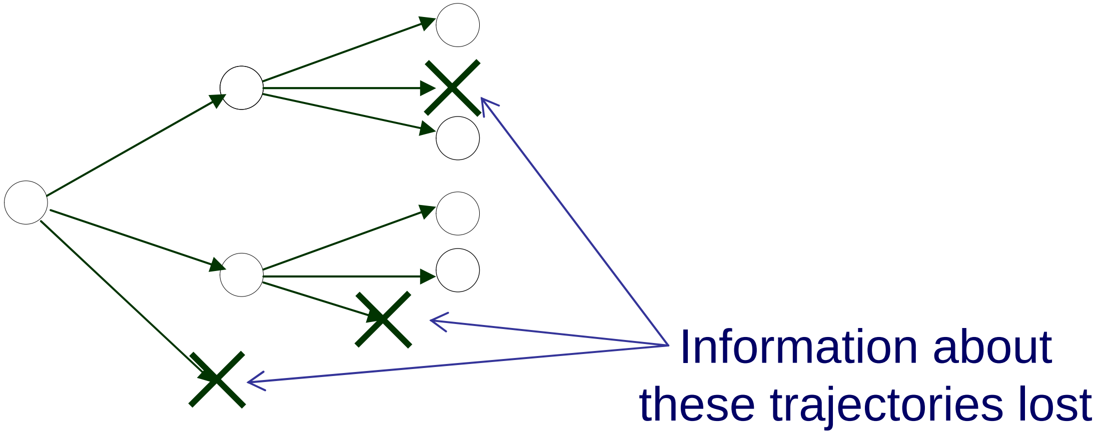
- Problems?

Hybrid Estimation

- Need a KF for **each possible mode trajectory**
  - – Why is this infeasible?
- Exponential in the number of time steps
- Exponential in number of components

Approximate Hybrid Estimation

- In practice, we must **approximate**
  - Only track some mode trajectories



Information about these trajectories lost

- Key technical challenge:
  - Which ones to track?

Approximate Hybrid Estimation

- Two main successful approaches:
  1. K-best enumeration
     - Greedily pick highest probability trajectories
  2. Particle filtering
     - Stochastic sampling approach

K-Best Enumeration

1. Store k trajectories at time t
2. Enumerate the successors at time $t+1$ in terms of posterior probability $b(\mathbf{x}_{d,1:t})$
3. Retain the k with the highest posterior probability
4. Discard remainder

- Complexity?

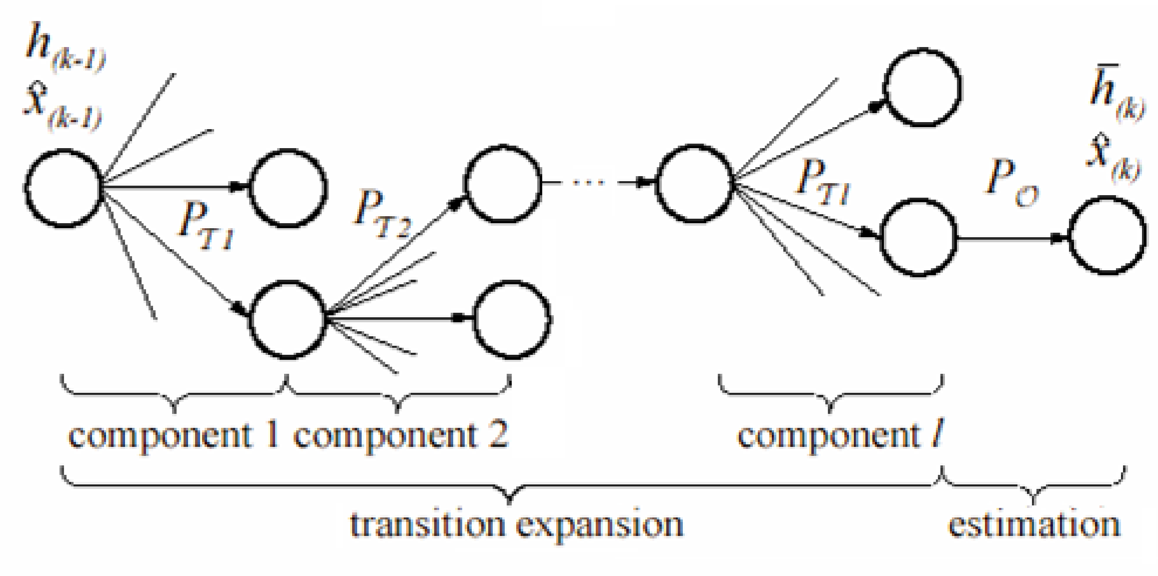
K-Best Enumeration

- Number of successors can be huge
  - Complexity $O(b^n)$ for b successor modes per component and n components
- How to avoid enumeration of **all** successors?

- Key ideas:
  - Use **conditional independence** of components
  - Frame problem as **path search**

K-Best Enumeration

- Path search through component mode assignments



- Use A* directed search

A* search

- Choose successors to node s greedily, based on:

$$f(s) = g(s) + h(s)$$

Cost of partial path from start node to node s

Heuristic function, lower bound on cost to go

- Guaranteed to:
  - Find a path if one exists
  - Find minimum cost path to goal

A* search for K-Best Enumeration

- Cost:
  - What are we trying to optimise?
    - Find maximum probability assignment to all component modes given the observation

Transition probability for component *i*

$$b(\mathbf{x}_{d,1:t}) = b(\mathbf{x}_{d,1:t-1}) \cdot P_O(\mathbf{y}_t) \cdot \prod_{i=1}^{n} P_{T_i}(\mathbf{x}_{d,t})$$

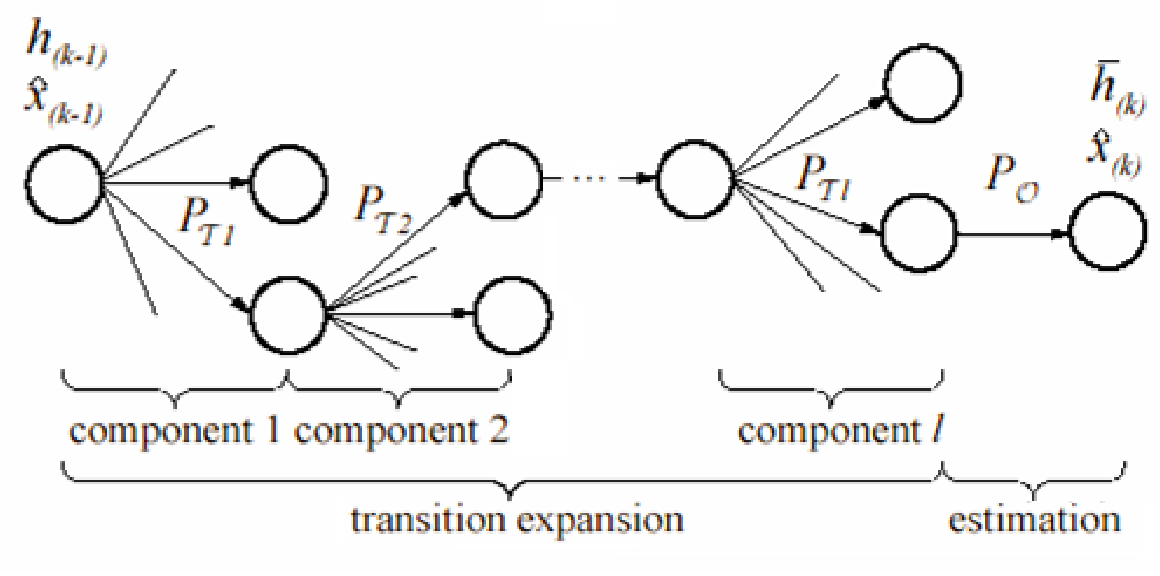
  - Equivalently we can minimise:

$$-\ln b(\mathbf{x}_{d,1:t}) = -\ln b(\mathbf{x}_{d,1:t-1}) - \sum_{i=1}^{n} \ln P_{T_i}(\mathbf{x}_{d,t}) - \ln P_O(\mathbf{y}_t)$$

A* search for K-Best Enumeration

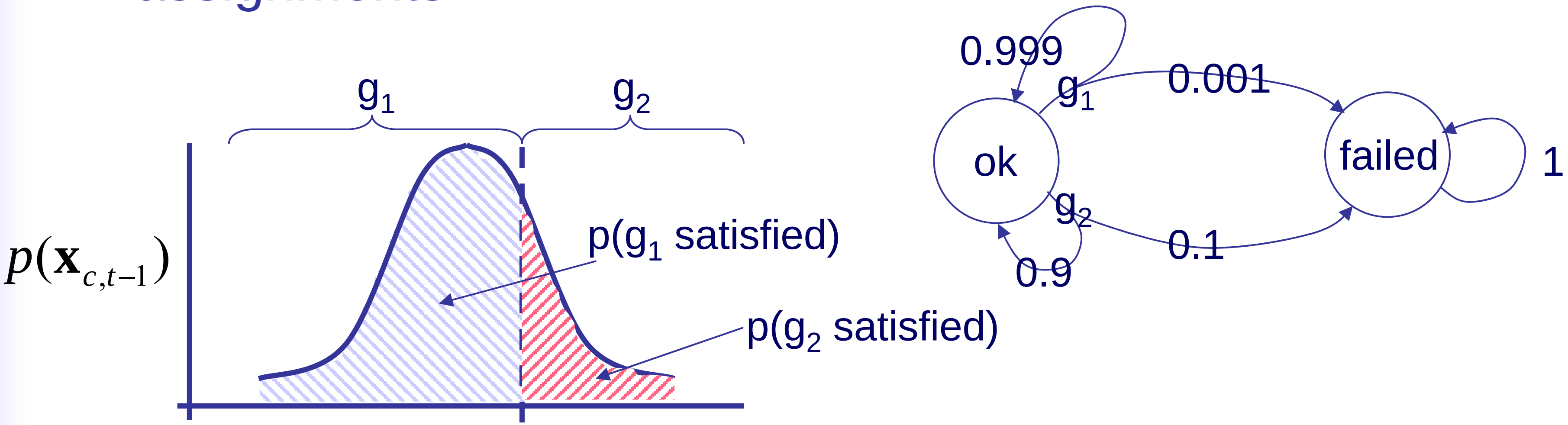
- Now problem can be posed as path search:
  - Arc costs sum to give cost for partial path
  - Arc costs positive

$$g(s) = -\ln b(\mathbf{x}_{d,1:t-1}) - \sum_{i=1}^{n} \ln P_{T_i}(\mathbf{x}_{d,t}) - \ln P_O(\mathbf{y}_t)$$



K-Best Enumeration

- Admissible heuristic?
  - Need an upper bound on probability of remaining mode assignments



- $P_T(\text{ok} \rightarrow \text{failed}) = p(g_1) \cdot 0.001 + p(g_2) \cdot 0.1$
- $P_T(\text{ok} \rightarrow \text{ok}) = p(g_1) \cdot 0.999 + p(g_2) \cdot 0.9$

K-Best Enumeration

- Admissible heuristic:
  - – For each component with mode not yet assigned:
  - – Find the transition with the highest probability
  - – Assume corresponding guard satisfied with probability 1

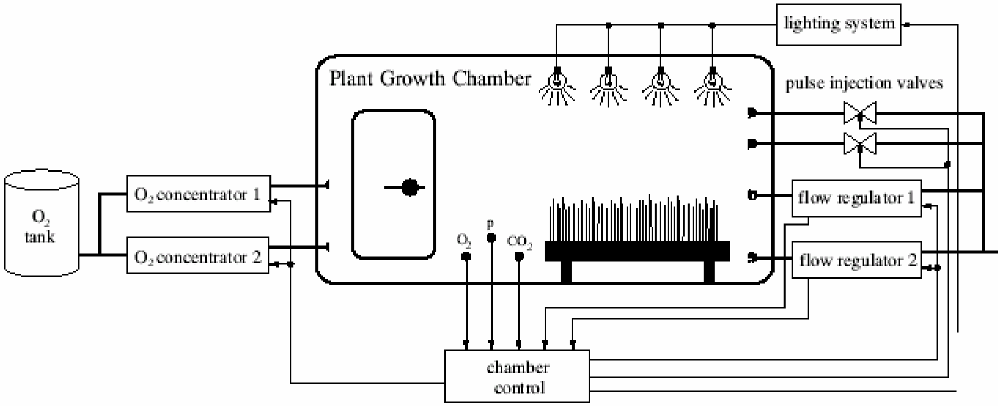
$$h(s) = -\sum_{\substack{\text{unassigned}\\ \text{components}}} \ln\left[\max(\mathrm{P}_{\mathrm{T}})\right]$$

K-Best Enumeration

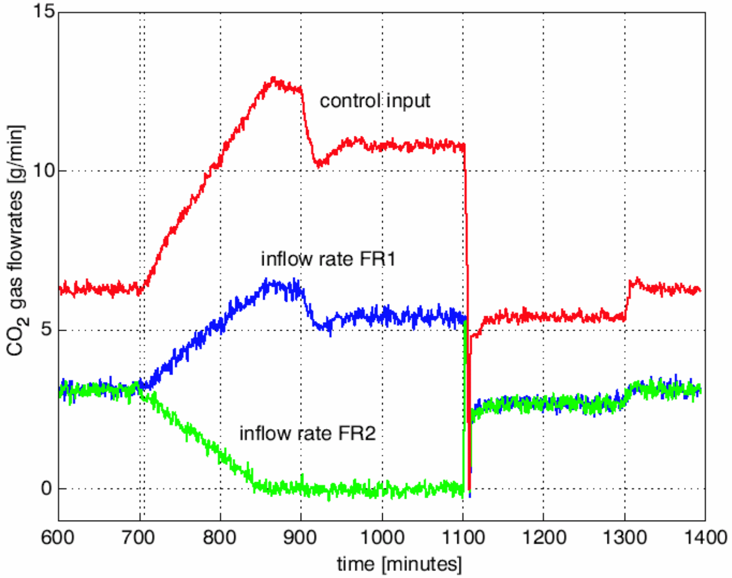
- Summary
  - Exponentiality in time:
    - Store k best trajectories at each time step
  - Exponentiality in number of components:
    - Efficient successor enumeration using A* search

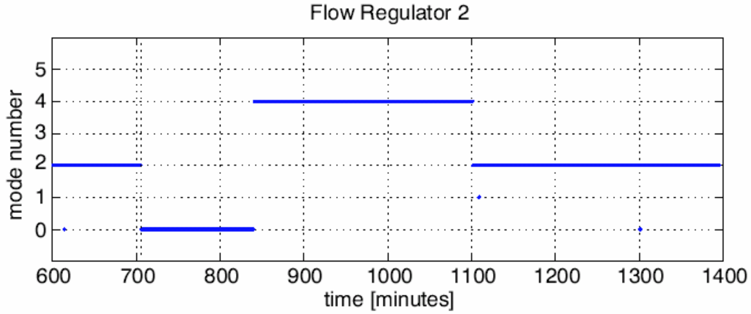
- Fault detection?
  - Probability of a given discrete mode easily obtained from hybrid belief state $p(\mathbf{x}_{d,t}, \mathbf{x}_{c,t} \mid \mathbf{y}_{1:t}, \mathbf{u}_{1:t})$
  - Fault detection problem encompassed in hybrid estimation framework

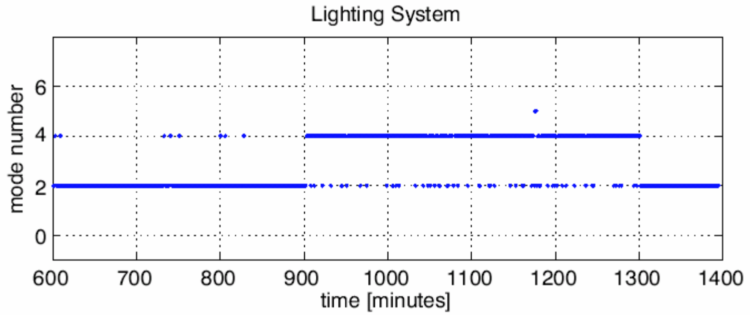
K-Best Enumeration




Courtesy NASA JPL







Discrete variables: operational mode {closed, open, stuck-closed, stuck-open}

Continuous variables: CO_2 flow, CO_2 & O_2 conc.

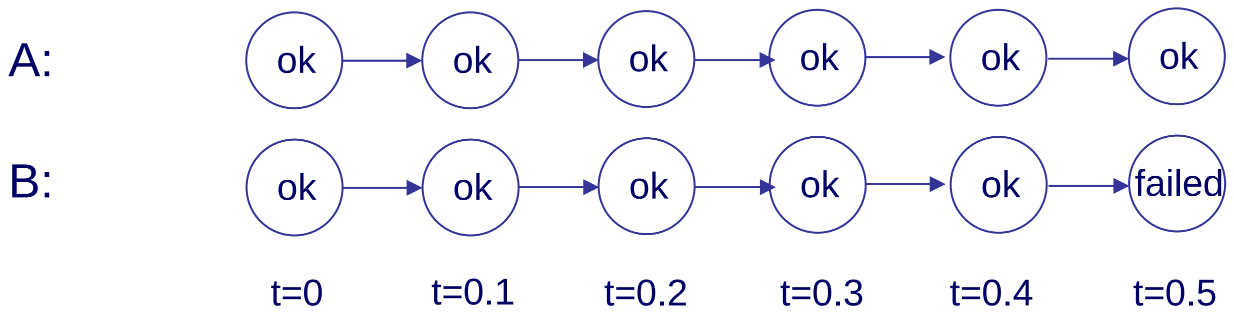
K-Best Enumeration

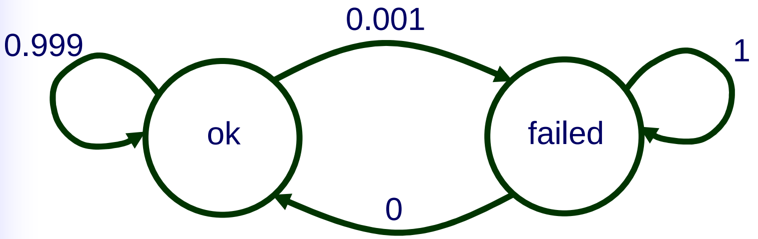
- Limitations?
- K-best enumeration discards many trajectories
  - Essentially greedy beam search
- Problem?

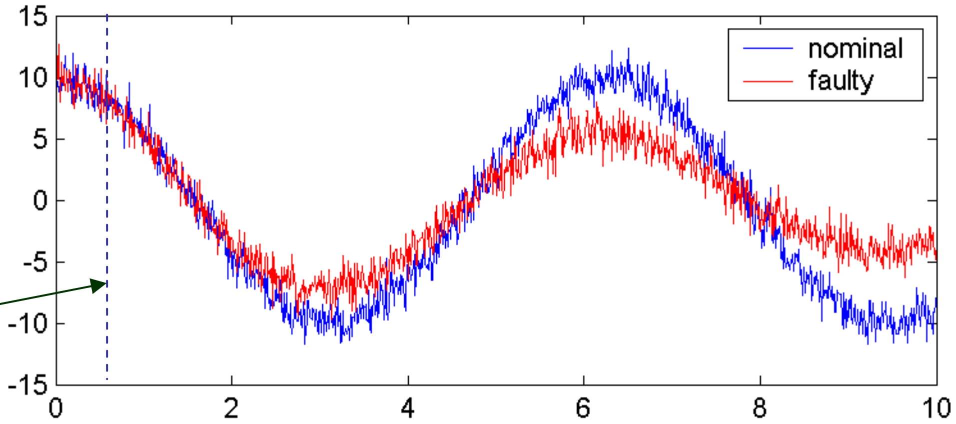
K-Best Enumeration

- Simple fault example
  - Consider candidate trajectories A and B

True trajectory







Fault occurs here

- Which trajectory is most likely?

Recent advances

- Particle filtering
- Smoothing
- Merging
- Risk sensitive estimation
- Mixed greedy/stochastic search

Questions?