Finite Horizon Control Design for Optimal Model Discrimination

Lars Blackmore and Brian Williams

November 27, 2006

Example Application

• Aircraft fault diagnosis
  – Finite set of models for system dynamics
  – Given data, estimate most likely model

• Standard approach: Multiple Model fault detection
  – Select between a finite set of stochastic linear dynamic systems using Bayesian decision rule

• Linearised aircraft model
  – Longitudinal dynamics

• Elevator actuator
  – Model 0: Actuator functional, \( B_0 = \begin{bmatrix} k & 0 \end{bmatrix} \)
  – Model 1: Actuator failed, \( B_1 = \begin{bmatrix} 0 & 0 \end{bmatrix} \)

Problem Statement

• Design a finite sequence of control inputs \( u = [u_1, \ldots, u_k] \) to minimize the probability of model selection error
  – Between two discrete-time, stochastic linear dynamic models
  – Subject to constraints on inputs and expected state

Example Experiment

Context – Model Selection

Model Identification
  – Which model best explains a given data set?
  1. Parameter adaptation
  2. Selection from a finite set of models
  • Model Selection

Control Design for Model Discrimination

• System inputs greatly affect performance of model selection algorithm
  – ‘Active’ model selection designs system inputs to discriminate optimally between models
  – Previous approaches include (Esposito\(^2\), Goodwin\(^3\), Zhang\(^4\))
  – Designed inputs have limited power to restrict effect on system
  – Maximization of information measure or minimization of detection delay
  – We extend these approaches as follows:
    1. Design inputs with explicit state and input constraints
    2. Bayesian cost function: probability of model selection error

• We present novel method that uses finite horizon constrained optimization approach to design control inputs for optimal model discrimination
  – Key idea: Minimise probability of model selection error subject to explicit input and state constraints

\[ x = \begin{bmatrix} V_x \\ V_y \\ \phi \end{bmatrix}, \quad \frac{dx}{dt} = \begin{bmatrix} A_x & B_u & w_x \\ C_x & D_u & v_y \\ 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} u, \quad y = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \]

\[ v_x \sim N(0, R), \quad w_x \sim N(0, Q) \]
Slide 3
L3 link discrimination to diagnosis
Lars, 12/8/2005

Slide 4
LB7 link to aircraft eg
Lars Blackmore, 12/2/2005
L11 link to aircraft eg
Lars, 12/10/2005
L12 talk about going to hard limits like MPC
Lars, 12/10/2005
L13 talk about interpretation of information?
Lars, 12/10/2005

Slide 5
L9 put in a picture?
Lars, 12/8/2005
Example Experiment

1. Let transients decay to zero
2. Request a large elevator displacement
   - Model 0: Actuator is working, large response observed
   - Model 1: Actuator failed, no response

Key ideas

1. Separate predicted distribution of observations corresponding to different models

   ![Graph showing probability distributions for different models]

2. Can view problem as finite horizon trajectory design
   - Planning distribution of future state
   - LP, MILP, QP commonly used

Trajectory Design Formulation – Cost Function

- Bayesian decision rule:
  - Choose $H_i$ if: $P(H_0 | y, u) > P(H_i | y, u)$

- $P(error | u) =$ probability that the wrong model is selected:

   $P(error | u) = \int p(y | H_i, u) P(H_i) \, dH_i$

   $P(error | u) = \int p(y | H_i, u) P(H_i) \, dH_i$

   ![Graph showing probability of error]

Technical Approach: Assumptions

- Two discrete-time, linear dynamic models, $H_0$ and $H_1$, can capture possible behaviors of system
  - One of models is true state of world for entire horizon
- Prior information about models:
  - Some prior distribution over the two models
  - Distribution over initial state conditioned on model
    ...may be viewed as current belief state from an estimator
- Gaussian process and observation noise
- Bayesian model selection used
  - Batch selection

Technical Approach: Outline

1. Define Bayesian cost function (probability of error)
2. Describe analytic upper bound to cost function
3. Show that upper bound is quadratic in inputs
4. Show that finite horizon problem formulation can be solved using Quadratic Programming

Pattern Classification, Duda, R., Hart, P., Stork, D., 2000

Trajectory Design Formulation – Cost Function

- The probability of model selection error is:

  $P(error | u) = \int p(y | H_i, u) P(H_i) \, dH_i$

  $P(error | u) = \int p(y | H_i, u) P(H_i) \, dH_i$

- The integral does not have a closed form solution, but an upper bound exists called the Battacharrya Bound

- For Gaussian distributions $p(y | H_0, u) \sim N(\mu_0, \Sigma_0)$ and $p(y | H_1, u) \sim N(\mu_1, \Sigma_1)$:

  $P(error | u) \leq \sqrt{P(H_0)P(H_1)} e^{-\frac{1}{4}(y-u_0)'[\Sigma_0^{-1}+\Sigma_1][y-u_0]+f(\Sigma_0, \Sigma_1)'}$

  $P(error | u) \leq \sqrt{P(H_0)P(H_1)} e^{-\frac{1}{4}(y-u_0)'[\Sigma_0^{-1}+\Sigma_1][y-u_0]+f(\Sigma_0, \Sigma_1)'}$

  Take logarithm:

  $\text{ln}(P(error)) \leq \text{ln}(\text{priors}) - \frac{1}{4}(y-u_0)'[\Sigma_0^{-1}+\Sigma_1][y-u_0]+f(\Sigma_0, \Sigma_1)$

  $\text{ln}(P(error)) \leq \text{ln}(\text{priors}) - \frac{1}{4}(y-u_0)'[\Sigma_0^{-1}+\Sigma_1][y-u_0]+f(\Sigma_0, \Sigma_1)$
L2 mention what y, H and u are
Lars, 12/8/2005

LB21 cut this?
Lars Blackmore, 12/5/2005

L1 mention what y, H and u are
Lars, 12/8/2005

LB9 Go through each term
Lars Blackmore, 12/2/2005
Trajectory Design Formulation – Cost Function

- Important analytic properties:
  1. Expected observations $\mu_0$ and $\mu_1$ are known linear functions of the inputs $u$
  2. The covariances $\Sigma_0$ and $\Sigma_1$ are not functions of the inputs $u$
  3. Covariances and priors do not affect optimization

- Cost function for trajectory design problem:
  \[
  J = f(\text{priors}) - \frac{1}{4}(\mu_1 - \mu_0)'\Sigma_0 + \Sigma_1'(\mu_1 - \mu_0) + f(\Sigma_0, \Sigma_1)
  \]
  - Cost function is quadratic in the inputs

Trajectory Design Formulation - Constraints

- As in many trajectory design problems, we may want to:
  - Ensure fulfillment of task defined in terms of expected state
  - Bound expected state of the system
  - Model actuator saturation
  - Restrict total fuel usage

- All of these are linear constraints

Trajectory Design Formulation - Summary

- What we have is a Quadratic Program:
  1. Cost function that is quadratic in the control inputs
  \[
  J = p_1 u_1 + p_2 u_2 + p_3
  \]
  2. Constraints that are linear in the control inputs
    - E.g. $|x_k| \leq x_{\text{max}}$ for all $k$
  - Quadratic Programs can be solved efficiently

- Now constrained active model discrimination possible:
  - On-line, while fulfilling defined task and ensuring safety

Simulation Results – Active Approach

- Elevator failure scenario
- Linearised, discrete-time longitudinal dynamics
- Pitch rate observed
- Horizon of 40 time steps

\[
\Delta t = 0.5s
\]

\[
\begin{align*}
  x_{k+1} &= A x_k + B u_k + w_k \\
  y_{k+1} &= C x_{k+1} + D u_k + v_k \\
  v_k &\sim N(0, R) \\
  w_k &\sim N(0, Q)
\end{align*}
\]

Results: Constrained Input and State

Battacharrya bound: 0.0036  QP solution time: 0.22s

Results: Manually Generated Sequence

Battacharrya bound: 0.16
Slide 13

LB18 Also mention concave
Lars Blackmore, 12/2/2005

Slide 15

L8 explain what I mean by safety
Lars, 12/8/2005

Slide 17

L7 mention constraints explicitly
Lars, 12/8/2005
Results: Altitude Change Maneuver

- Optimised control input (discrimination optimal)
- Optimised control input (fuel optimal)

Expected altitude (working actuator, discrimination optimal)
Expected altitude (working actuator, fuel optimal)
Expected altitude (faulty actuator)

Limitations

- Linear systems only
  - Linearise about an equilibrium point
  - Feedback linearization
- Not directly minimising the probability of error
  - No guarantees about tightness of bound
  - Low upper bound sufficient in many cases
- Two models only
  - Battacharrya bound does not apply to more than two hypotheses
  - Submission soon on generalized approach

Conclusion

- Novel algorithm for model discrimination between any two linear systems
- On-line solution possible due to efficient Quadratic Programming formulation
- Arbitrary linear state and control constraints can be incorporated
  - Fulfill specified task defined in terms of system state
  - Guarantee safe execution
  - Maintain state within linearisation region
  - ... while optimally detecting failures

Questions?

Results: Constrained Elevator Angle

- Optimized sequence drives aircraft at Short Period Oscillation (SPO) mode

Backup

Results: Constrained Elevator Angle

- Optimized sequence drives aircraft at Short Period Oscillation (SPO) mode

Battacharrya bound: 0.0021
QP solution time: 0.19s
up to now, plan is safe but now go to task fulfillment
Lars Blackmore, 12/2/2005
Alternative Criteria

- **Battacharyya bound**
  \[ (\mu_i - \mu_b) \left[ \Sigma_o + \Sigma_i \right]^{-\frac{1}{2}} (\mu_i - \mu_b) \]

- Baram’s Distance
- KL divergence
  \[ (\mu_i - \mu_b) \Sigma_i^{-\frac{1}{2}} (\mu_i - \mu_b) \]
- ‘Symmetric’ KL divergence
  \[ (\mu_i - \mu_b) \left[ \Sigma_o^{-\frac{1}{2}} + \Sigma_i^{-\frac{1}{2}} \right] (\mu_i - \mu_b) \]
- Information
  \[ f(M_o) \]

Concave Quadratic Programming

- O(N) Linear Programs must be solved
- Each LP typically O(NM) number of simplex ops
- M = # constraints
- N = size of QP = (# output variables) x (horizon length)

Open-loop vs Closed-loop

- Design is open loop
- But can be used within an MPC closed-loop framework
- Efficient QP solution makes this possible

Cost Criterion

- Can be handled in very similar manner, assuming detector is cost-optimal

Unbounded Objective Function

- An optimal solution of negative infinity cannot occur with bounded \( u \) if either covariance > 0
- We can get a p(error) of zero for bounded \( u \) if:
  - One of the priors is zero
  - One of the covariances has zero determinant
- Otherwise for bounded \( u \) we cannot.