Diagnosis as Approximate Belief State Enumeration for Probabilistic Concurrent Constraint Automata

Oliver B. Martin  
Brian C. Williams  
omartin_williams@mit.edu  
Massachusetts Institute of Technology

Michel D. Ingham  
michel.ingham@jpl.nasa.gov  
Jet Propulsion Laboratory

June 2, 2005

Paper Contribution

- **Best-First Belief State Enumeration**
  - Increased PCCA estimator accuracy by computing the Optimal Constraint Satisfaction Problem (OCSP) utility function directly from the HMM propagation equation.
  - Improved PCCA estimator performance by framing estimation as a single OCSP.

Simplified Decent Stage IMU/PS System

- Two Components
  - Inertial Measurement Unit (IMU)
  - Power Switch (PS)

MSL Entry Decent & Landing Sequence

IMU Constraint Automaton
Estimation of PCCA

Compute the belief state for each estimation cycle.

- Belief State Evolution visualized with a Trellis Diagram
- Complete history knowledge is captured in a single belief state by exploiting the Markov property
- Belief states are computed using the HMM belief state update equations

HMM Belief State Update Equations

- Propagation Equation
  \[ P(x_{t+1}^{(0)}, \mu^{(0)}) = \sum_{x_t} P(x_{t+1}^{(0)} = x_t | x_t, \mu^{(0)}) P(x_t | \omega^{(0)}, \mu^{(0)}) \]
- Update Equation
  \[ P(x_{t+1}^{(1)}, \mu^{(1)}) = \frac{P(x_{t+1}^{(1)} | \omega^{(1)}, \mu^{(1)}) \cdot P(x_{t+2} | \omega^{(2)}, \mu^{(2)})}{\sum_{x_{t+2}} P(x_{t+2}^{(0)} | \omega^{(0)}, \mu^{(0)}) P(x_{t+1} | \omega^{(1)}, \mu^{(1)})} \]

Approximations to PCCA Estimation

1. The belief state can be accurately approximated by maintaining the k most likely estimates
2. The probability of each state can be accurately approximated by the most likely trajectory to that state
3. The observation probability can be reduced to 1.0 for observations consistent with the state, and 0.0 for observations inconsistent with the state

Belief State Representation

- Best-First Belief State Enumeration (BFBSE)
- Best-First Trajectory Enumeration (BFTE)
**Simple IMU/PS Scenario**

- Best-First Belief State Enumeration (BFBSE)
- Best-First Trajectory Enumeration (BFTE)

**PS Automaton**

**PCCA Estimation as an OCSP**

**Definition 1** An OCSP \( \{y, f, C\} \) is a problem of the form \( \arg \max f(x) \text{ subject to } C(y) \), where \( x \subseteq y \) is a vector of decision variables, \( C(y) \) is a set of state constraints, and \( f(x) \) is a multi-attribute utility function.

For PCCA Estimation:

- \( x \) is the set of reachable target modes
- \( C(y) \) requires that the observations, modal constraints, and interconnections must be consistent
- \( f(x) \) is the estimate probability for state \( x \)

**Accuracy Results**

- EO-1 Model (12 components)
- 30 estimation cycles (nominal operations)

**Complexity Analysis**

Recall A* best case: \( n \cdot b \), worst case: \( b^n \)

- Best-First Trajectory Enumeration (BFTE)
  - \( n \) arithmetic computation
  - \( k \) OCSPs
- Best-First Belief State Enumeration (BFBSE)
  - \( n \cdot k \) arithmetic computations
  - 1 OCSP

<table>
<thead>
<tr>
<th>Space</th>
<th>Time</th>
<th>Space</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>BFTE</td>
<td>( k \cdot n )</td>
<td>( k \cdot n \cdot (n + C) )</td>
<td>( k \cdot b^n )</td>
</tr>
</tbody>
</table>
| BFSE  | \( n \cdot b \) | \( n \cdot b \cdot (n - k + C) \) | \( b^n \cdot (n - k + C) \)
Performance Results

- Heap Memory Usage

Performance Results

- Run-time (1.7 GHz Pentium M, 512MB RAM)

Current Work

- Extend BFBSE to use both HMM belief state update equations
  - Use observation probabilities within the conflict-directed search to avoid unlikely candidates
  - Done efficiently using a conditional probability table

(Published in S.M. Thesis and i-SAIRAS '05)

Backup Slides

Simple Two Switch Scenario

- Most likely trajectories (k=2)
  - Very reactive
  - Gross lower-bound
  - Extraneous computation
  - Multiple OCSP instances
  - Estimates generated and thrown away

Approximate Belief State Enumeration (k=2)

- Complete probability
  - Assuming approximate belief state is the true belief state
  - Tighter lower bound
- Single OCSP
  - Best-first order using A*
- Unfortunately, Not MPI
Belief State Update

- Complete probability distribution is calculated using the Hidden Markov Model Belief State Update equations.
- A Priori Probability:
  \[ P(x_t^n | x_{t-1}^{n-1}) = \frac{1}{Z_t} \sum_{s_{t-1}^{n-1}} P(s_{t-1}^{n-1}) P(x_t^n | s_{t-1}^{n-1}) P(x_{t-1}^{n-1}) \]

- Solved as single OCSP
- \( A^* \) Cost Function:
  \[ f(n) = P(x_t^n | x_{t-1}^{n-1}) = \max_{s_{t-1}^{n-1}} P(x_t^n | s_{t-1}^{n-1}) P(x_{t-1}^{n-1}) \]

Three Switch Enumeration Example

- Assume:
  - No commands
  - No observations
- Enumeration scheme same as most likely trajectories:
  - Expand tree by adding mode assignments
  - Only difference is the cost function

Why max in \( f(n) \)?

- Initial Approximate Belief State and Transition Probabilities
- \( A^* \) Cost Function

Tree Expansion

- When mode assignment next?
- Tree expansion
- Due to no MPI, all children of each mode must be placed on the queue before deciding which mode to expand next

Guaranteed Admissible

- \( f(Sw1=on, Sw2=on, Sw3=on) = 0.48 \) (on the queue)
- Node is in the queue and was dequeued and expanded

- Best trajectory found

Tree Expansion

- \( Sw1=on \) has the best cost so it is dequeued and its children are expanded

- Initial Approximate Belief State and Transition Probabilities

- \( A^* \) Cost Function

- Enumerated scheme same as most likely trajectories
- Expand tree by adding mode assignments
- Only difference is the cost function
Tree Expansion

- Sw1=off is now the next best and is dequeued and expanded
- \( f(Sw1=on, Sw2=on) = 0 \)
- \( f(Sw1=on, Sw2=bkn) = 0.343 \)

Initial Approximate Belief State and Transition Probabilities

A* Cost Function

Continue expanding to get more

Performance Impact

- Approximate belief state enumeration successfully framed as an OCSP with an admissible heuristic for A*
- Uses same OCSP scheme as the most likely trajectories algorithm
- Changed only by using a different cost function
- Conflicts can still be used to prune branches
- A* is worse case exponential in the depth of the tree but the tree depth will not change
- Only one OCSP solver necessary
- No redundant conflicts to be generated
- No extraneous estimates generated and thrown out
- MPI no longer holds
- Queue will contain a larger number of implicants

Tree Expansion

- \( f(Sw1=on, Sw2=bkn) = 0.343 \)
- \( f(Sw1=on, Sw2=on) = 0 \)

Tree Expansion

- \( f(Sw1=on, Sw2=bkn) = 0.343 \)
- \( f(Sw1=on, Sw2=on) = 0 \)

Tree Expansion

- \( f(Sw1=on, Sw2=on) = 0 \)
- \( f(Sw1=on, Sw2=bkn) = 0.343 \)

Tree Expansion

- \( f(Sw1=on, Sw2=bkn) = 0.343 \)
- \( f(Sw1=on, Sw2=on) = 0 \)

A* Cost Function

\[ f(X) = \frac{1}{2} \left( \sum_{i} P(x_i | a_i) + \sum_{j} \max P(x_j | a_j) \right) \]

Tree Expansion

- \( f(Sw1=on, Sw2=bkn) = 0.343 \)
- \( f(Sw1=on, Sw2=on) = 0 \)

Tree Expansion

- \( f(Sw1=on, Sw2=bkn) = 0.343 \)
- \( f(Sw1=on, Sw2=on) = 0 \)

Tree Expansion

- \( f(Sw1=on, Sw2=bkn) = 0.343 \)
- \( f(Sw1=on, Sw2=on) = 0 \)

Tree Expansion

- \( f(Sw1=on, Sw2=bkn) = 0.343 \)
- \( f(Sw1=on, Sw2=on) = 0 \)

A* Cost Function

\[ f(X) = \frac{1}{2} \left( \sum_{i} P(x_i | a_i) + \sum_{j} \max P(x_j | a_j) \right) \]

Tree Expansion

- \( f(Sw1=on, Sw2=bkn) = 0.343 \)
- \( f(Sw1=on, Sw2=on) = 0 \)

Tree Expansion

- \( f(Sw1=on, Sw2=bkn) = 0.343 \)
- \( f(Sw1=on, Sw2=on) = 0 \)

Tree Expansion

- \( f(Sw1=on, Sw2=bkn) = 0.343 \)
- \( f(Sw1=on, Sw2=on) = 0 \)

A* Cost Function

\[ f(X) = \frac{1}{2} \left( \sum_{i} P(x_i | a_i) + \sum_{j} \max P(x_j | a_j) \right) \]

Tree Expansion

- \( f(Sw1=on, Sw2=bkn) = 0.343 \)
- \( f(Sw1=on, Sw2=on) = 0 \)

Tree Expansion

- \( f(Sw1=on, Sw2=bkn) = 0.343 \)
- \( f(Sw1=on, Sw2=on) = 0 \)

A* Cost Function

\[ f(X) = \frac{1}{2} \left( \sum_{i} P(x_i | a_i) + \sum_{j} \max P(x_j | a_j) \right) \]

Tree Expansion

- \( f(Sw1=on, Sw2=bkn) = 0.343 \)
- \( f(Sw1=on, Sw2=on) = 0 \)

Tree Expansion

- \( f(Sw1=on, Sw2=bkn) = 0.343 \)
- \( f(Sw1=on, Sw2=on) = 0 \)

A* Cost Function

\[ f(X) = \frac{1}{2} \left( \sum_{i} P(x_i | a_i) + \sum_{j} \max P(x_j | a_j) \right) \]

Tree Expansion

- \( f(Sw1=on, Sw2=bkn) = 0.343 \)
- \( f(Sw1=on, Sw2=on) = 0 \)

Tree Expansion

- \( f(Sw1=on, Sw2=bkn) = 0.343 \)
- \( f(Sw1=on, Sw2=on) = 0 \)

A* Cost Function

\[ f(X) = \frac{1}{2} \left( \sum_{i} P(x_i | a_i) + \sum_{j} \max P(x_j | a_j) \right) \]

Tree Expansion

- \( f(Sw1=on, Sw2=bkn) = 0.343 \)
- \( f(Sw1=on, Sw2=on) = 0 \)

Tree Expansion

- \( f(Sw1=on, Sw2=bkn) = 0.343 \)
- \( f(Sw1=on, Sw2=on) = 0 \)

A* Cost Function

\[ f(X) = \frac{1}{2} \left( \sum_{i} P(x_i | a_i) + \sum_{j} \max P(x_j | a_j) \right) \]