Micro Air Vehicle Flight in GPS-Denied Environments: Planning in Information Space

Nicholas Roy
Friday, October 17th, 2008

Robust Robotics Group at MIT

Mission: Building smart learning and planning algorithms that allow robustness to real-world uncertainty in sensors, actuators and models
The Mission
Mission Vehicle

- Payload is GPS/IMU/compass/pressure sensor, electronics and microprocessor, datalink transmitter (900 MHz), camera and analog video transmitter (2.4 GHz).
- Onboard electronics provide attitude stabilization and GPS waypoint control using state estimation at 1000 Hz.
- Datalink, video antennas below frame, GPS/IMU/compass above frame.
- Camera field-of-view was 90°, and could be pitched from 0° (forward looking) to 90° (straight down).

Mission Vehicle

- 405 g, 29cm
- 6 rotors provide superior thrust-weight performance
- Top speed is 10m/s, maximum pitch angle of vehicle is 22°
- Maximum altitude during mission profile is 50m
- Flight time with batteries is 10-12 minutes
Actual Mission Profile

<table>
<thead>
<tr>
<th></th>
<th>Maximum height (m)</th>
<th>Distance traveled (m)</th>
<th>Total flight time (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual Mission</td>
<td>35.7</td>
<td>1759.2</td>
<td>710.0</td>
</tr>
<tr>
<td>Maximum height</td>
<td>13.9</td>
<td>1247.2</td>
<td>621.1</td>
</tr>
<tr>
<td>Distance traveled</td>
<td>28.8</td>
<td>1280.5</td>
<td>644.7</td>
</tr>
</tbody>
</table>

Mission Execution
Quadrotor Helicopter

AscTec Hummingbird
System Diameter – 53cm
Flight time with 200g payload – 12 min
Real-time, active control required

Hokuyo Laser Range-finder Sensor
Range – 4 m, Field-of-View – 240°
Update rate – 10 hz
Quadrotor Helicopter

AscTec Hummingbird
System Diameter – 53cm
Flight time with 200g payload – 12 min
Real-time, active control required

Hokuyo Laser Range-finder Sensor
Range – 4 m, Field-of-View – 240°
Update rate – 10 hz

Sensor Limitations and Indoor Flight

- Given:
  - Map of environment
  - Start, goal locations

- Plan path for **autonomous** helicopter navigation
  - Sensor limitations
- Maximize likelihood of reaching goal
  - Minimize uncertainty
Control Models

Planning Trajectories in GPS-denied Environments

- Standard shortest path:
  \[
  \min \sum_{t=0}^{T} C(s(t), a(t))
  \]
Planning Trajectories in GPS-denied Environments

• Standard shortest path:

\[
\min \sum_{t=0}^{T} C(s(t), a(t))
\]

Shortest-path solution has "dangerously" imprecise localization
Planning Trajectories in GPS-denied Environments

- Standard shortest path:
  \[ \min \sum_{t=0}^{T} C(s(t), a(t)) \]

- Information space planning:
  1. Minimize expected costs
  \[ \min E_{s(t)} \left[ \sum_{t=0}^{T} C(s(t), a(t)) \right] \]
  2. Compute a plan mapping distributions \( P(S) \) to actions

Motion Planning

Goal: shortest path, subject to kinematic and environmental constraints
Motion Planning in High Dimensional Configuration Spaces

1. Sample poses in $C_{\text{free}}$
2. Add edges between mutually-visible points
3. Perform graph search

Assumes a controller exists to transfer from $x_t$ to $x_{t+1}$

State vs. Information Space

- Large covariances can lead to poor plan execution
Many Existing Techniques

- LQR, $H_\infty$ control
  - Non-convexity of global planning problem creates problems
  - Unsuitable for global planning

- Partially Observable Markov Decision Processes
  - Discretization of state, action, observation space (or discretization of information space) leads to computational intractability
  - Unsuitable for high-dimensional spaces
Motion Planning in Information Space

1. Sample distributions where \( p(x \in C_{\text{obs}}) < \varepsilon \)
2. Add edges between points where \( p(x \in C_{\text{obs}}) < \varepsilon \) along path
3. Perform graph search

Problem: Edge Construction

- Need \( u_{0:T} \) such that \( p(x|u_{0:T}) = p(x') \)
- Possible solution: sample waypoints, use forward simulation to compute full posterior
Example Belief Roadmap

Problem: Edge Construction

- Need to perform forward simulation (and belief prediction) along each edge for every start state
- Computing minimum cost path of 30 edges: ≈100 seconds
- Not an issue for single queries: clearly a problem for multi-query planning
Multi-Step Update as One-Step

**EKF Covariance Update**

Control: $\bar{\Sigma}_t = G\Sigma_{t-1}G^T + R$

Measurement: $\Sigma_t = (\bar{\Sigma}_t^{-1} + HQ^{-1}H^T)^{-1}$

![Diagram of Multi-Step Updates and One-Step Update]

**Solution: Decomposition**

- Key idea: factor the covariance matrix
  $$\Sigma = BC^{-1}$$

- Motion update
  $$\bar{\Sigma}_t = \bar{E}_t\bar{D}_t^{-1}$$

  $$\begin{bmatrix} \bar{D}_t \\ \bar{E}_t \end{bmatrix} = \begin{bmatrix} 0 & G_t^{-T} \\ G_t & R_t G_t^{-T} \end{bmatrix} \begin{bmatrix} B_{t-1} \\ C_{t-1} \end{bmatrix}$$
Solution: Decomposition

- Key idea: factor the covariance matrix
  \[ \Sigma = BC^{-1} \]
- Measurement update
  \[ \Sigma_t = B_tC_t^{-1} \]
  \[
  \begin{bmatrix}
  B_t \\
  C_t
  \end{bmatrix} = \begin{bmatrix}
  0 & 1 \\
  1 & M_t
  \end{bmatrix}
  \begin{bmatrix}
  \bar{D}_t \\
  \bar{E}_t
  \end{bmatrix}
  \]

Solution: Decomposition

- One-step transfer function for the covariance:
  \[ \zeta_t = \begin{bmatrix}
  0 & I \\
  I & M_t
  \end{bmatrix}
  \begin{bmatrix}
  0 & G_t^{-T} \\
  G_t & R_t G_t^{-T}
  \end{bmatrix} \]
  \[ \Rightarrow \begin{bmatrix}
  B_T \\
  C_T
  \end{bmatrix} = \left( \prod_{t=0}^{T} \zeta_t \right)
  \begin{bmatrix}
  B_0 \\
  C_0
  \end{bmatrix} \]
- (To recover covariance, \( \Sigma = BC^{-1} \))
- This trick is not new.
  - Kailleth et al., Linear State Estimation.
The Belief Roadmap Algorithm

1. Sample means from $C_{free}$, build graph and transfer functions
2. Propagate covariances by performing graph search
Improving Sampling

Uniform Sampling

Sensor-Uncertainty Sampling
## Running Time

<table>
<thead>
<tr>
<th>Method</th>
<th>tr(Σ)</th>
<th>Build time</th>
<th>Search time</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRM</td>
<td>16.046</td>
<td>0.036</td>
<td>.001</td>
</tr>
<tr>
<td>BRM, Uniform Sampling</td>
<td>4.223</td>
<td>18.920</td>
<td>0.039</td>
</tr>
<tr>
<td>BRM, Sensor-Uncertainty Sampling</td>
<td>1.094</td>
<td>25.589</td>
<td>0.032</td>
</tr>
</tbody>
</table>

BRM in action (2x speed)
Limitations

- No way to couple high-level planning to sensing and model improvement
Planning in Incomplete Models

Planning problem: choose the actions given the current robot pose $s(t)$ and map $M(t)$ such that

$$\min E_{s,M(t)} \left[ \sum_{t=0}^{T} C(s(t), a(t), M(t)) \right]$$

- Problem:
  - No exact parametric model of $p(\text{map})$
  - Massively computationally intractable
Multistep Policies

• Why are n-step policies useful?

One step policy (iterated)

Two step policy

Reinforcement Learning

• Input:
  – Current and goal vehicle pose
  – Current map estimate

• Output
  – Trajectory that minimizes expected cost

• Learn actions that minimize expected cost in practice

• Core algorithm: stochastic function approximation

\[ f : \mathbb{R}^3 \times \mathbb{R}^n \times \mathbb{R}^3 \rightarrow \mathbb{R}^3 \]
Support Vector Machines

- Non-linear regression

\[ x = (x, M) \]
\[ f : x \rightarrow a \]
\[ h(x) = \text{sgn}\left( \sum_{j=1}^{N} a_j K\left(x_j \cdot x\right) + b \right) \]
\[ f(x) = \begin{cases} a, & \text{if } h(x) > 0 \\ a, & \text{otherwise} \end{cases} \]

where

\[ K(x, z) = \langle \phi(x) \cdot \phi(z) \rangle \]
and \( \phi(x) \) is some “feature” or component of \( x \).

Map Error Minimization in Stata

IEEE International Conference on Robotics and Automation 2006, T. Kollar
National Conference on Artificial Intelligence 2008, T. Kollar
Map Error Minimization in Stata

Previous state of the art: computing each exploration action takes 30 minutes
Learned controller: computing each exploration action takes milliseconds

Learning to Reduce Weather Forecast Error

- Want to use mobile sensors (e.g., UAVs) to gather additional atmospheric data to minimize forecast error
- Almost exactly the same problem as map exploration: find the trajectory that minimizes expected cost
- Can use almost exactly the same learning algorithms
LittleDog

Additional Applications

- Wheelchair Interaction at The Boston Home
- Sensor planning
- Mobile manipulation
- Plan control in time-constrained domains
Summary

• Robust, long-term autonomy in large-scale environments

• Planning algorithms for worlds in which we have limited knowledge of the state, model of the system, or a map of the world

• Technical approaches
  • Finding good representations of real-world problems
  • Using statistical machine learning techniques to solve the optimization problems