Improving the Performance of Sampling-Based Motion Planning with Symmetry-Based Gap Reduction

by: Peng Cheng, Emilio Frazzoli, and Steven LaValle

Phillip Coleman

Algorithms & Applications Group
Parasol Lab, Dept. of Computer Science,
Texas A&M University
Outline

- MPD Problem
- Motion Planning with Gap Reduction
- Simulations
- Conclusion
MPD Problem

- Motion Planning with Differential Constraints
  - Kinodynamic Planning
  - Nonholonomic Planning
- Differential Constraints
  - Velocity, Acceleration, Momentum. Moving Obstacles
MPD Problem

- **Difficulties**
  - Nonintegrable Constraints
  - Complex configuration space
  - Configuration can not be reached using any trajectory

- **Solutions**
  - Complete
    - Only a few systems
  - Two stage
    - Computationally Complex
  - Sampling-based
MPD Problem – Sampling Based

- **State Space X**
  - $X \subset \mathbb{R}^n$
  - Higher dimension than C Space
  - If $q$ is invalid in C also invalid in $X$

- **Input Space U**
  - $U \subset \mathbb{R}^m$ ($m \leq n$)
  - Controls that effect dynamic constraints
  - Motion equation
    - $x' = f(x,u)$
MPD Problem – Sampling Based

- **Solution**
  - Concatenation of Controls
    
    \[(\tilde{u}_1\tilde{u}_2)(t) = \begin{cases} 
    \tilde{u}_1(t) & t \in [0, \bar{t}(\tilde{u}_1)) \\
    \tilde{u}_2(t-t_1) & t \in [\bar{t}(\tilde{u}_1), \bar{t}(\tilde{u}_1) + \bar{t}(\tilde{u}_2)] 
    \end{cases}\]

  - \(\Phi(u,x,t) = x_0 + \int_0^t f(\tilde{x}(\tau), \tilde{u}(\tau)) d\tau\)

  - Exact Solution
    
    - If for all \(t\), \(\Phi(u,x,t)\) is valid
    - \(\Phi(u,x,t(u)) = x_{\text{goal}}\)
MPD Problem – Example

- A nonholonomic car
- Constant forward 60 mph
- State \((x, y, \theta, u_g, \omega)\)
MPD Problem – Example

- Input $u$ is steering angle
- Motion Equation

\[
\dot{x} = v_x \cos(\theta) - v_y \sin(\theta) \quad \dot{\theta} = \omega
\]

\[
\dot{y} = v_x \sin(\theta) + v_y \cos(\theta) \quad \dot{\omega} = \frac{(f_{yf}a - f_{yr}b)}{I}
\]

\[
\dot{v}_y = -v_x \omega + \frac{(f_{yf} + f_{yr})}{M}
\]
MPD Problem – Gaps

- Trajectories determined by inputs
- Not any two configurations can be connected
- Gaps can be formed
  - Near goal
  - Between controls
  - Gap Tolerance
  - Like Narrow Passage Problem
MPD Problem – Gaps

- **Sampling-Based Method**
  - Search graph initialized with one or more states
  - Generate a new trajectory from current state
  - Update Search Graph
  - Check to see if within gap tolerance
    - If so return concatenation of controls
  - Return to step 2 if no solution found
Outline

- MPD Problem
- Motion Planning with Gap Reduction
- Simulations
- Conclusion
Motion Planning with Gap Reduction

- Difficult to close gaps
- Find valid solution within large tolerance
- Perturb solution until within desired tolerance
- Constant time evaluation of perturbations

![Diagram showing motion planning with gap reduction](attachment:image.png)
Motion Planning with Gap Reduction

- Symmetries
  - $G$ is a Lie group
  - Any action in $G$ on the state is a smooth map
    - $\Psi: G \times X \to G$ is a smooth function
    - $\Psi(e, x) = x$
    - $\Psi(g, \Psi(h, x)) = \Psi(gh, x)$
  - Invarient with respect to the action of $G$
    - $\Psi \circ \Phi(u, x, t(u)) = \Phi(u, x, t(u)) \circ \Psi$
Motion Planning with Gap Reduction

- **Coasting Trajectories**
  - \( \Phi (u, x, t) = \Psi (\exp(Et), x) \)
    - \( E \) is an element of \( G \)
  - Group Displacement
    - Parameterized by time

\[ \Phi (u, x, t) = \Psi (\exp(Et), x) \]
\[ E \in G \]

Parameterized by time
Motion Planning with Gap Reduction

- Gap Reduction using Symmetries
  - Efficient Final State Evaluation
    - Constant Time
    - Group Displacement
      - \( \Psi (x_f) \)
      - Without reintegrating \( u \)
    - Insert multiple coasting trajectories
      - Perturb duration

Motion Planning with Gap Reduction

\[ x'_f = \Phi_{u_4}^{t_4} \circ \Phi_{u_3}^{\delta t_3} \circ \Phi_{u_2}^{t_3} \circ \Phi_{u_2}^{\delta t_2} \circ \Phi_{u_2}^{t_2} \circ \Phi_{u_1}^{\delta t_1} \circ \Phi_{u_1}^{t_1}(x_{\text{init}}) \]

\[ = \Phi_{u_4}^{t_4} \circ \Psi_{h_3} \circ \Phi_{u_3}^{t_3} \circ \Psi_{h_2} \circ \Phi_{u_2}^{t_2} \circ \Psi_{h_1} \circ \Phi_{u_1}^{t_1}(x_{\text{init}}) \]

\[ = \Psi_{h_3} \circ \Psi_{h_2} \circ \Psi_{h_1}(x_f) \]
Motion Planning with Gap Reduction

- **Heuristic** for optimal coasting trajectories
  - Perturb trajectories iteratively
  - Only a subset effected each time step
  - Gradient Descent
    - Convergence rate

\[
\alpha^2 = 1 - \frac{\left(\sum_i \xi_i^2 \lambda_i^2\right)^2}{\left(\sum_i \xi_i^2 \lambda_i^3\right)\left(\sum_i \xi_i^2 \lambda_i\right)}
\]
Motion Planning with Gap Reduction

- Heuristic for optimal coasting trajectories
  - Three coasting trajectories
  - Perturb two vectors
Motion Planning with Gap Reduction

- Incorporating Gap Reduction
  1. **Detect** solution candidate
     1. Within a larger gap tolerance
  2. **Make** two gap states differ by a group action
     1. Preprocessing Method to reduce gap
  3. **Eliminate** gap
     1. Iterative perturbations
     2. Finish if gap is within final tolerance
Outline

- MPD Problem
- Motion Planning with Gap Reduction
- Simulations
- Conclusion
Simulations

- Multiple planners
  - Unidirectional
  - Bidirectional
  - PRM
- 20 trials per planner/problem combination
- Terminates if no solution within 400000 iter.
- Gap tolerance – 100 – 0.1
Simulations

- Two Systems
  - Nonholonomic Car and Trailer
  - Finite Friction model Roller Racer
Simulations

- Computational Cost for basic Sampling

**RUNNING TIME WITH DIFFERENT GAP TOLERANCES**

<table>
<thead>
<tr>
<th>$\epsilon_g$</th>
<th>Max. Time</th>
<th>Min. Time</th>
<th>Avg. Time</th>
<th>Succ.</th>
</tr>
</thead>
<tbody>
<tr>
<td>100.0</td>
<td>158.7</td>
<td>2.6</td>
<td>29.7105</td>
<td>20</td>
</tr>
<tr>
<td>10.0</td>
<td>7113.7</td>
<td>2.6</td>
<td>1992.7</td>
<td>20</td>
</tr>
<tr>
<td>1.0</td>
<td>60736.2</td>
<td>2075.8</td>
<td>30576.3</td>
<td>11</td>
</tr>
<tr>
<td>0.1</td>
<td>61785.7</td>
<td>60168.1</td>
<td>60736.2</td>
<td>0</td>
</tr>
</tbody>
</table>
Simulations

- Comparison with Classic Numerical Method

<table>
<thead>
<tr>
<th></th>
<th>Ob.</th>
<th>Sy.</th>
<th>Pl.</th>
<th>Sel.</th>
<th>$T_{all}$</th>
<th>$N_f$</th>
<th>Succ.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car</td>
<td>N</td>
<td>Y</td>
<td>uni.</td>
<td>Y</td>
<td>7.5e2</td>
<td>8.2e7</td>
<td>20</td>
</tr>
<tr>
<td>Car</td>
<td>N</td>
<td>N</td>
<td>uni.</td>
<td>Y</td>
<td>2.3e4</td>
<td>4.5e9</td>
<td>20</td>
</tr>
<tr>
<td>Car</td>
<td>Y</td>
<td>Y</td>
<td>uni.</td>
<td>Y</td>
<td>3.4e4</td>
<td>7.9e7</td>
<td>20</td>
</tr>
<tr>
<td>Car</td>
<td>Y</td>
<td>N</td>
<td>uni.</td>
<td>Y</td>
<td>9.1e4</td>
<td>9.5e9</td>
<td>20</td>
</tr>
<tr>
<td>Trailer</td>
<td>N</td>
<td>Y</td>
<td>uni.</td>
<td>Y</td>
<td>2.2e2</td>
<td>9.4e6</td>
<td>20</td>
</tr>
<tr>
<td>Trailer</td>
<td>N</td>
<td>N</td>
<td>uni.</td>
<td>Y</td>
<td>2.4e5</td>
<td>4.7e10</td>
<td>20</td>
</tr>
<tr>
<td>Trailer</td>
<td>Y</td>
<td>Y</td>
<td>uni.</td>
<td>Y</td>
<td>3.3e2</td>
<td>1.1e7</td>
<td>20</td>
</tr>
<tr>
<td>Trailer</td>
<td>Y</td>
<td>N</td>
<td>uni.</td>
<td>Y</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Simulations

- Effects of selecting coasting trajectories

<table>
<thead>
<tr>
<th></th>
<th>Ob.</th>
<th>Sy.</th>
<th>Pl.</th>
<th>Sel.</th>
<th>$T_{all}$</th>
<th>$N_f$</th>
<th>Succ.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car</td>
<td>N</td>
<td>Y</td>
<td>uni.</td>
<td>Y</td>
<td>2.4e3</td>
<td>1.4e3</td>
<td>20</td>
</tr>
<tr>
<td>Car</td>
<td>N</td>
<td>Y</td>
<td>uni.</td>
<td>N</td>
<td>3.7e3</td>
<td>5.3e3</td>
<td>20</td>
</tr>
<tr>
<td>Trailer</td>
<td>N</td>
<td>Y</td>
<td>uni.</td>
<td>Y</td>
<td>4.6e2</td>
<td>3.2e3</td>
<td>20</td>
</tr>
<tr>
<td>Trailer</td>
<td>N</td>
<td>Y</td>
<td>uni.</td>
<td>N</td>
<td>6.1e2</td>
<td>1.3e4</td>
<td>20</td>
</tr>
</tbody>
</table>
Simulations

- Effects of selecting coasting trajectories
Conclusions

- Motion Planning under Dynamic Constraints
- Symmetry Based-Gap Reduction
  - Improved performance
  - Efficient evaluation
- Problems
  - Relies on coasting trajectories, symmetry