Toward More Efficient Motion Planning with Differential Constraints

Maciej Kalisiak

Final Oral Exam
December 14th, 2007
Outline

1. Motion Planning (MP)
   - What is MP?
   - Types of MP problems
   - MP is hard

2. Viability

3. Contributions
   - MP in highly constrained problems
   - MP w/viability filtering
   - Viability-based safety enforcement

4. Conclusion
Outline

1. Motion Planning (MP)
   - What is MP?
   - Types of MP problems
   - MP is hard

2. Viability

3. Contributions
   - MP in highly constrained problems
   - MP w/viability filtering
   - Viability-based safety enforcement

4. Conclusion
What is Motion Planning (MP)?

- in a nutshell: 
  "how to get from A to B?"

- sometimes also: 
  "... optimally?"

- example problems
in a nutshell:

“how to get from A to B?”

sometimes also:

“... optimally?”

example problems
What is Motion Planning (MP)?

- in a nutshell:
  "how to get from A to B?"
- sometimes also:
  "... optimally?"
- example problems
What is Motion Planning (MP)?

- in a nutshell: “how to get from A to B?”
- sometimes also: “... optimally?”
- example problems
What is Motion Planning (MP)?

- in a nutshell: “how to get from A to B?”
- sometimes also: “... optimally?”
- example problems
Approach

- solved by converting to dual problem (agent → point)
- complication: often cannot manipulate agent directly

\[ x = (x, y, \theta) \]
\[ \mathcal{X} = x \times y \times \theta \]
\[ x \in \mathcal{X} \]
Approach

- solved by converting to dual problem (agent $\rightarrow$ point)
- complication: often cannot manipulate agent directly
Approach

- solved by converting to dual problem (agent → point)
- complication: often cannot manipulate agent directly
Types of MP problems

- **kinematic**
- **nonholonomic**
- **kinodynamic**

E.g., “Piano Mover’s Problem”
Types of MP problems

common types:
- kinematic
- nonholonomic
- kinodynamic

e.g., agents w/rolling contacts
Types of MP problems

common types:
- kinematic
- nonholonomic
- kinodynamic

e.g., inertia & balance play big role
Types of MP problems

common types:
- kinematic
- nonholonomic
- kinodynamic

Differential Constraints (DC)
- DC: constraints on $q'$
  $(\frac{d}{dt}$ of agent configuration)
- DCs very common, but make MP more difficult

Maciej Kalisiak
Toward More Efficient MP w/Differential Constraints
MP is hard

hardness

- Piano Mover’s Problem: \( \rightarrow \) PSPACE-complete
- MP problems w/DC: at least as hard

why?

- “curse of dimensionality”
- real world problems often high-D
- DCs complicate search space further
MP is hard

hardness

- Piano Mover’s Problem: → PSPACE-complete
- MP problems w/DC: at least as hard

why?

- “curse of dimensionality”
- real world problems often high-D
- DCs complicate search space further
Outline

1. Motion Planning (MP)
   - What is MP?
   - Types of MP problems
   - MP is hard

2. Viability

3. Contributions
   - MP in highly constrained problems
   - MP w/viability filtering
   - Viability-based safety enforcement

4. Conclusion
What is Viability?

"definition"

- **viable** state: $\exists$ an evasive action
- nonviable state: constraint violation unavoidable
What is Viability?

“definition”

- viable state: ∃ an evasive action
- nonviable state: constraint violation unavoidable
What is Viability?

“definition”

- viable state: $\exists$ an evasive action
- nonviable state: constraint violation unavoidable

why of interest?

- crops up in many contexts, useful
- exploited throughout thesis:
  - to expedite MP
  - to aid in user-control
Outline

1 Motion Planning (MP)
   - What is MP?
   - Types of MP problems
   - MP is hard

2 Viability

3 Contributions
   - MP in highly constrained problems
   - MP w/viability filtering
   - Viability-based safety enforcement

4 Conclusion
Overall goal of thesis

- **aim:** explore some novel ideas in MP
- **focus:** improving MP speed
- **grand vision:** MP with motion “macro-primitives”
Outline

1. Motion Planning (MP)
   - What is MP?
   - Types of MP problems
   - MP is hard

2. Viability

3. Contributions
   - MP in highly constrained problems
   - MP w/viability filtering
   - Viability-based safety enforcement

4. Conclusion
key idea: “any progress” is better than “no progress”
MP in highly constrained problems

- improvement to **RRT** algorithm
- highly-constrained problems: poor performance
- proposed: **RRT-Blossom**
- result: big speed ups (>10x)
MP in highly constrained problems

- improvement to RRT algorithm
- highly-constrained problems: poor performance
  - proposed: RRT-Blossom
  - result: big speed ups (>10x)
MP in highly constrained problems

- improvement to RRT algorithm
- highly-constrained problems: poor performance
- proposed: RRT-Blossom
- result: big speed ups (>10x)
RRT operation review

- grows two trees (from $q_{init}$ and $q_{goal}$)
- each tree grows toward $q_{tgt}$
RRT operation review

- grows two trees (from \( q_{\text{init}} \) and \( q_{\text{goal}} \))
- each tree grows toward \( q_{\text{tgt}} \)
RRT operation review

- grows two trees (from $q_{init}$ and $q_{goal}$)
- each tree grows toward $q_{tgt}$
RRT operation review

- grows two trees (from $q_{init}$ and $q_{goal}$)
- each tree grows toward $q_{tgt}$
RRT-Blossom

- allow *receding* edges...
- but not if *regressing*
- filter with regression test
- bottlenecks

**key idea:** “*any progress*” is better than “*no progress*”
RRT-Blossom

- allow **receding** edges...
- but not if **regressing**
- filter with regression test
- **bottlenecks**

### Key Idea

**any progress** is better than **no progress**

Regression if:

$$\exists \text{other} \mid \rho(\text{parent}, \text{leaf}) > \rho(\text{other}, \text{leaf})$$
**RRT-Blossom**

- allow **receding** edges...
- but not if **regressing**
- filter with regression test
- bottlenecks

**key idea:** “*any progress*” is better than “*no progress*”
Outline

1. Motion Planning (MP)
   - What is MP?
   - Types of MP problems
   - MP is hard

2. Viability

3. Contributions
   - MP in highly constrained problems
   - MP w/viability filtering
   - Viability-based safety enforcement

4. Conclusion
MP w/viability filtering

drawbacks of tree-based MP:
- tactile-only sensing
- search ignores prior attempts

general idea:
- “work smarter, not harder”
- add “sight” + “learning” → faster MP
drawbacks of tree-based MP:
- tactile-only sensing
- search ignores prior attempts

general idea:
- “work smarter, not harder”
- add “sight” + “learning” → faster MP
Key extensions

“sight”
- virtual sensors: distance along path
- yield “locally situated” state

“learning”
- prior trajectories $\rightarrow$ viability models
- models parametrized using sensors
  $\rightarrow$ local models
  $\rightarrow$ transferrable
- ideally: bootstrapping
Key extensions

“sight”
- virtual sensors: distance along path
- yield “locally situated” state

“learning”
- prior trajectories $\rightarrow$ viability models
- models parametrized using sensors
  $\rightarrow$ local models
  $\rightarrow$ transferrable
- ideally: bootstrapping
Exploiting viability

observations

- currently: search in all of $X_{\text{free}}$
- but $X_{\text{free}}$ includes $X_{\text{ric}}$
- $x_{\text{goal}}$ usually unreachable from $x \in X_{\text{ric}}$

Maciej Kalisiak  
Toward More Efficient MP w/Differential Constraints
Exploiting viability

observations

- currently: search in all of $\mathcal{X}_{free}$
- but $\mathcal{X}_{free}$ includes $\mathcal{X}_{ric}$
- $x_{goal}$ usually unreachable from $x \in \mathcal{X}_{ric}$
Exploiting viability

**observations**

- currently: search in all of $X_{\text{free}}$
- but $X_{\text{free}}$ includes $X_{\text{ric}}$
- $x_{\text{goal}}$ usually unreachable from $x \in X_{\text{ric}}$
Exploiting viability

**Observations**
- currently: search in all of $\mathcal{X}_{free}$
- but $\mathcal{X}_{free}$ includes $\mathcal{X}_{ric}$
- $x_{goal}$ usually unreachable from $x \in \mathcal{X}_{ric}$

$\Rightarrow$ **Avoid futile searching!**
- model agent viability
- keep MP search within $\text{Viab}(\mathcal{X}_{free})$
- observed: speed-up of up to 10x
Results: model transfer

agent

problem posed

model trained on

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>RRT-CT</th>
<th>RRT-Blossom</th>
<th>RRT-Blossom/VF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem posed</td>
<td>371.5s</td>
<td>21.0s</td>
<td>5.6s</td>
</tr>
<tr>
<td>Model trained on</td>
<td>209.9s</td>
<td>13.5s</td>
<td>3.6s</td>
</tr>
<tr>
<td></td>
<td>2148.6s</td>
<td>305.7s</td>
<td>34.3s</td>
</tr>
</tbody>
</table>
Results: model transfer

agent

problem posed

model trained on

<table>
<thead>
<tr>
<th>problem</th>
<th>algo.</th>
<th>runtimes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RRT-CT</td>
<td>371.5s</td>
</tr>
<tr>
<td></td>
<td>RRT-Blossom</td>
<td>21.0s</td>
</tr>
<tr>
<td></td>
<td>RRT-Blossom-VF</td>
<td>5.6s</td>
</tr>
<tr>
<td></td>
<td>RRT-CT</td>
<td>209.9s</td>
</tr>
<tr>
<td></td>
<td>RRT-Blossom</td>
<td>13.5s</td>
</tr>
<tr>
<td></td>
<td>RRT-Blossom-VF</td>
<td>3.6s</td>
</tr>
<tr>
<td></td>
<td>RRT-CT</td>
<td>2148.6s</td>
</tr>
<tr>
<td></td>
<td>RRT-Blossom</td>
<td>305.7s</td>
</tr>
<tr>
<td></td>
<td>RRT-Blossom-VF</td>
<td>34.3s</td>
</tr>
</tbody>
</table>
Results: tree structure

- RRT-CT
- RRT-Blossom
- RRT-Blossom w/VF
  - no filtering
  - viability filtering
Outline

1. Motion Planning (MP)
   - What is MP?
   - Types of MP problems
   - MP is hard

2. Viability

3. Contributions
   - MP in highly constrained problems
   - MP w/viability filtering
   - Viability-based safety enforcement

4. Conclusion
Viability-based safety enforcement

- Assisted control:
  - Inherently useful
  - Facilitates obtaining user-demonstrated training data
  - Helpful in user-assisted MP (future work)

- **Key idea:** Viability more reliable for detecting imminent danger
Viability-based safety enforcement

- assisted control:
  - inherently useful
  - facilitates obtaining user-demonstrated training data
  - helpful in user-assisted MP (future work)

- **key idea:** viability more reliable for detecting imminent danger
Collision avoidance

**typical (collision-based)**
- based on predictive lookahead ($T_h$ seconds)
- weakness: $T_h$ is finite
  - $T_h$ may be too small
  - safety $\uparrow$ as $T_h \to \infty$

**better: viability-based safety enforcement**
- only a minimal lookahead needed
- longer lookaheads: milder corrections
Collision avoidance

typical (collision-based)
- based on predictive lookahead ($T_h$ seconds)
- weakness: $T_h$ is finite
  - $T_h$ may be too small
  - safety↑ as $T_h \to \infty$

better: viability-based safety enforcement
- only a minimal lookahead needed
- longer lookaheads: milder corrections
Collision avoidance

typical (collision-based)
- based on predictive lookahead (\( T_h \) seconds)
- weakness: \( T_h \) is finite
  - \( T_h \) may be too small
  - safety \( \uparrow \) as \( T_h \rightarrow \infty \)

better: viability-based safety enforcement
- only a minimal lookahead needed
- longer lookaheads: milder corrections
Collision avoidance

typical (collision-based)
- based on predictive lookahead ($T_h$ seconds)
- weakness: $T_h$ is finite
  - $T_h$ may be too small
  - safety $\uparrow$ as $T_h \to \infty$

better: viability-based safety enforcement
- only a minimal lookahead needed
- longer lookaheads: milder corrections
Collision avoidance

typical (collision-based)
- based on predictive lookahead ($T_h$ seconds)
- weakness: $T_h$ is finite
  - $T_h$ may be too small
  - safety ↑ as $T_h \to \infty$

better: viability-based safety enforcement
- only a minimal lookahead needed
- longer lookaheads: milder corrections
Collision avoidance

**typical (collision-based)**
- based on predictive lookahead ($T_h$ seconds)
- weakness: $T_h$ is finite
  - $T_h$ may be too small
  - safety $\uparrow$ as $T_h \to \infty$

**better: viability-based safety enforcement**
- only a minimal lookahead needed
- longer lookaheads: milder corrections
Operation

$F^i(x_k, v_k)$

$F^i(x_k, \hat{u}_j)$

$\mathcal{L}_0$

$\mathcal{L}_1$

$\mathcal{L}_2$

$\mathcal{L}_3$
Viability of control actions

\[ \hat{U} \]

\[ T_h \quad T_{eb} \]

Maciej Kalisiak
Toward More Efficient MP w/Differential Constraints
Experiments

agents

environments
Results

Viab(\(X_{\text{free}}\)) model

environment

enforcement
Conclusion

**contributions**
- better handling of constrained environments in RRT
- more efficient MP by narrowing search to $\text{Viab}(\mathcal{X}_{\text{free}})$
- more robust threat avoidance in computer-assisted control

**future work:**
- learning appropriate *actions* from motion data
- MP w/motion “macro primitives”
- evaluate viability filtering with other MPs
- *local* viability models for safety enforcement
- (near-)optimal solutions for MP w/DC
- human-derived motion data (e.g., style content)
- human-guided MP: selection of style or topology
Conclusion

- **contributions**
  - better handling of constrained environments in RRT
  - more efficient MP by narrowing search to $\text{Viab}(X_{\text{free}})$
  - more robust threat avoidance in computer-assisted control

- **future work:**
  - learning appropriate *actions* from motion data
  - MP w/motion “macro primitives”
  - evaluate viability filtering with other MPs
  - *local* viability models for safety enforcement
  - (near-)optimal solutions for MP w/DC
  - human-derived motion data (e.g., style content)
  - human-guided MP: selection of style or topology