Path Planning for Swarms by Combining Probabilistic Roadmaps and Potential Fields

Alex Wallar

Dept. of Electrical Engineering and Computer Science
Catholic University of America

http://faculty.cua.edu/plaku

Motion Planning

Proposed approach automatically plans the necessary motions that enable the robot to reach a desired destination while avoiding collisions with obstacles

[movie]

Motivated by applications in

- Navigation
- Exploration
- Search-and-Rescue

Computational Challenges

Motion planning requires searching a vast high-dimensional state space for *dynamically-feasible motions* that avoid collisions

Taking motion dynamics into account is essential to ensure that the planned motions can be followed in the physical world

- Motion dynamics can be complex
- Often involve non-linear ODEs

Computational Challenges

[movie]

- Snake-like robot model consists of several links attached to each other
  Continuous state consists of $s = (x, y, \theta_0, v, \psi, \theta_1, \theta_2, \ldots, \theta_N)$
- Motion dynamics modeled as a car pulling trailers
  The differential equations of motions are
  $$\dot{x} = v\cos(\theta_0) \quad \dot{y} = v\sin(\theta_0) \quad \dot{\theta}_0 = v\tan(\psi) \quad \dot{v} = a \quad \dot{\psi} = \omega$$
  $$\dot{\theta}_i = \frac{v}{d}\left(\prod_{j=1}^{i-1}\cos(\theta_{j-1} - \theta_j)\right)(\sin(\theta_{i-1}) - \sin(\theta))$$
  Robot controls: a (acceleration); ω (rotational velocity of steering wheel)

Related Work

Sampling-Based Motion Planning

Expand a tree $\mathcal{T}$ of collision-free and dynamically-feasible motions

- select a state s from which to expand the tree
- sample control input u
- generate new trajectory by applying u to s

Successful Motion Planners

- RRT [LaValle, Kuffner: IJRR 2001]
- EST [Hsu et al: IJRR 2002]
- PDST [Ladd, Kavraki: RSS 2005]
- SYCLOP [Plaku, Kavraki, Vardi: IEEE TRO 2010]

Related Work

Sampling-Based Motion Planning

Expand a tree $\mathcal{T}$ of collision-free and dynamically-feasible motions

- select a state s from which to expand the tree
- sample control input u
- generate new trajectory by applying u to s

Successful Motion Planners

- RRT [LaValle, Kuffner: IJRR 2001]
- EST [Hsu et al: IJRR 2002]
- PDST [Ladd, Kavraki: RSS 2005]
- SYCLOP [Plaku, Kavraki, Vardi: IEEE TRO 2010]

Related Work

Sampling-Based Motion Planning

Expand a tree $\mathcal{T}$ of collision-free and dynamically-feasible motions

- select a state s from which to expand the tree
- sample control input u
- generate new trajectory by applying u to s

Successful Motion Planners

- RRT [LaValle, Kuffner: IJRR 2001]
- EST [Hsu et al: IJRR 2002]
- PDST [Ladd, Kavraki: RSS 2005]
- SYCLOP [Plaku, Kavraki, Vardi: IEEE TRO 2010]

Related Work

Sampling-Based Motion Planning

Expand a tree $\mathcal{T}$ of collision-free and dynamically-feasible motions

- select a state s from which to expand the tree
- sample control input u
- generate new trajectory by applying u to s

Successful Motion Planners

- RRT [LaValle, Kuffner: IJRR 2001]
- EST [Hsu et al: IJRR 2002]
- PDST [Ladd, Kavraki: RSS 2005]
- SYCLOP [Plaku, Kavraki, Vardi: IEEE TRO 2010]

Related Work

Sampling-Based Motion Planning

Expand a tree $\mathcal{T}$ of collision-free and dynamically-feasible motions

- select a state s from which to expand the tree
- sample control input u
- generate new trajectory by applying u to s

Successful Motion Planners

- RRT [LaValle, Kuffner: IJRR 2001]
- EST [Hsu et al: IJRR 2002]
- PDST [Ladd, Kavraki: RSS 2005]
- SYCLOP [Plaku, Kavraki, Vardi: IEEE TRO 2010]

Related Work

On difficult kinodynamic motion-planning problems it has been noted that

- Exploration frequently gets stuck
- Progress slows down

- Exploration guided by limited information, such as distance metrics and nearest neighbors
- Lack of global sense of direction toward goal
- Difficult to discover new promising directions toward goal

[Donald et al., 93; Kim et al., 05; Plaku et al., 05; Yershova et al., 05; Ladd, Kavraki 05; Choset et al., 05; LaValle 06; Hsu et al., 06; Burns, Brock 07; Plaku et al. '10; Sucan, Kavraki '11]

New Approach

- Treats motion planning with dynamics as a search problem in a *hybrid* state space composed of continuous and discrete components
- Couples sampling-based motion planning with discrete search

discrete layer: guide motion planning
continuous layer: expand tree of feasible motions
interplay: update guide to reflect motion-planning progress

- Significantly improves computational efficiency over related work

Workspace Decomposition

Workspace decomposition provides discrete layer as adjacency graph

$$G = (R, E)$$

- R denotes the regions of the decomposition
- $E = \{(r_i, r_j) : r_i, r_j \in R \text{ are physically adjacent}\}$

Backward Discrete Search to Estimate Region Costs

Workspace decomposition provides discrete layer as adjacency graph

$$G = (R, E)$$

- R denotes the regions of the decomposition
- $E = \{(r_i, r_j) : r_i, r_j \in R$ are physically adjacent$\}$

cost(r) estimates the difficulty of reaching the goal region from r

defined as length of shortest path in $G = (R, E)$ from r to goal

$[cost(r_1), cost(r_2), \ldots, cost(r_n)]$

computed by running once A* shortest-path algorithm backwards from goal

Region Selection based on Cost Estimates

At each iteration, a region r_{from} is selected from which to expand the search tree T

Tree vertices are grouped according to the regions that have been reached

$$\text{vertices}(T, r) = \{v : v \in T \wedge \text{region}(v) = r\}$$

$$\Gamma = \{r : r \in R \wedge |\text{vertices}(T, r)| > 0\}$$

r_{from} is selected from Γ according to the probability distribution

$$prob(r) = \frac{1}{1 + cost^2(r)} / \sum_{r' \in R} \frac{1}{1 + cost^2(r')}$$

selection scheme balances greedy with methodical

Forward Discrete Search to Compute Discrete Plans

Discrete plan σ consists of a sequence of regions connecting r_{from} to goal

Exploitation (with probability p)

- σ computed as the shortest-path in $G = (R, E)$ from r_{from} to goal

Exploration (with probability $1 - p$)

- σ computed as a random path in $G = (R, E)$ from r_{from} to goal

Sampling-based Motion Planning to Expand the Search Tree

Expand tree of motions using the discrete plan as a guide

- Select vertices from regions associated with the discrete plan
- Sample input controls
- Generate new trajectories toward next region in the discrete plan
- Allow for deviations from the discrete plan

Interplay: Update Discrete Plan to Reflect Motion-Planning Progress

Experimental Results

[movie]

Compare computational efficiency to successful sampling-based motion planners

- RRT [LaValle, Kuffner]
- Syclop [Plaku, Kavraki, Vardi]

running time obtained as average of 30 different runs

Experimental Results (cont.)

[movie]

Compare computational efficiency to successful sampling-based motion planners

- RRT [LaValle, Kuffner]
- Syclop [Plaku, Kavraki, Vardi]

running time obtained as average of 30 different runs

Summary

Approach

- Treats motion planning with dynamics as a search problem in a *hybrid* state space composed of continuous and discrete components
- Couples sampling-based motion planning with forward and backward discrete search over a workspace decomposition
- Obtains significant computational speedups