Robotics 2
Kinodynamic Motion Planning for a Holonomic Robot

Christoph Sprunk,
Giorgio Grisetti, Cyrill Stachniss,
Kai Arras, Wolfram Burgard
Context

- Get robot *nicely* from start to goal
Nicely means
- smooth
- predictable
- as fast as possible
- constraint respecting
  - velocities
  - accelerations
  - centrifugal forces
  - ...

Context
Overview

- Motivation
  - Autonomous Navigation
  - Related Work
- System
  - Overview
  - Trajectory Representation
  - Velocity Profiles
  - Optimization
- Experiments
  - Setup
  - Travel Tasks
- Conclusion
Motivation

Navigation for holonomic robots so far
- manual: fork lifts, hydraulic ramps, ...
- autonomous
  - mostly low level control
  - kinematic analysis

- omniRob
- holonomic platform
- goal: autonomous navigation
omniRob - Holonomic Motion
Autonomous Navigation

High level task description
- move robot from start to goal (map provided)

Kinodynamic motion planning
- opposed to reactive systems
- control *how* the robot reaches the goal
- respect kinodynamic constraints
- predictability of behavior
- consider real costs
Related Work

Search based approaches

- stitch trajectories from motion primitives
- not at the same time
  - curvature continuity with re-planning
  - globally planned velocities
- curse of dimensionality
- holonomic systems: additional dimensions

☐ Extend optimization approach for differential/synchro drive
System Overview

- smooth trajectory defines position, velocity over time
- (anytime) optimization wrt cost function

- separation of trajectory generation and execution
  - small FB loop
  - abstraction
  - control theory
Initial Path

- sequence of turns on spot and translations
- collision free
- waypoint providing algorithm
  - A*
  - piano mover
  - Voronoi graph based
  - . . .
Quintic Bézier Spline

(a) 2d projection
(b) 3d view

Quintic Bezier Segment

- stitched from quintic polynomials
  - for $x, y, \Theta$ each
- curvature continuity
- localism
- free access to 1st and 2nd derivative at start/end
Path Modeling

Requirements
- closely fit shape of initial path (collision free)
- represent smooth paths
- continuous change of shape by change of parameters

Modeling
- waypoints $\rightarrow$ segment start/end points
- inner waypoint derivatives with heuristics
- turn on the spot $\leftrightarrow$ simultaneous translation and rotation
  - $\Theta$-entry/$\Theta$-exit points
Path Modeling – Θ-entry/Θ-exit

- insertion of Θ-entry/Θ-exit points on the 2d path
- subdivision of segments, 2d shape unchanged
- rotation at $W_i$ distributed between $W_i^{\text{entry}}$ and $W_i^{\text{exit}}$
- constant orientation between $W_i^{\text{exit}}$ and $W_{i+1}^{\text{entry}}$
- change of orientational behavior through movement of Θ-entry/Θ-exit points
Velocity Profile

- path + velocity profile → trajectory
- enables predictability
- fastest traversal of path
- respecting constraints
  - maximum velocities (translational, rotational)
  - obstacle imposed speed limit (safety)
  - maximum wheel velocities
  - maximum centrifugal acceleration
  - maximum accelerations (x,y,Θ)
  - payload based accelerational constraints (not platform limits)
- discretization depending on translation, rotation
Optimization

- optimize path shape wrt cost function
  - time of travel
  - energy efficiency
  - steering effort
  - ...

- parameters
  - waypoint 2D location
  - waypoint 2D tangent elongation
  - waypoint tangent, orientational component
  - \( \Theta \)-entry/\( \Theta \)-exit point location
  - orientational movement distribution (combined dimension for orientation at waypoints, entry/exit-points)
Optimization Algorithm

- RPROP inspired (Resilient backPROPagation)
  - derivative free
  - robust convergence

- while planning time left
  - optimize parameters independently
  - continue with next parameter as soon as better trajectory has been found

- cost function: cost = t_{travel}
Optimization – Alternative Costs

- energy efficiency depends on direction of travel relative to robot’s orientation

- exemplary penalty function for non-forward travel

\[
F = \frac{1}{\text{arc length}} \int_{0}^{\text{arc length}} 1 - |\cos \gamma(s)| \, ds
\]

\[
\text{cost} = t_{\text{travel}} + \alpha \cdot F \cdot t_{\text{travel}}
\]
Optimization – Examples

F=0, $t_{\text{travel}} = 39.15$ s.

Initial Path
Optimization – Examples

\[ F = 0.519, \ t_{\text{travel}} = 14.46 \text{ s.} \]

Optimized: time of travel
Optimization – Examples

F=0.067, $t_{\text{travel}}=16.28$ s.

Optimized: time of travel, energy efficiency
Setup – Hardware

(a) omniRob
(b) omniMove wheel

The holonomic robot omniRob: technology demonstrator by research and predevelopment department of KUKA Roboter GmbH, Augsburg.
Setup – Software

- CARMEN robot navigation toolkit
- value iteration based path planner (2d)
- Monte Carlo localization
Travel Tasks

- medium range travel
- map area approximately 11.2 m x 9.4 m

- short distance reorientation
- resemble repetitive pick & place task at high frequency

- tasks executed with different constraint parameter sets
Medium Range Travel
Short Distance Reorientation
Travel Tasks – Tracking Error

Translational tracking error

Graph showing translational error [m] for different scenarios:
- odo
- global
- goal error

Legend:
- medium range, param. set 1
- medium range, higher vels
- short distance, param. set 1
- short distance, higher vels, accs
Medium Range Travel – Constraint Influences

- parameter set 1
- increased velocities
- reduced trans. velocity
Updating Trajectories

Why
- plan longer trajectories by stitching new one
- react to unmapped obstacles

Procedure
- predict
  - position
  - velocity
- join new segment, continuous in
  - curvature
  - velocity
- choose $t_{\text{plan}}$
  (exploit anytime)

1. position at re-planning time
2. assumed position at start time of new plan
Medium Range Travel – Driven Paths

- overall planned trajectory
- superseded trajectories
- estimated driven path

(a) reduced updating
(b) frequent updating, every $\approx 1.65$ s

- re-planning also accounts for odometry drift, localization error
Unmapped Obstacle

Planned trajectories over distance map of environment
Unmapped Obstacle
Narrow Passage

- passage width 120 cm
- robot width 86 cm,
- safety margin 20 cm,
- discretization margin 5 cm

(a) driven path over distance map

(b) orientation of driven path

- corridor width: 9 cm
- manually supplied waypoints
Narrow Passage
Conclusion

Trajectory modeling
- represents orientational component
- enables optimization starting from initial paths

Optimization
- towards optimal trajectory
- different cost functions

Experimental results
- precise tracking of trajectories
- low prediction error
- unmapped obstacles
- plan longer trajectories by curvature continuous stitching