Online Generation of Kinodynamic Trajectories

Boris Lau, Christoph Sprunk,

Patrick Pfaff, Wolfram Burgard

University of Freiburg, Germany

KUKA GmbH, Germany

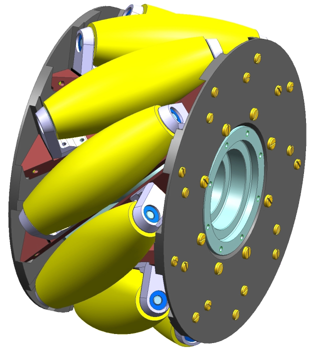
Introduction

- Given a map and a set of waypoints…
- Generate smooth trajectories for mobile robots

Neobotix ME-470

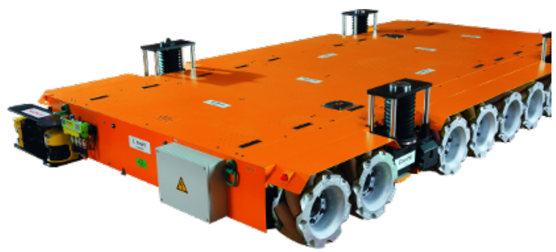
Permobil C500 Corpus

Differential Drive



KUKA omniWheel

KUKA omniRob

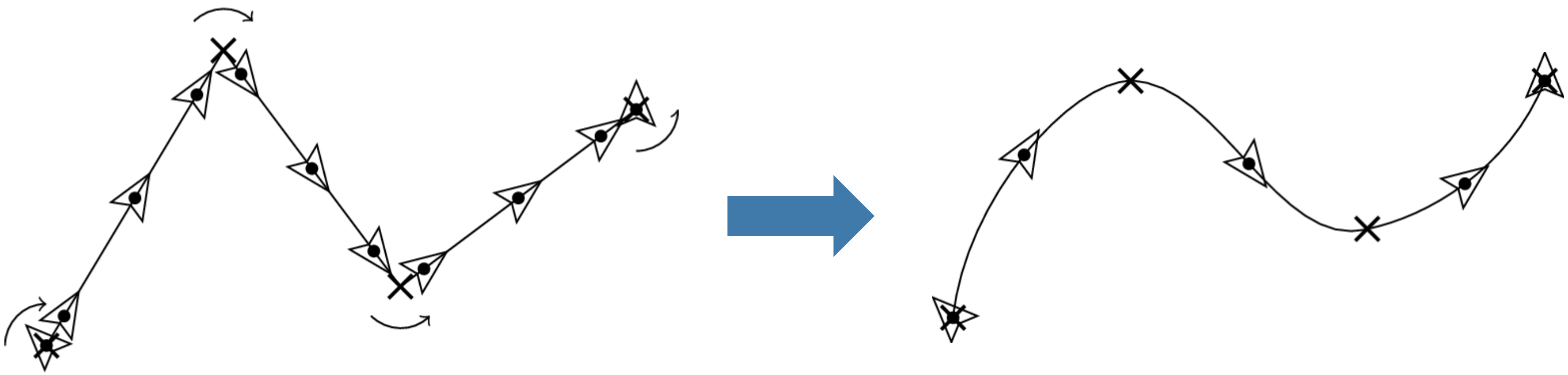


KUKA omniMove

Omnidirectional Drive

Introduction

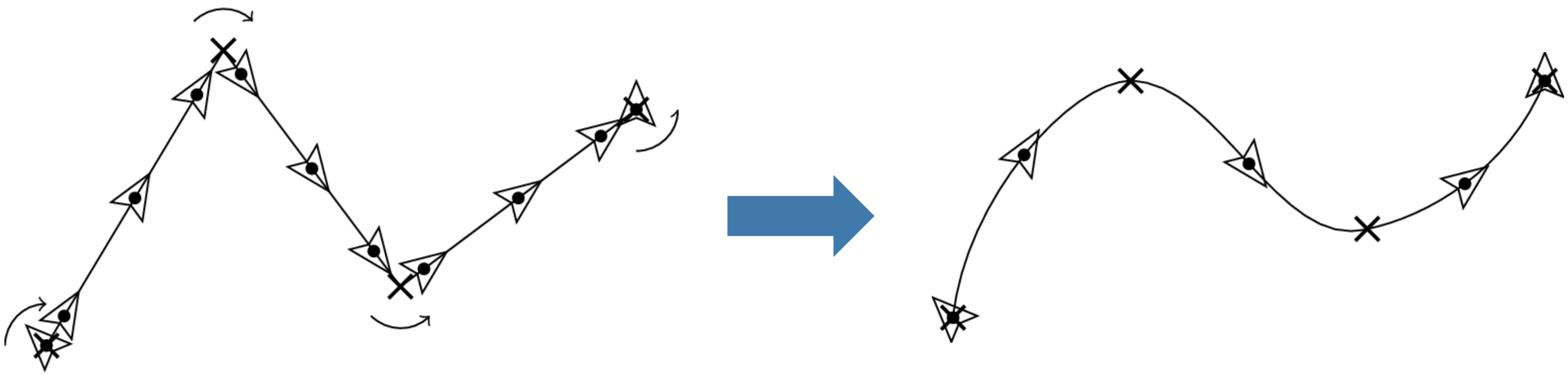
- Given a map and a set of waypoints…
- Generate smooth trajectories for mobile robots



Waypoint Path **Optimized Trajectory**

Introduction

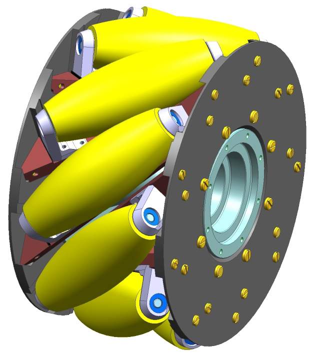
- A trajectory Q(t) defines robot poses for every point in time



Waypoint Path **Optimized Trajectory**

Introduction

- A trajectory Q(t) defines robot poses for every point in time



KUKA omniWheel

$$\langle x, y \rangle = Q(t)$$

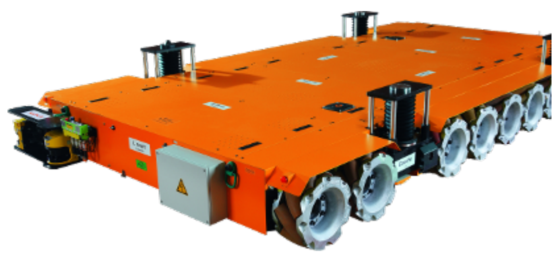
Neobotix ME-470

Permobil C500 Corpus

Differential Drive

$$\langle x, y, \theta \rangle = Q(t)$$

KUKA omniRob



KUKA omniMove

Omnidirectional Drive

Introduction

- A trajectory Q(t) defines robot poses for every point in time

$$\langle x, y \rangle = Q(t) \quad \Big| \quad \langle x, y, \theta \rangle = Q(t)$$

- Derivatives specify velocities

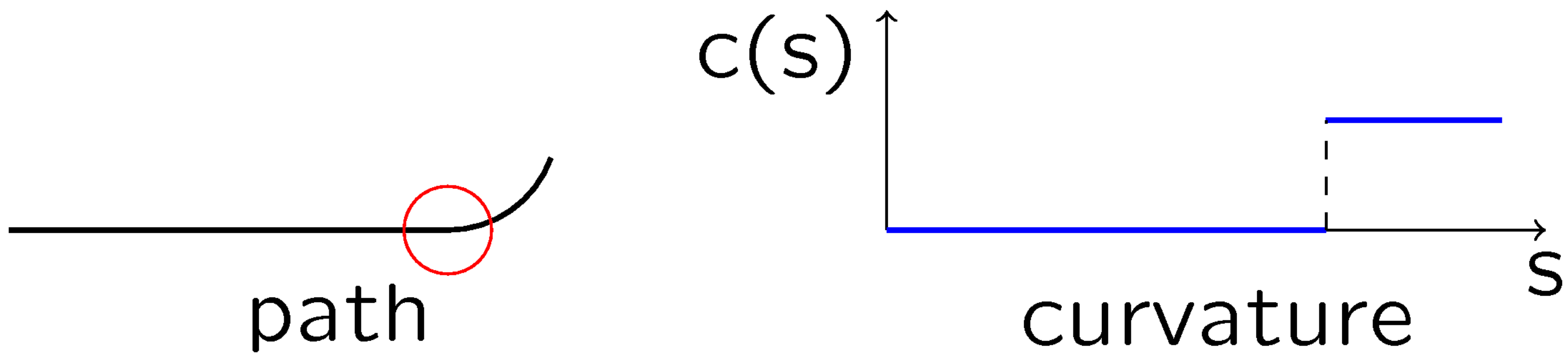
$$v = \left\| \dot{Q} \right\| \qquad \langle \dot{x}, \dot{y}, \dot{\theta} \rangle = \dot{Q}$$

$$\omega = v \cdot \frac{\dot{Q} \times \ddot{Q}}{\left\| \dot{Q} \right\|^3} \qquad \langle \ddot{x}, \ddot{y}, \ddot{\theta} \rangle = \ddot{Q}$$

Differential Drive | **Omnidirectional Drive**

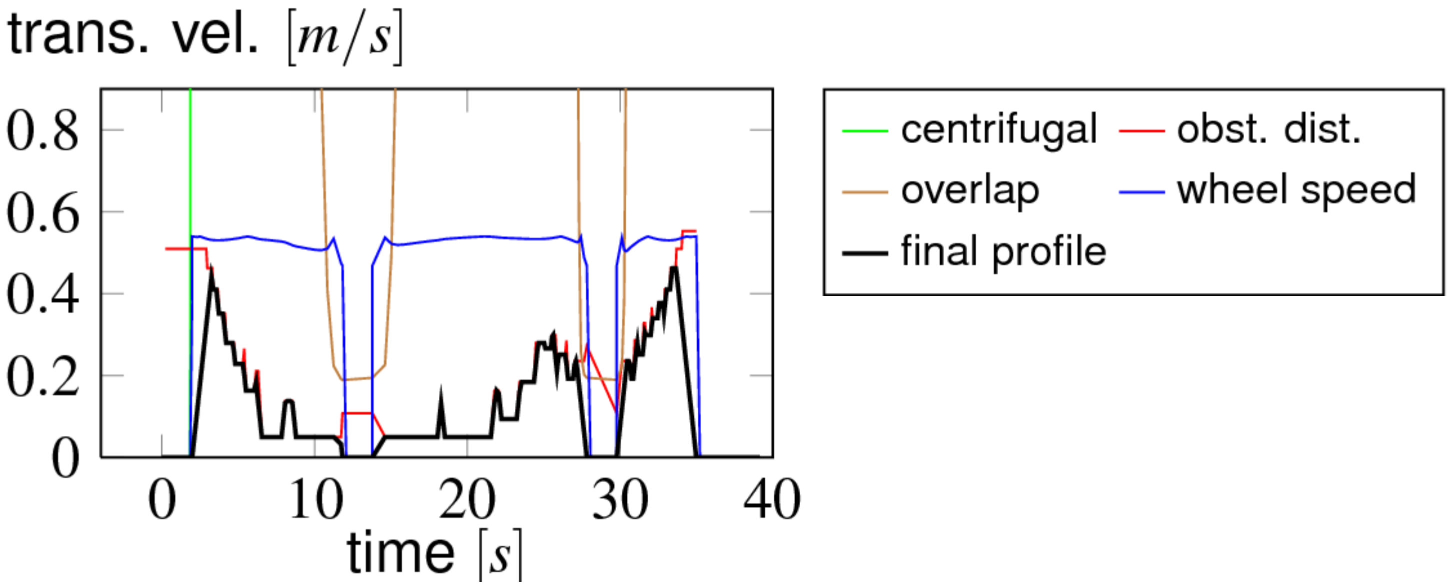
Introduction

- Real-world application requirements (1)
  - Smooth trajectories, continuous curvature
  - Discontinuities cause problems:
    For car: turn steering wheel infinitely fast
    For differential drive: infinite acceleration
  - If impossible, deviation from trajectory



Introduction

- Real-world application requirements (2)
  - Feasible trajectories
    - Kinodynamic constraints by platform or load
    - Max. velocity, acceleration, centripetal force
    - Violation → deviation from planned trajectory and possibly damage to hardware or load



Introduction

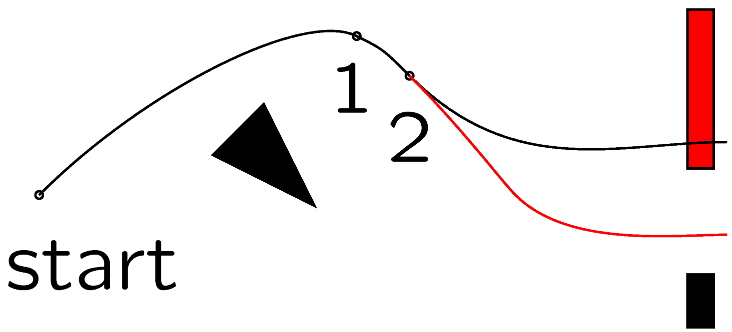
- Real-world application requirements (3)
  - Reasonable Trajectories
    - Short → no unnecessary detours
    - Fast → smooth curves where possible, rather than stopping and turning on the spot
    - Energy efficient, e.g., forward for omniDrive
    - As expected by humans, e.g., prefer forward
    - Our approach: user-defined cost functions

Introduction

- Real-world application requirements (4)
  - Complete Planning
    - If a possible trajectory exists, find it
  - Avoid obstacles
    - Consider obstacles during planning
    - Consider unmapped and unexpected obstacles
  - Robot shape can be non-circular

Introduction

- Real-world application requirements (5)
  - Replan while moving
    - to account for unexpected obstacles/passages
    - to account for odometry drift
    - to recover from localization errors
    - We need to be able to attach new trajectory pieces online and without discontinuities



1 position at replanning time
2 start of new segment

Related Work

Reviewer 2: “The term "**kinodynamics**" is not known in the fields of kinematics, dynamics, or control”.

- A lot of previous work…
  - "Randomized **kinodynamic** Planning" by LaValle, Kuffner Jr., 679 citations (google scholar)
  - "Randomized **kinodynamic** motion planning with moving obstacles" by Hsu et al., 394 citations
  - 1500 other papers for “**kinodynamic** planning”
- Very passionate attacks by reviewers
- Existing approaches did not satisfy our requirements
- Toy problem solutions claim too much

Related Work

Reviewer 2: "First, the idea of trajectory optimization [...] is much older than ALL of the cited references. Similar work was done in the mid 80's by [3 references] – two used cubic splines to model the trajectory."

Cubic curves: no C2 continuity and replanning at the same time

	Cubic Polynomials		Clothoids
Property	Bézier	B-Spline	Cl.-Spline
C^1 continuity	✓	✓	✓
C^2 continuity	-	✓	✓
passes control points	✓	-	✓
strong correlation	✓	-	✓
localism	✓	-	✓
freely set 1st deriv. start	✓	✓	-
freely set 2nd deriv. start	-	-	-

Smooth / Accuracy

Replanning

Related Work

- Curvature discontinuities
  - C. Mandel and U. Frese: “Comparison of wheelchair user interfaces for the paralysed: Head-joystick vs. verbal path selection from an offered route-set”, ECMR 2007
  - T. M. Howard and A. Kelly: “Optimal rough terrain trajectory generation for wheeled mobile robots”, Intl. Journal of Robotics Research, vol. 26, pp. 141–166, 2007.
  - M. Likhachev and D. Ferguson: “Planning long dynamically-feasible maneuvers for autonomous vehicles”, Robotics: Science and Systems (RSS), Zurich, 2008
- No replanning → static environments
  - Z. Shiller and Y. Gwo: “Dynamic motion planning of autonomous vehicles,” IEEE Trans. on Robotics and Automation, vol. 7, 1991.

Related Work

- No consideration of obstacles at all
  - D. J. Balkcom, P. A. Kavathekar, and M. T. Mason: "Time-optimal trajectories for an omni-directional vehicle", Intl. Journal of Robotics Research, vol. 25, no. 10, pp. 985–999, 2006.
  - O. Purwin and R. D'Andrea: "Trajectory generation and control for four wheeled omnidirectional vehicles", Robotics and Autonomous Systems, vol. 54, pp. 13–22, 2006.

Related Work

- Path deformation without guaranteed solution (not complete)
  - F. Lamiraux, D. Bonnafous and O. Lefebvre: "Reactive Path Deformation for Nonholonomic Mobile Robots", IEEE Transactions on Robotics, vol 20, No 6, December 2004.
  - J. Connors and G. Elkaim: "Manipulating B-Spline based paths for obstacle avoidance in autonomous ground vehicles", National Meeting of the Institute of Navigation, San Diego, USA, 2007.

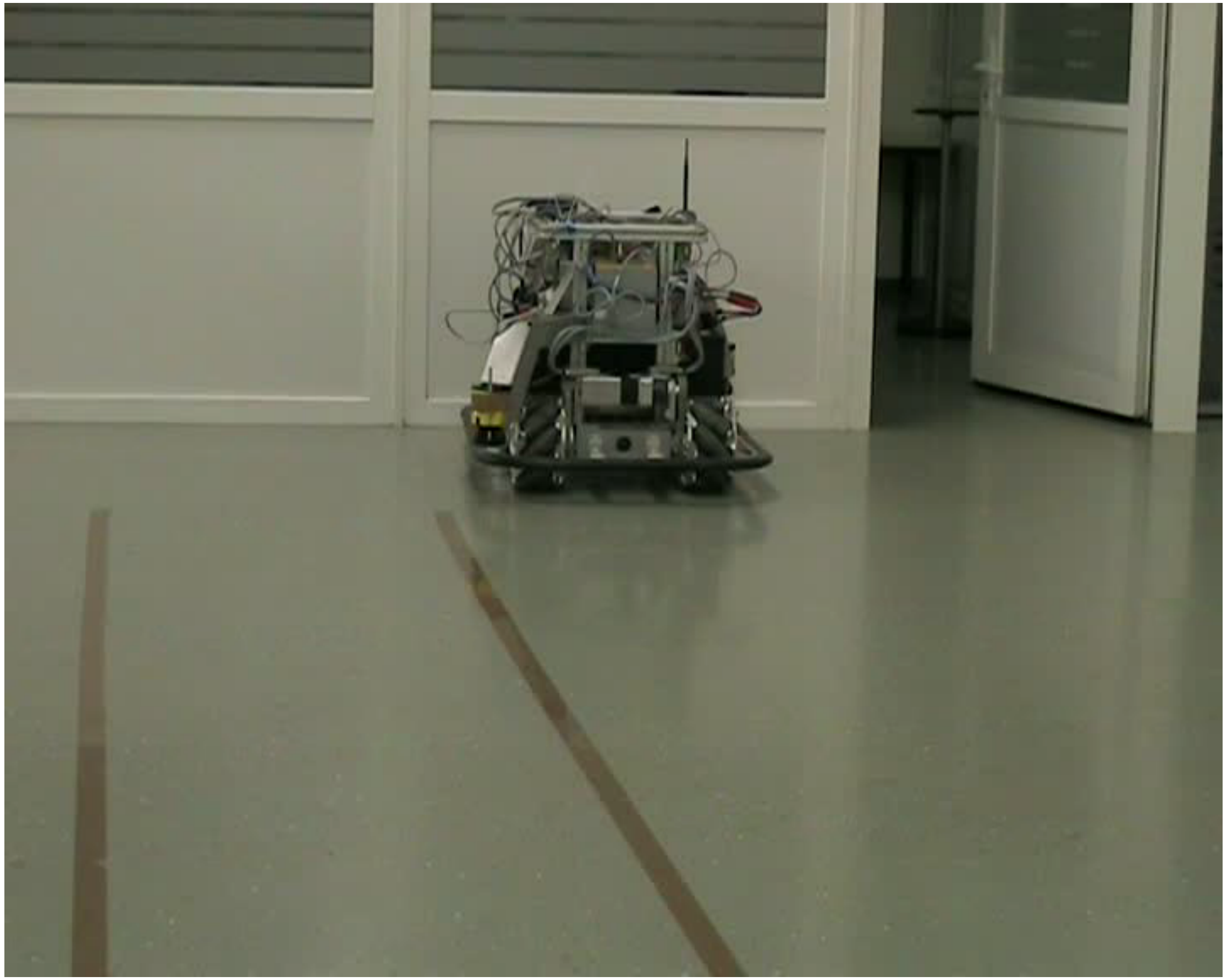
Related Work

- Search based approaches
  - Suffer from curse of dimensionality:
    - coarse discretization of actions,
    - restriction to lane-changing, or
    - no planning of velocities.
  - Omnidirectional: additional dimensions

- M. Likhachev and D. Ferguson: “Planning long dynamically-feasible maneuvers for autonomous vehicles”, Robotics: Science and Systems (RSS), Zurich, 2008
- S. M. LaValle and J. J. Kuffner: “Randomized kinodynamic planning”, International Journal of Robotics Research, 20(5): 378--400, May 2001.

Related Work

- Omnidirectional vs. Differential Drive



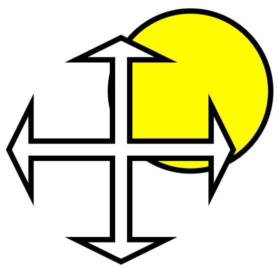
Related Work

- Omnidirectional vs. Differential Drive
  - No non-holonomic constraint
  - Orientation is an independent dimension
    → state space and control space larger
    → trajectories have more parameters

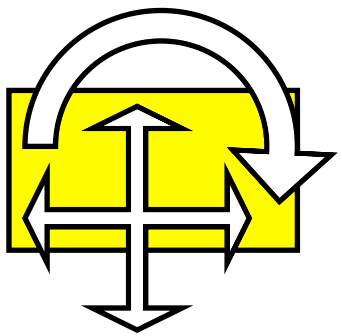
Platform	State space	Control space
Orientation-free holonomic	4D: $x, y, \dot{x}, \dot{y}$	2D: $\ddot{x}, \ddot{y}$
Differential drive / Ackermann	5D: $x, y, \theta, v, \omega/\phi$	2D: $\dot{v}, \dot{\omega}/\dot{\phi}$
Non-circular holonomic	**6D:** $x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}$	**3D:** $\ddot{x}, \ddot{y}, \ddot{\theta}$

Related Work

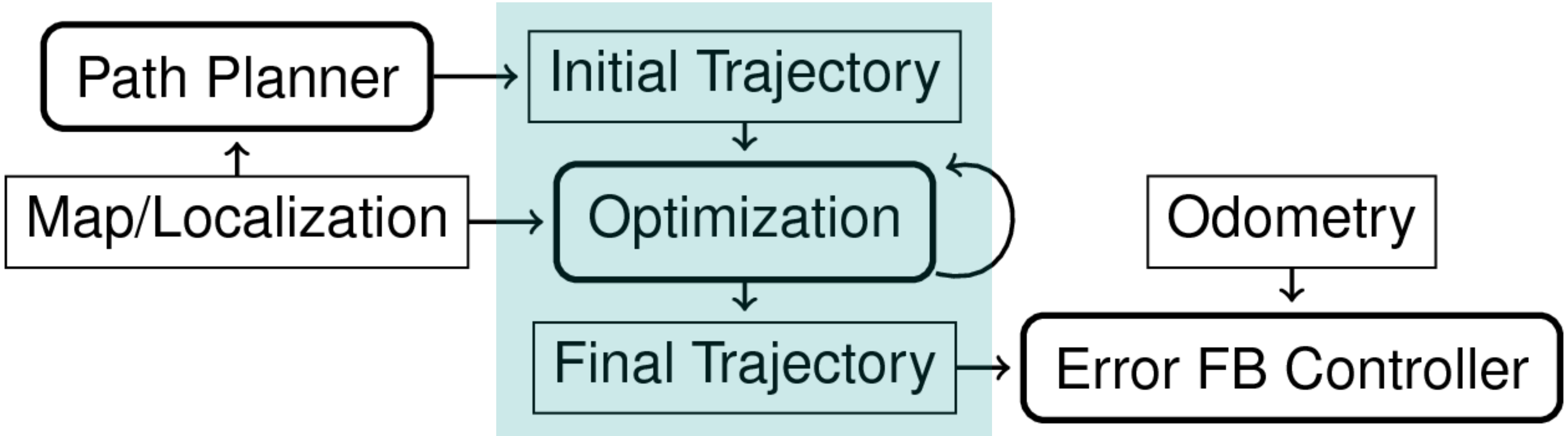
- For omnidirectional platforms:
  no orientation at all (x-y only)
  - O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach", Intl. Conf. on Robotics and Automation, vol. 1, Detroit, USA, 1999.
  - D. Hsu, R. Kindel, J.C. Latombe, S. Rock: "Randomized Kinodynamic Motion Planning with Moving Obstacles", Intl. J. of Robotics Research, 21(3), 2002.



vs.

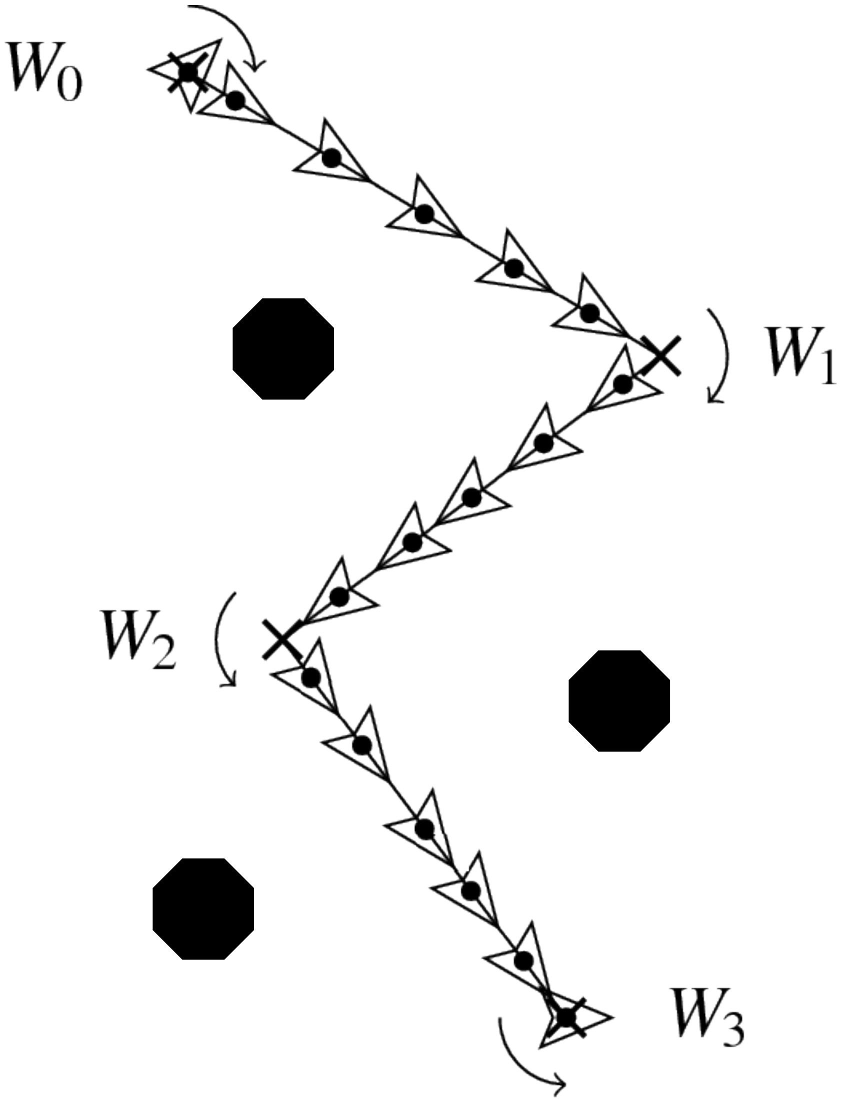


System Overview



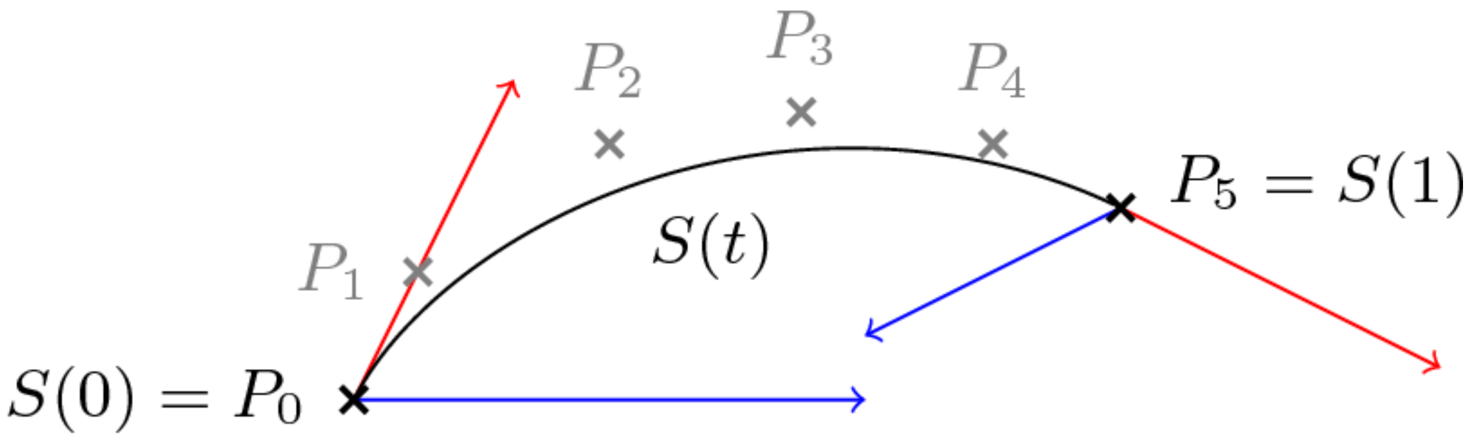
- Initial trajectory connects waypoints
- (Anytime) optimization w.r.t. cost function
- Separation of trajectory generation and execution
  - small FB loop
  - abstraction

Initial Path



- Sequence of turns on spot and translations
- Collision free
- Waypoint-providing algorithms:
  - A*
  - Piano mover
  - RRTs
  - Voronoi graph based
  - …

Quintic Bézier Splines

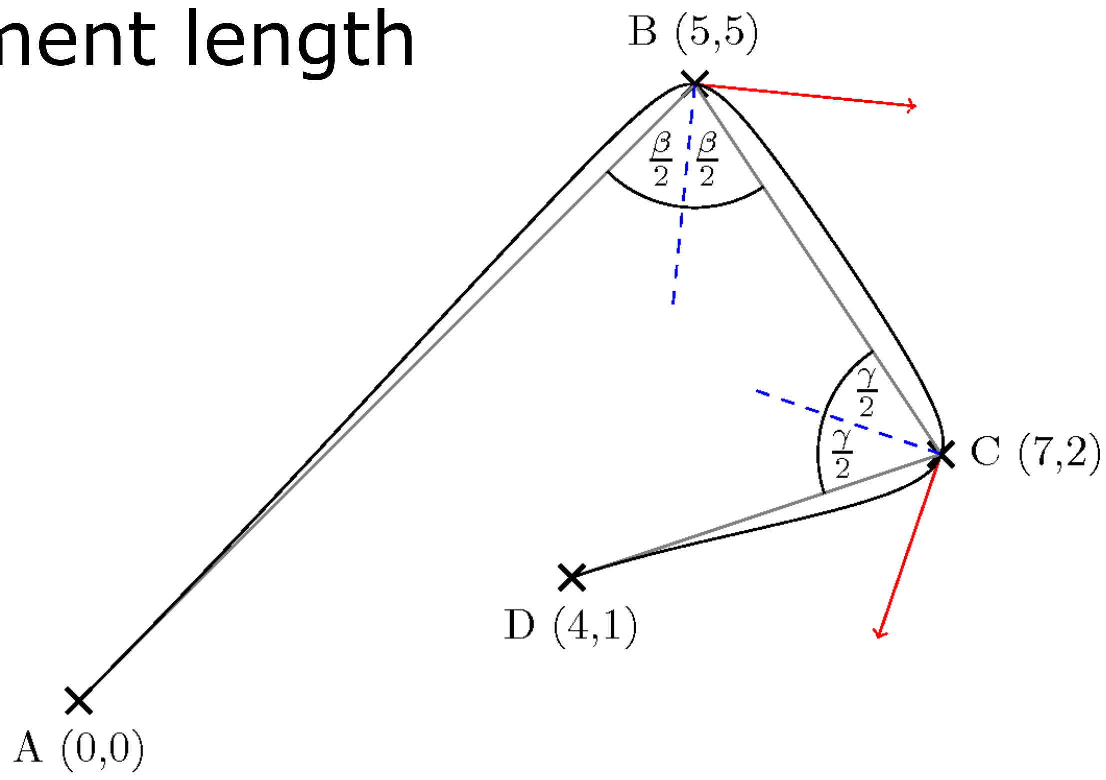


Quintic Bézier spline segment (2D polynomial)

- Compared to cubic splines, additional degrees of freedom allow to
  - choose 1st and 2nd derivative at start/end
  - join segments with continuous curvature
  - use heuristics to mimic cubic splines

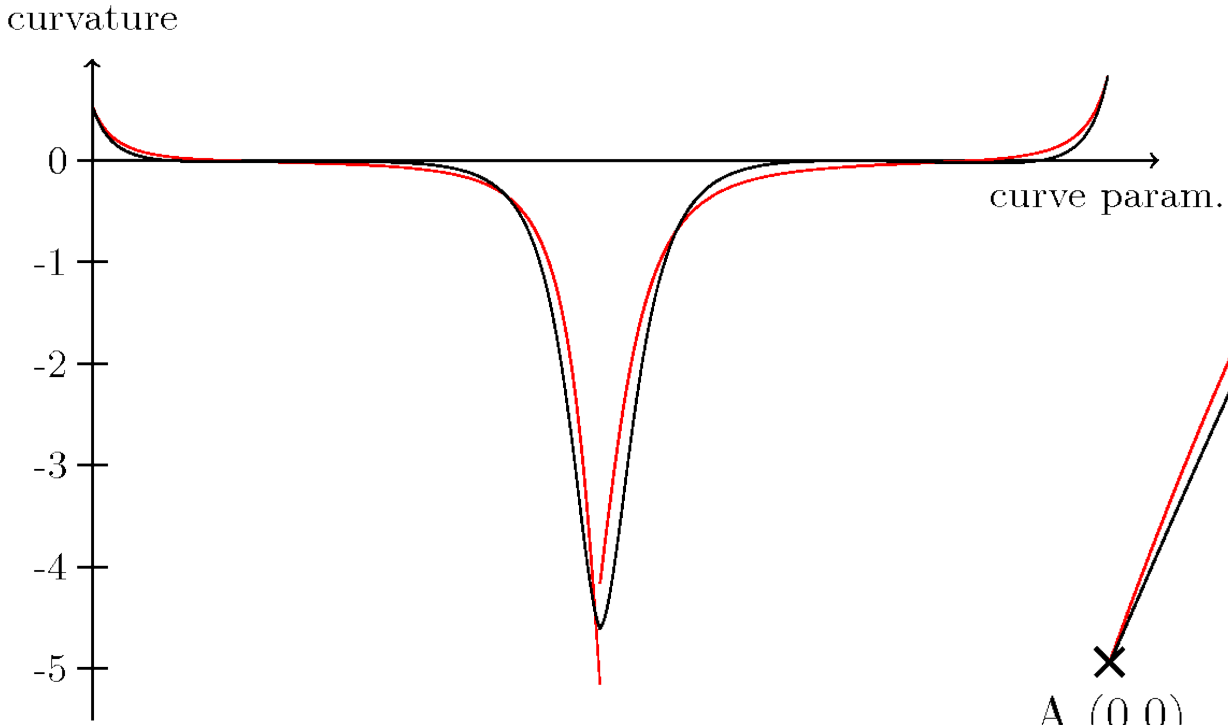
Quintic Bézier Spline Heuristics

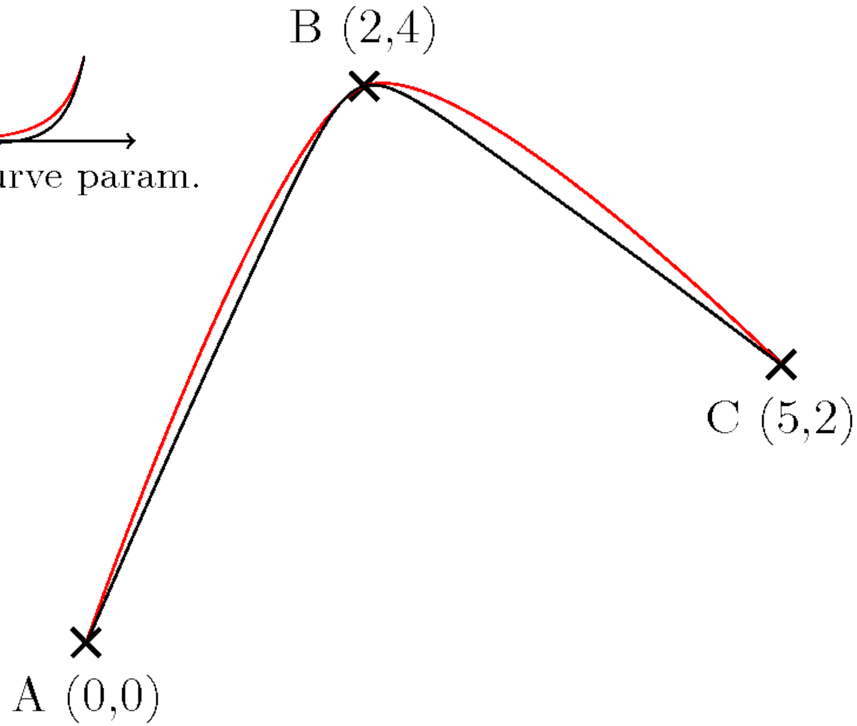
- Tangent properties
  - perpendicular to angle bisector
  - magnitude proportional to minimum adjacent segment length



Quintic Bézier Spline Heuristics

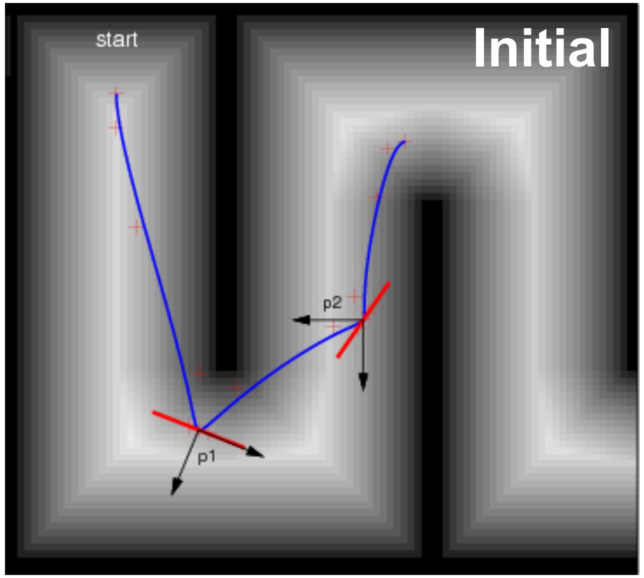
- Second derivatives
  - set to weighted average of values for corresponding Cubic spline
  - mimics Cubic Bézier spline behavior

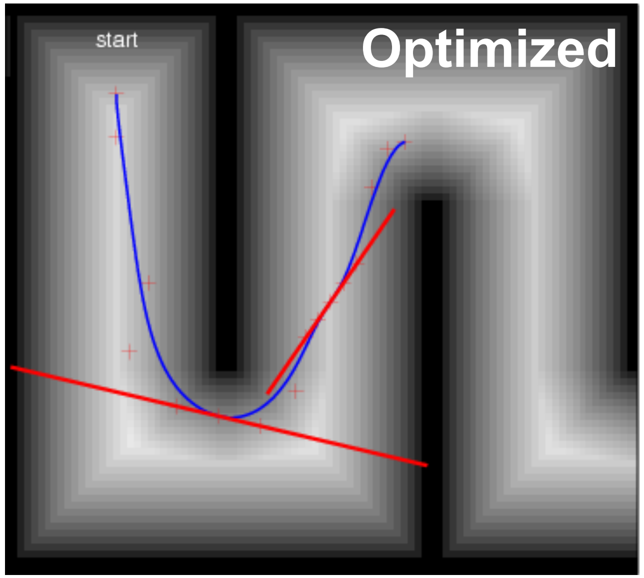




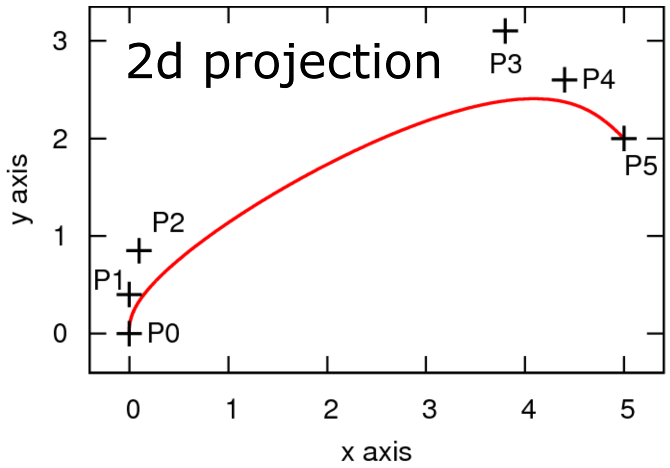
Path Modeling

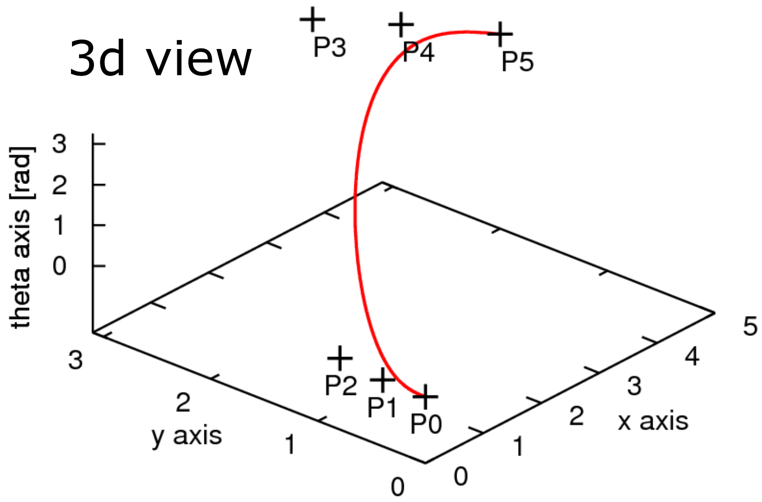
- Connect waypoints with spline segments
- Initially: small derivatives, approximates straight lines
- Optimization adjusts
  - position of waypoints
    → increases obstacle distance
  - lengths of tangents
    → smooth curves





Omnidirectional Paths



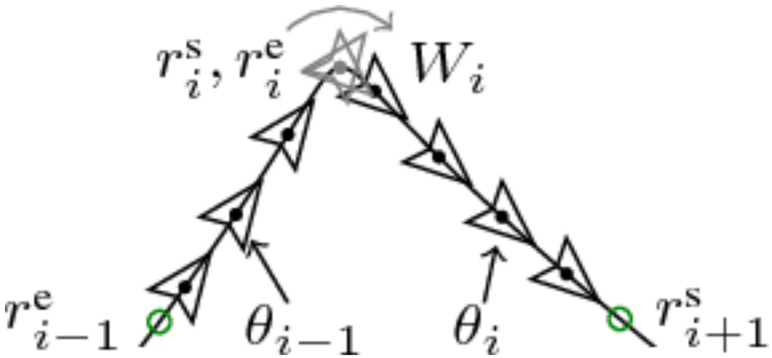


Quintic Bézier pline segment (3D polynomial)

- Additional dimension for orientation Θ
  - Simple spline interpolation causes continuous rotation, but we need pure translations as well
  - Idea: add an additional model layer for Θ

Distribution of Rotation

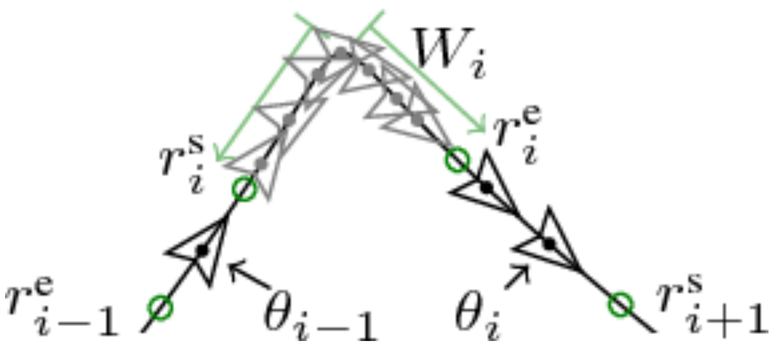
- Rotation control points
  - Determine where rotation takes place
    - mark start and end of rotation
  - Insertion around waypoints
  - 2D shape not affected



If points coincide with waypoint, the robot turns on the spot at them

Distribution of Rotation

- Rotation control points
  - Determine where rotation takes place
    - mark start and end of rotation
  - Insertion around waypoints
  - 2D shape not affected



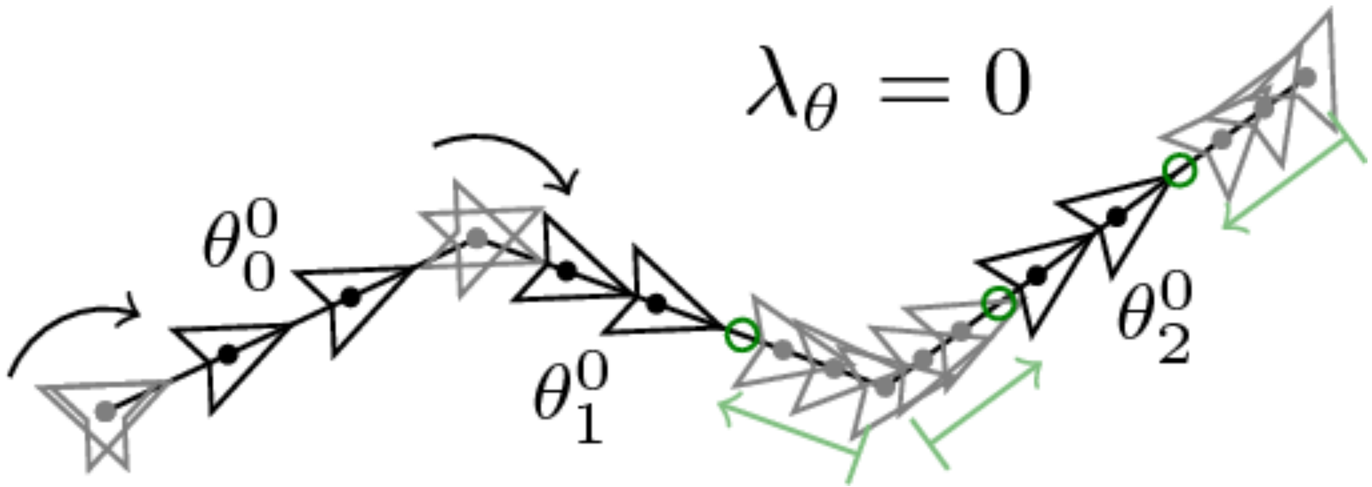
If the points are moved along the segments, we obtain simultaneous translation and rotation

Omnidir. Paths – Orientations

- Interpolation between two extremal orientation profiles via $\lambda_\Theta \in [0, 1]$

(a) Orientation as obtained from the waypoint providing algorithm, corresponds to $\lambda_\Theta=0$.

(b) Completely distributed orientation, corresponds to $\lambda_\Theta=1$.

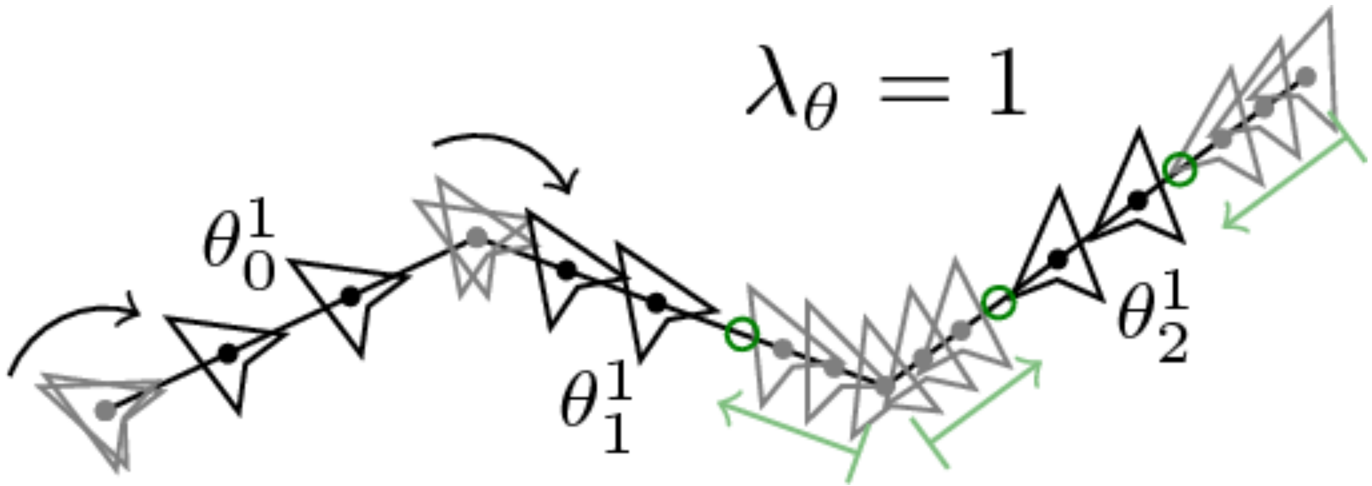


Omnidir. Paths – Orientations

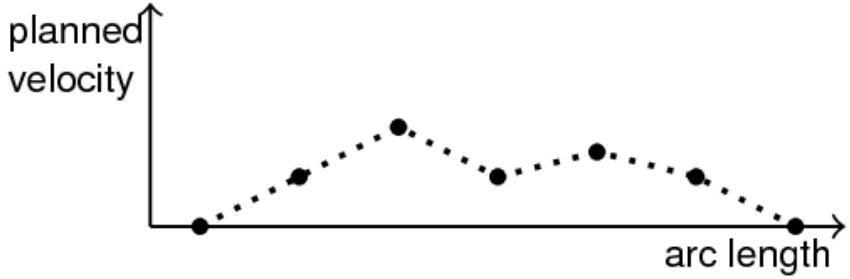
- Interpolation between two extremal orientation profiles via $\lambda_\Theta \in [0, 1]$

(a) Orientation as obtained from the waypoint providing algorithm, corresponds to $\lambda_\Theta=0$.

(b) Completely distributed orientation, corresponds to $\lambda_\Theta=1$.



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 acceleration constraints (not platform limits)
- Discretization depending on translation, rotation

Optimization

- Optimize path shape: cost function
  - time of travel, path length, …
  - energy efficiency, steering effort, …
- Parameters
  - 2D waypoint and tangent lengths
  - For omnidirectional robots:
    - Waypoint tangent, rotational component
    - Rotation control points
    - Rotational movement distribution (combined dimension for orientation at waypoints, rotation control points)

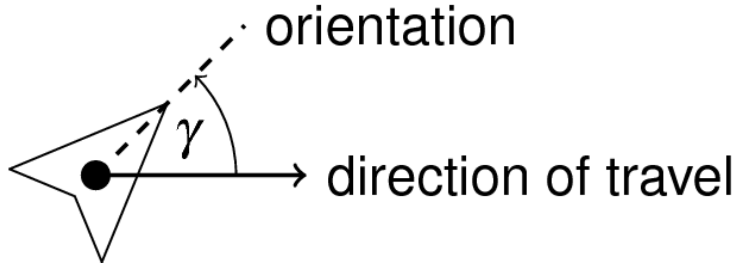
Optimization – Algorithm

- RPROP inspired (Resilient backPROPagation)
  - Derivative free
  - Robust convergence

- While planning time left
  - Optimize parameters independently
  - Continue with next parameter after each improvement
- Cost function: cost = t_{travel}

Optimization – Alternative Costs

- omniDrive: energy efficiency depends on direction of travel relative to robot's orientation

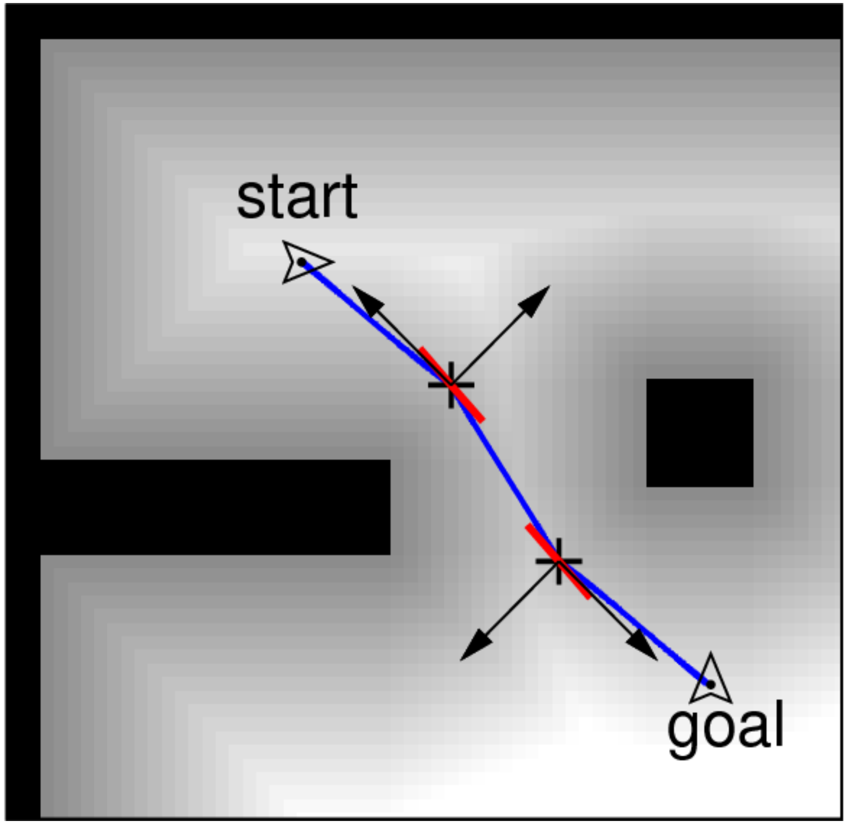


- Exemplary penalty function for non-forward travel

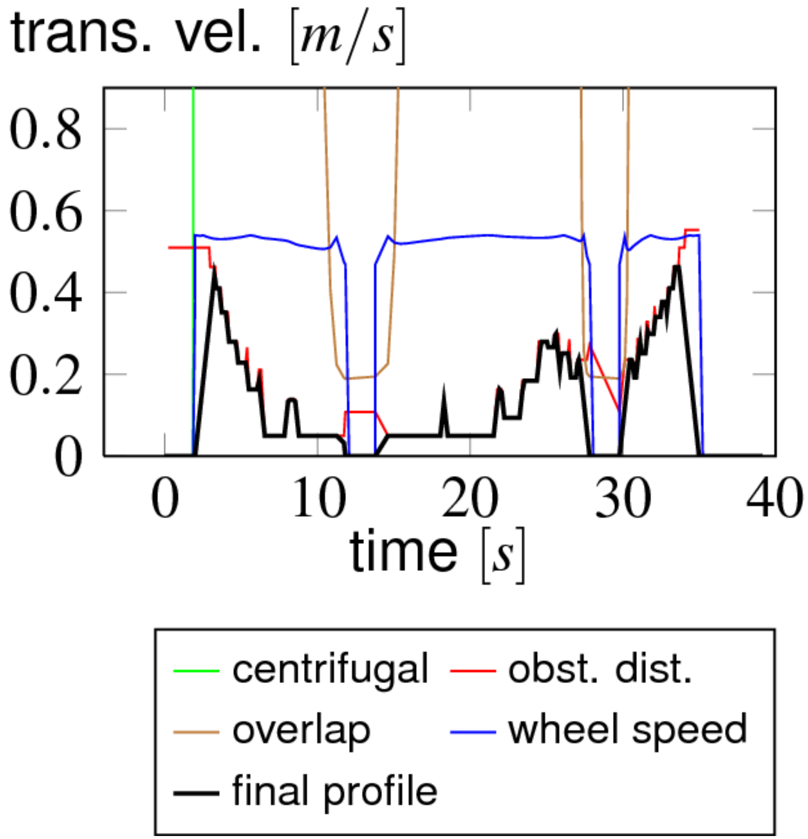
$$F = \frac{1}{\text{arc length}} \int_{0}^{\text{arc length}} 1 - |\cos \gamma(s)| \, \mathrm{d}s$$

$$\text{cost} = t_{\text{travel}} + \alpha \cdot F \cdot t_{\text{travel}}$$

Optimization – Examples

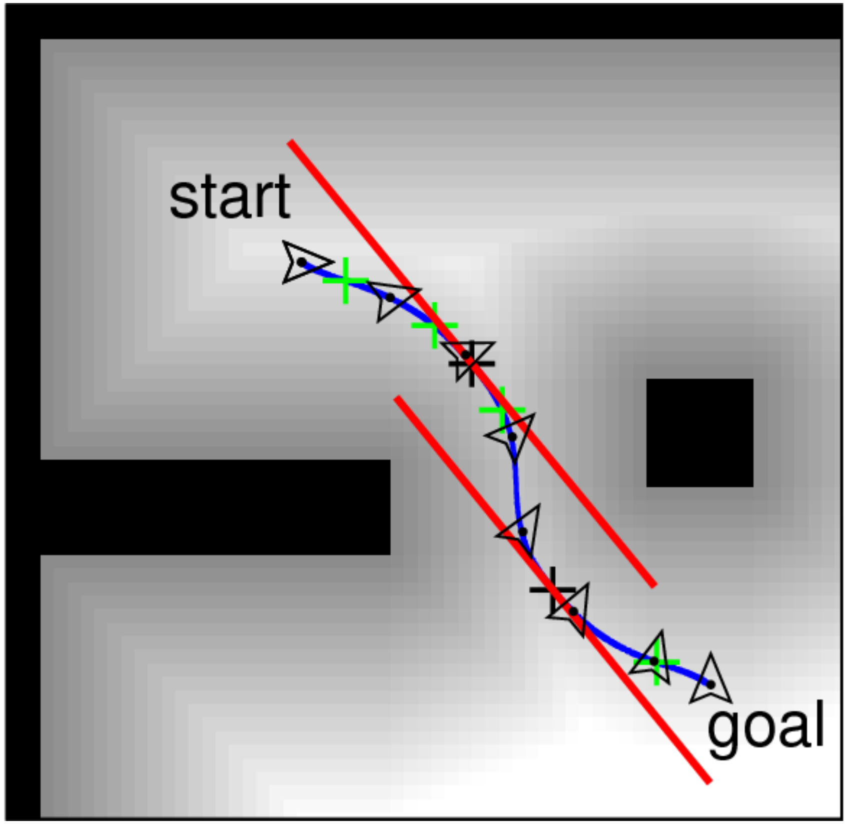


F=0, t_{travel}=39.15 s.

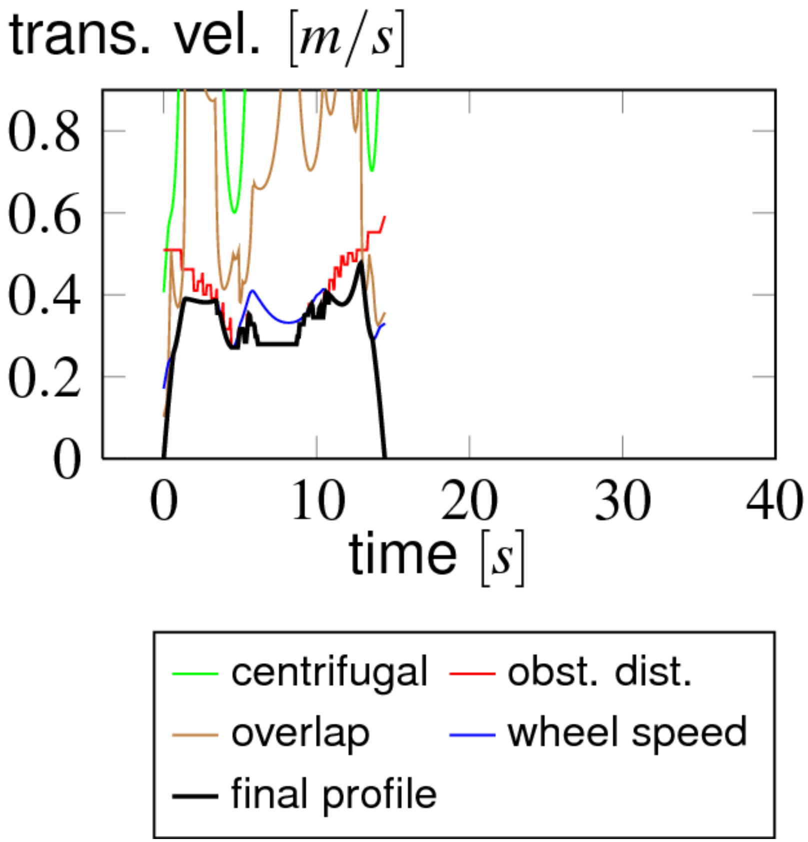


Initial Path

Optimization – Examples

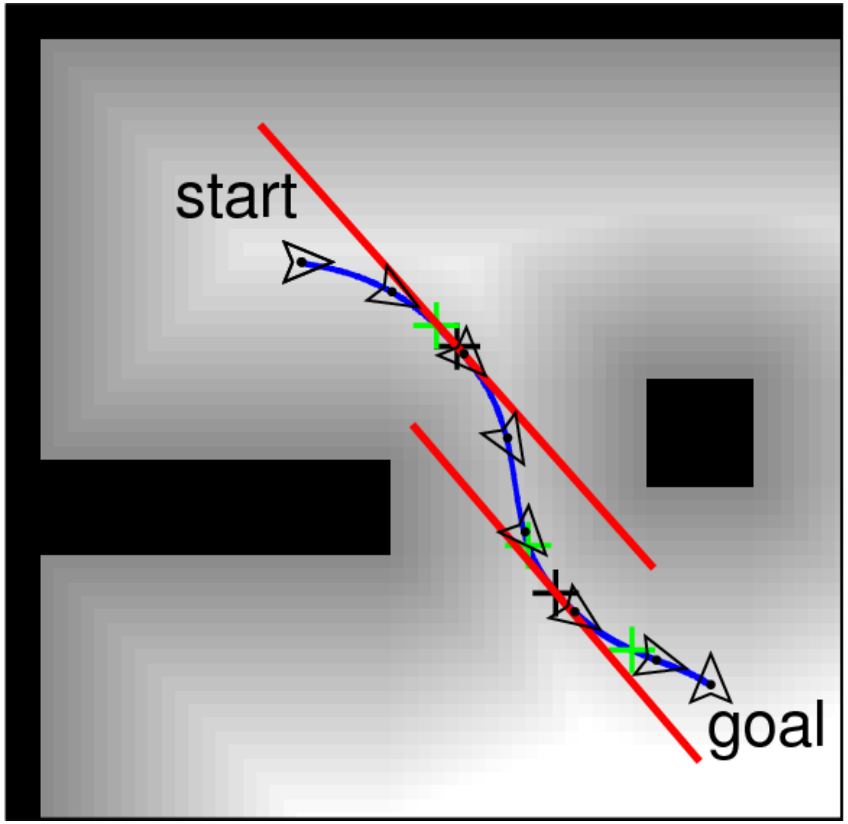


F=0.519, t_{travel}=14.46 s.

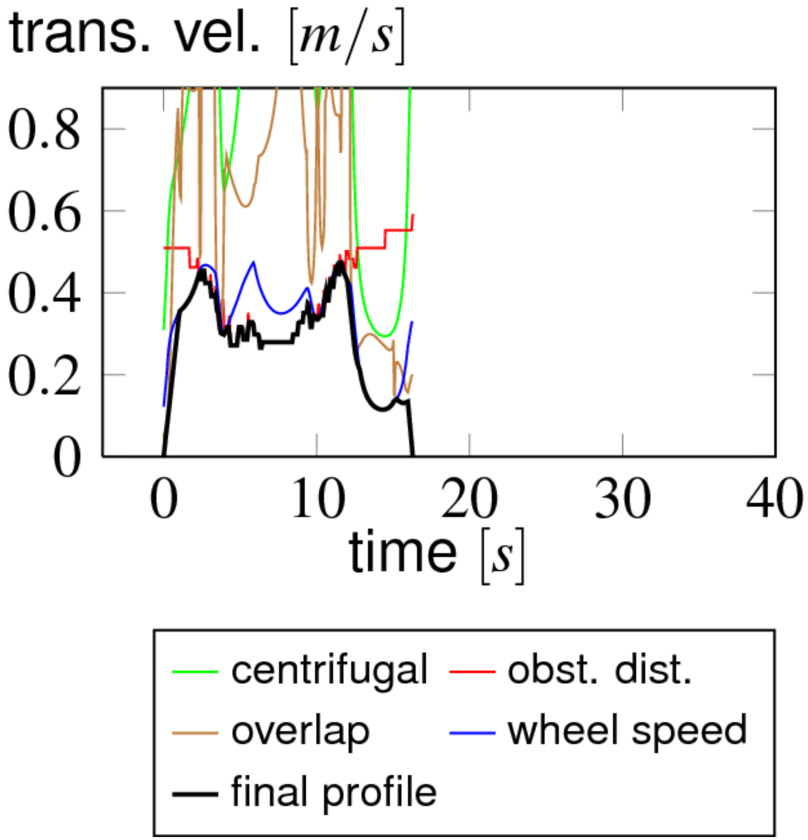


Optimized: time of travel

Optimization – Examples



F=0.067, t_{travel}=16.28 s.



Optimized: time of travel, energy efficiency

Feedback Controller

- For differential and synchro drive
  - Dynamic feedback linearization controller
  - Input: $Q(t), \dot{Q}(t), \ddot{Q}(t)$
  - Relies on good odometry
  - Accurate timestamps
  - High frequency

G. Oriolo, A. De Luca, M. Vendittelli: "WMR control via dynamic feedback linearization: design, implementation, and experimental validation", IEEE Transactions on Control Systems Technology, 10(6), Nov. 2002

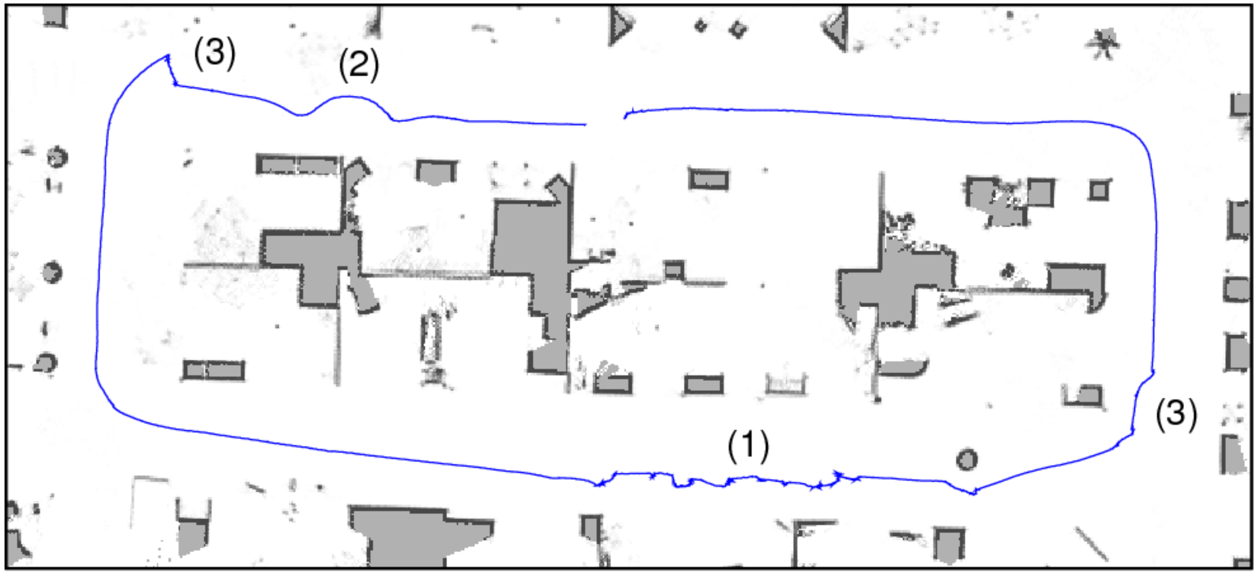
Feedback Controller

- For omnidirectional drive: self-made controller
- Computes a velocity command v_{com} at time step t:

$$v_{\text{com}} = \dot{T}(t + t_{\text{del}}) + \kappa \odot (T(t) - X(t))$$

- Time-parameterized trajectory T, derivative $\dot{T}$
- Robot pose estimate X (odometry)
- $\kappa = (2, 2, 0.2)^T$
- Command execution delay $t_{\text{del}} = 0.1$ seconds

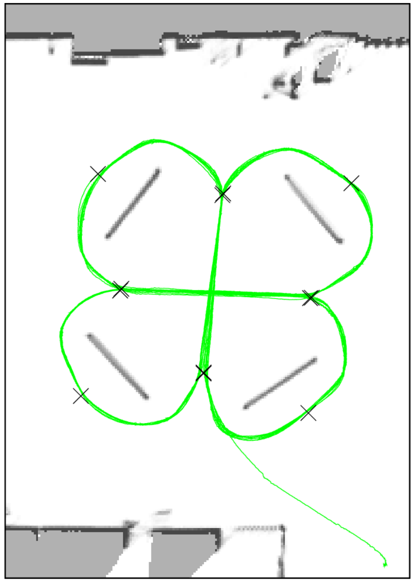
Experiment – Populated Area



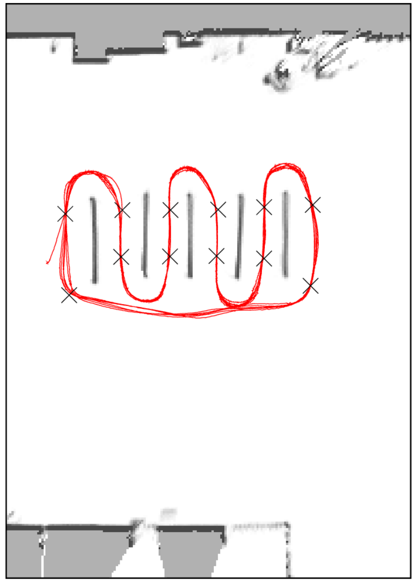
- Example run of at an exhibition
  - (1) was extensively challenged by visitors
  - (2) smoothly avoids obstacles seen in advance
  - (3) can make sharp turns when necessary

Experiment – Obstacle Courses

- Two obstacle courses
- Pre-defined waypoints
- Pioneer P3-DX robot
- Differential drive
- Odometry and laser
- Operate at 35 Hz



„clover“, 10 laps

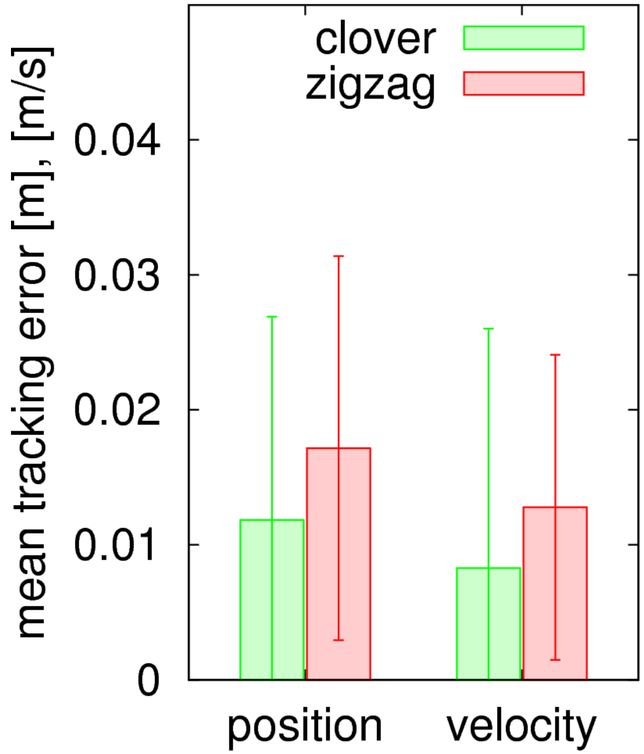


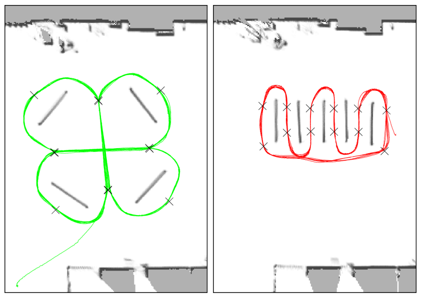
„zig-zag“, 5 laps

- v_{max} = 0.5 m/s
- Replanning every 0.4 s

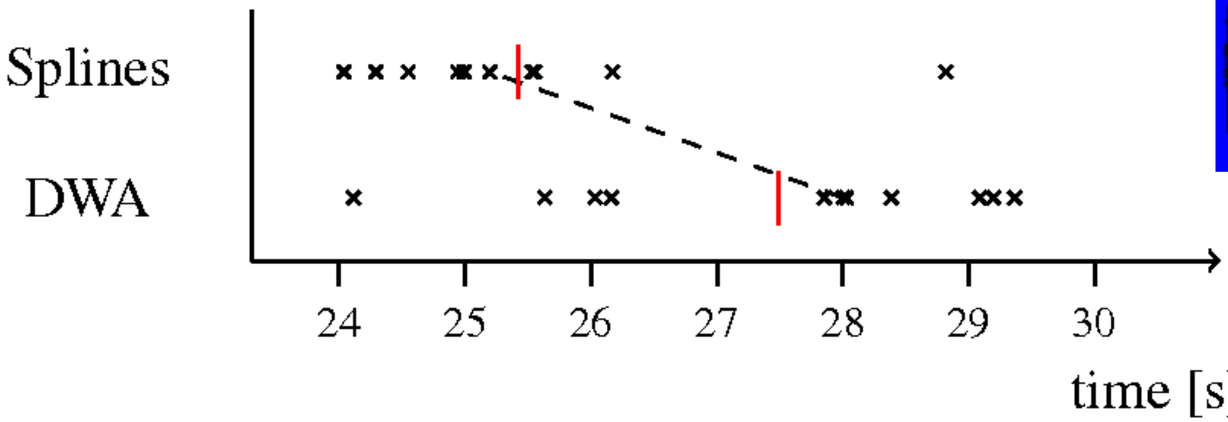
Experiment – Obstacle Courses

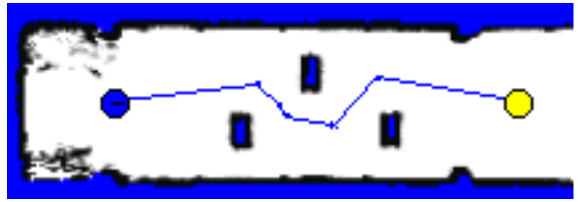
- Error: deviation from planned trajectory
- Averaged over time
- On average 1-2cm in position, 1-2cm/s in velocity
- Global error (localization), below map resolution (5cm)

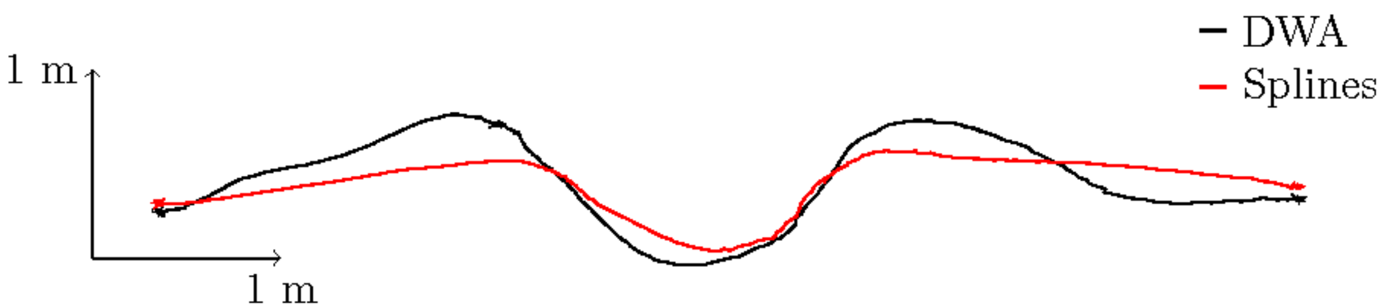




Experiment – Comparison DWA



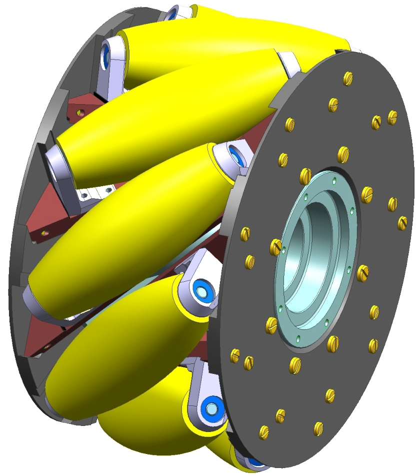




- 11 runs each, both with identical constraints
- Splines lead to faster and shorter paths

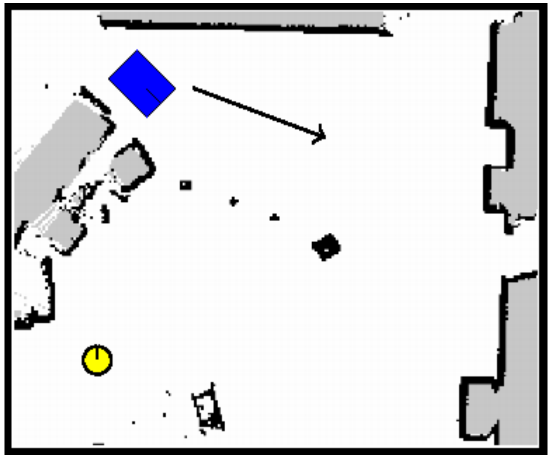
Omnidirectional Platform

(a) KUKA omniRob

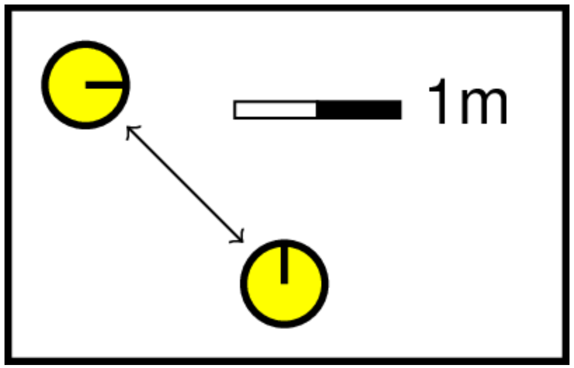


(b) KUKA omniMove wheel

Navigation Tasks



- Transportation task
- Map area approximately 11.2 m x 9.4 m



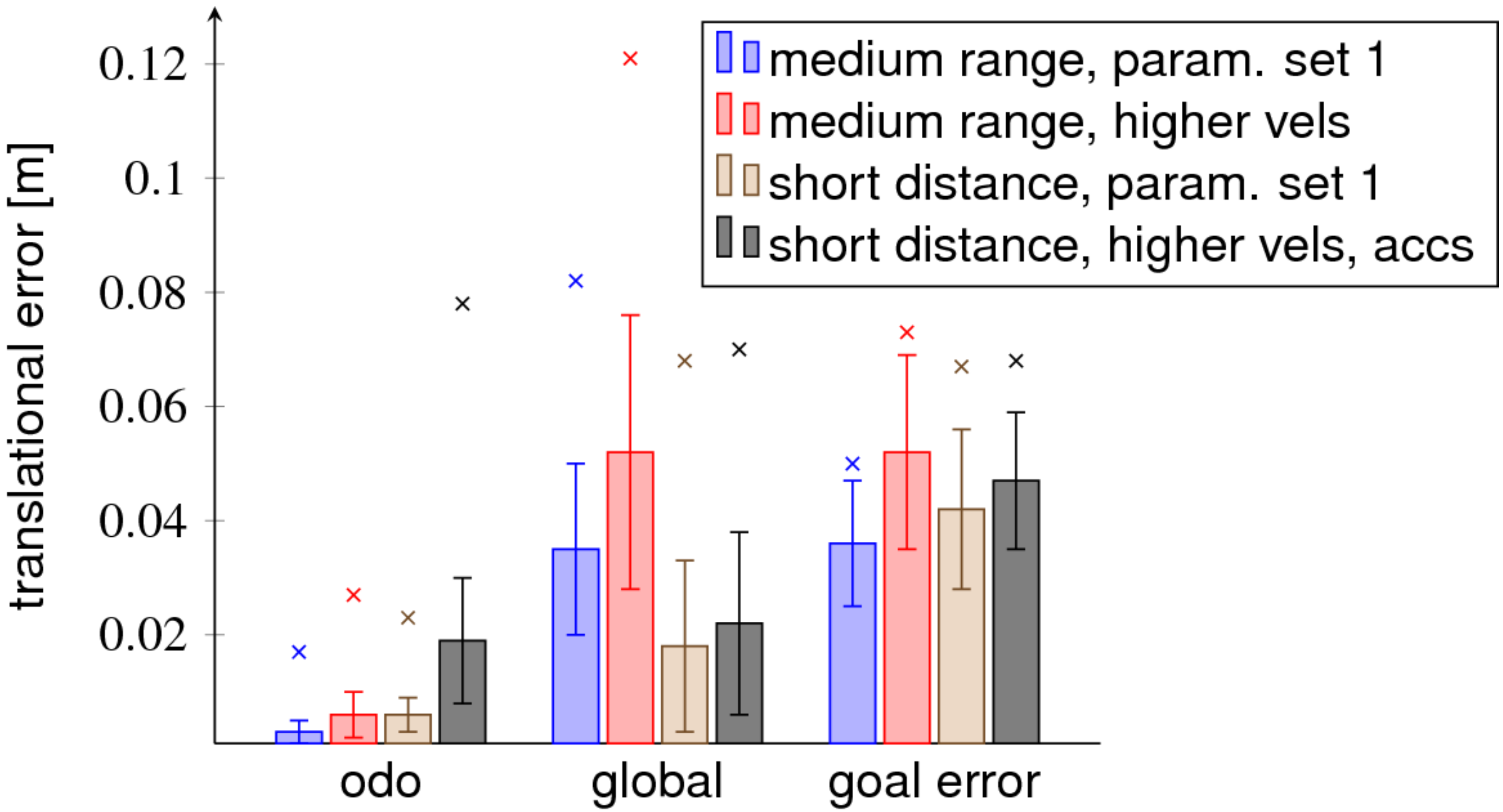
- Short distance reorientation
- Resemble repetitive pick & place task at high frequency

- Tasks executed with different constraint parameter sets

Transportation Task

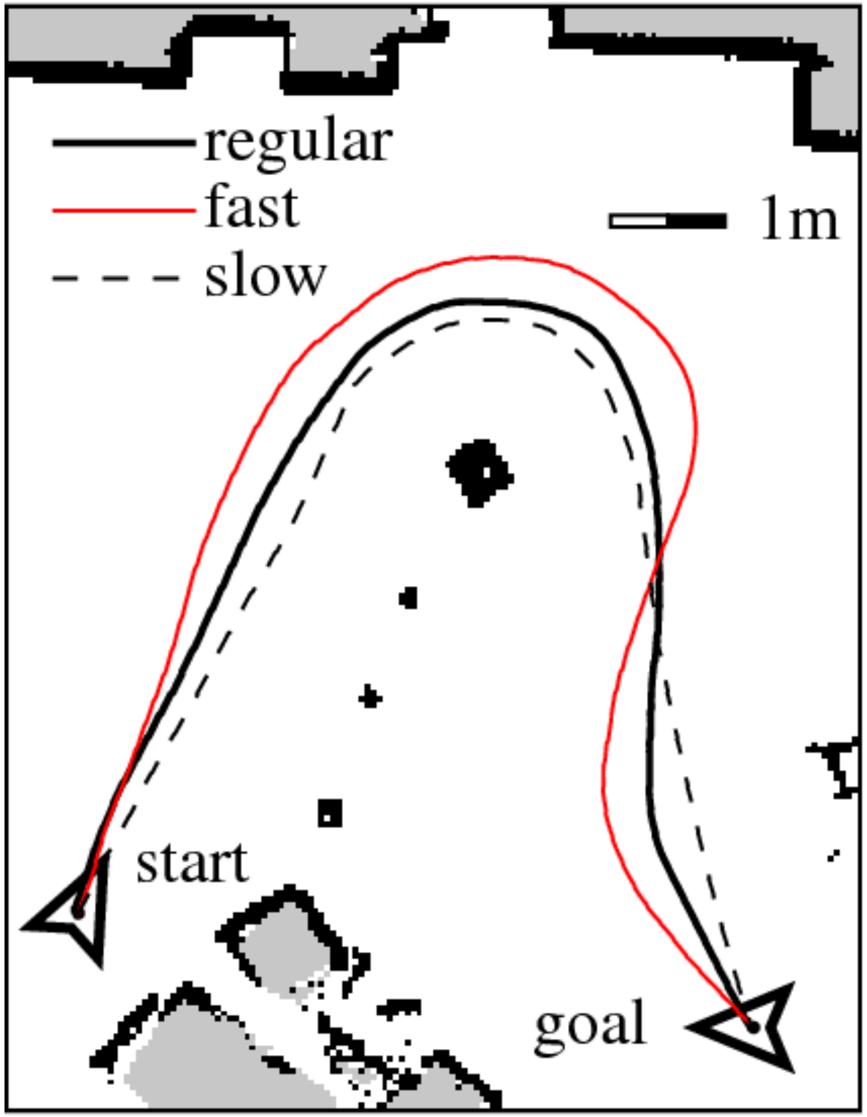
Short Distance Reorientation

Travel Tasks – Tracking Error



Translational tracking error

Transportation Task – Paths



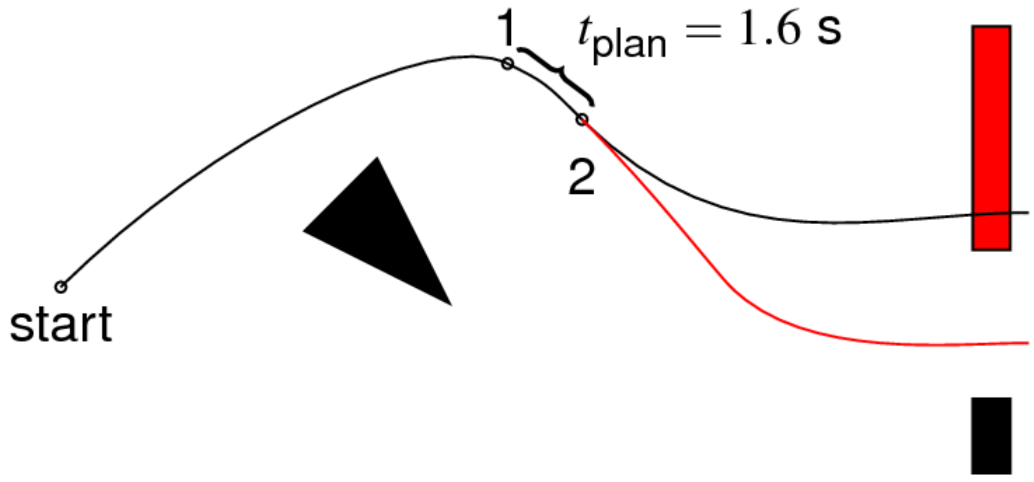
- Optimization adapts to constraints
- Obstacle distance
- Dynamics

Updating Trajectories

Why:

- Plan longer trajectories by stitching new one
- React to unmapped obstacles

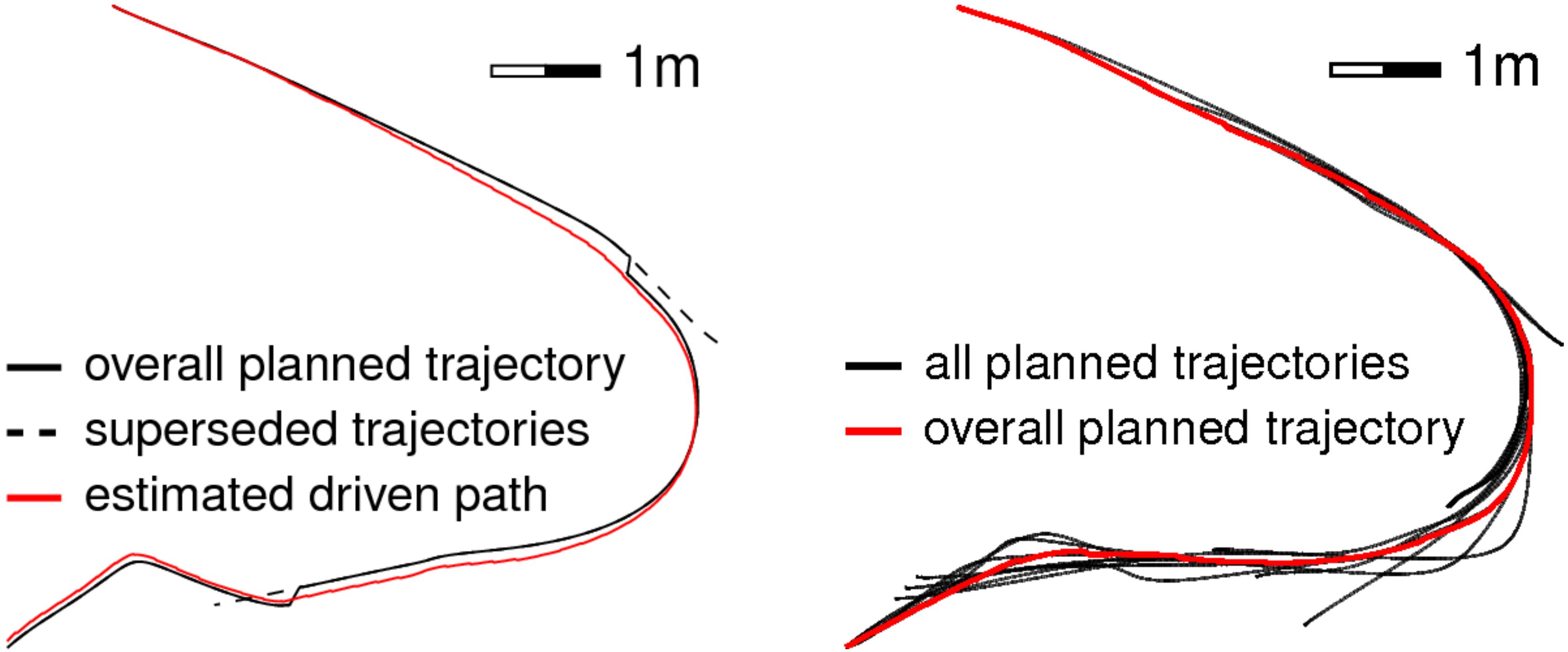
Procedure



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

1 position at re-planning time
2 assumed position at start time of new plan

Transportation Task – Replanning

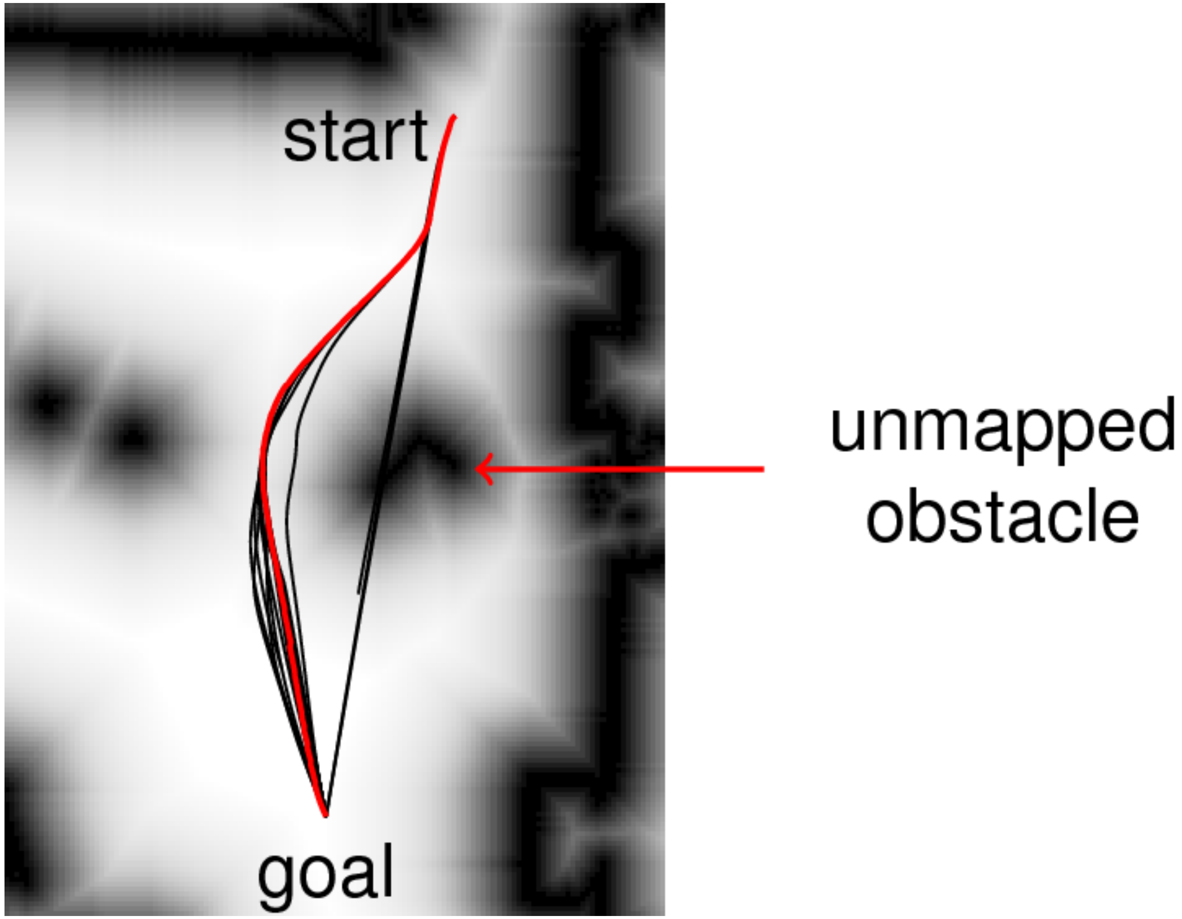


(a) reduced updating

(b) frequent updating, every 1.65s

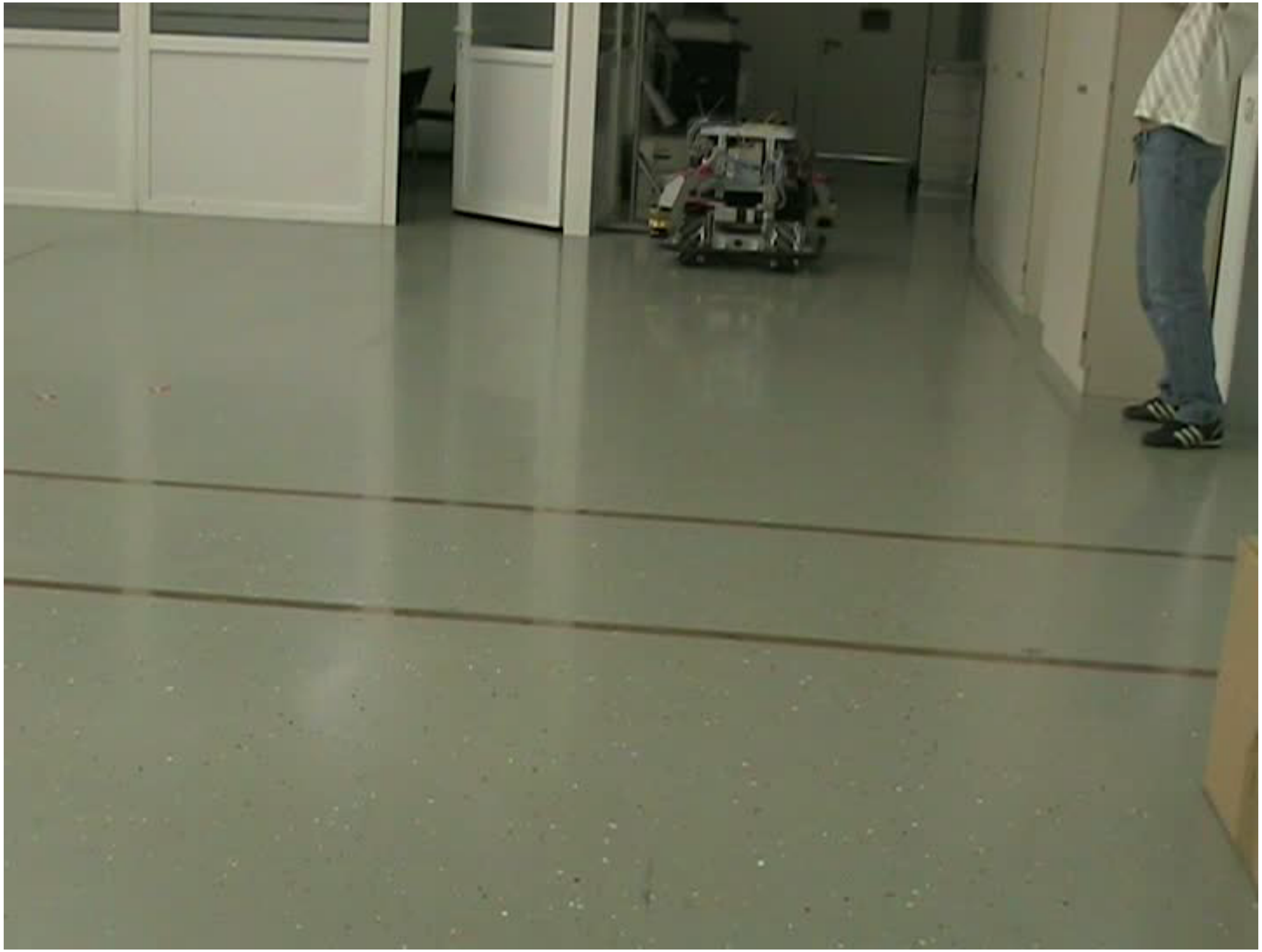
- Replanning also accounts for odometry drift, localization error

Unmapped Obstacle

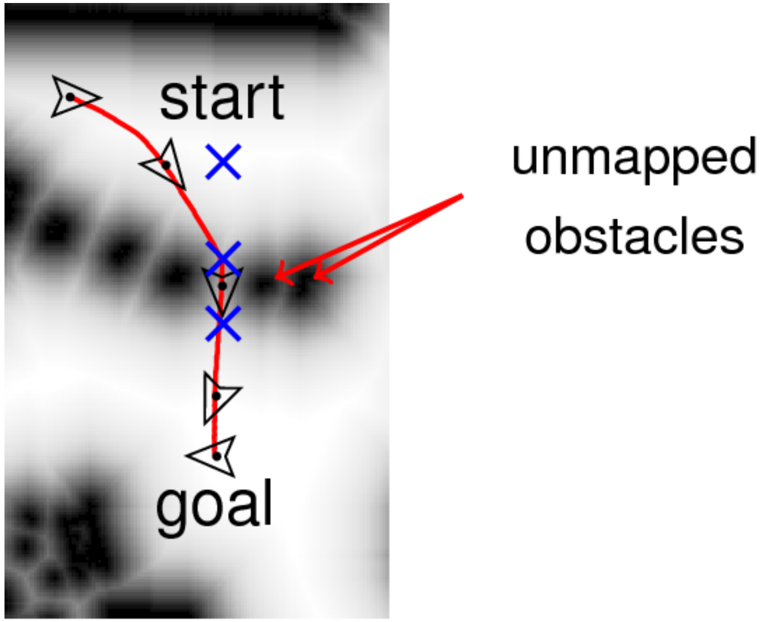


Planned trajectories over distance map of environment

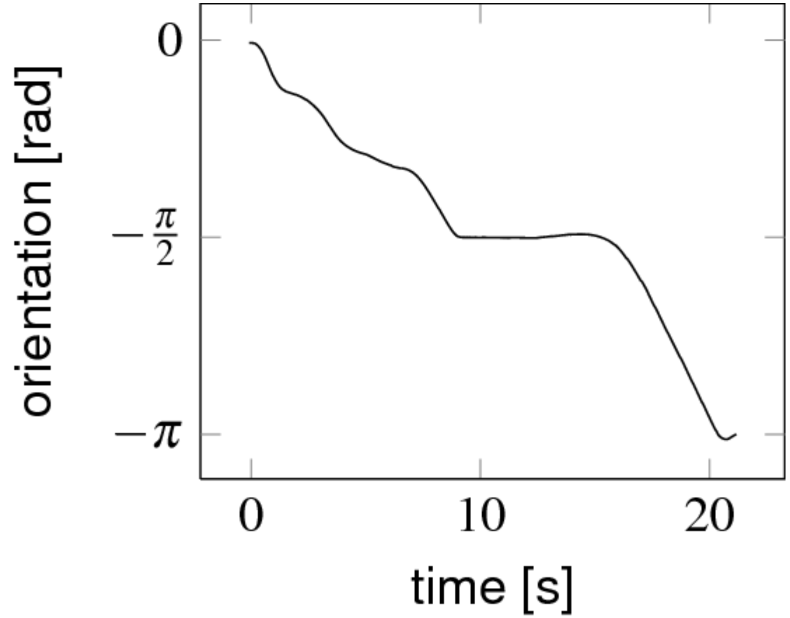
Unmapped Obstacle



Narrow Passage



(a) driven path over distance map



(b) orientation of driven path

- Passage width 120 cm
- Robot width 86 cm,
- Safety margin 20 cm,
- Discretization margin 5 cm
- Corridor width: 9 cm
- Manually supplied waypoints

Narrow Passage

Conclusion

- Our approach effectively restricts search/planning space
  - starts from initial path from a waypoint planner (which determines the topology of the path)
  - decouples path shape and velocities
  - Uses novel path representation (with a small set of meaningful parameters)
- Optimization w.r.t. cost function
  - Not just any path but good trajectory

Conclusion

- Omni-directional trajectory modeling
  - Represents orientational component
  - Enables optimization starting from initial paths (guaranteed solution)
- Experimental results
  - Precise tracking of trajectories
  - Low prediction error
  - Unmapped obstacles
  - Plan longer trajectories by curvature continuous stitching

Thank You for Your Attention!