Fast Reactive Risk-aware Motion Planning

Probabilistic Chekov

Sylvia Dai

February, 2018

Outline

Outline

Motivation

Problem Statement
- Definitions
- Assumptions

Approach
- Chekov Roadmap + TrajOpt
- LQG-MP
- System Diagram

Potential Extensions

Limitations of Current Planners

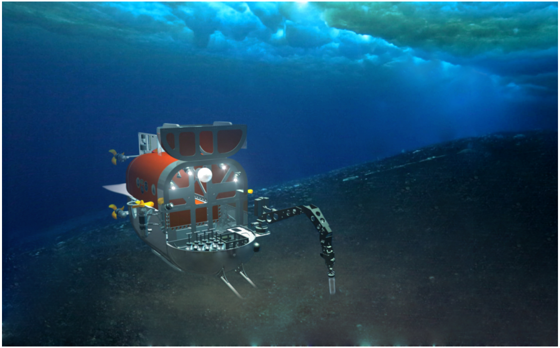
- Many traditional motion planners assume deterministic motions and full state knowledge

Limitations of Current Planners

- Many traditional motion planners assume deterministic motions and full state knowledge
- Most risk-aware planners are limited to low-DOF robots, simple (convex) environments

Motivation

Underwater Manipulation



Motivation

Underwater Manipulation

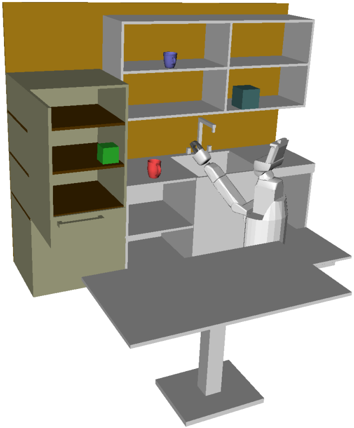
- Risk: currents, inner waves, vortices, sensor noises
- Optimality: limited battery and time

Motivation

Human Support Robot

Motivation

Human Support Robot



Solution

Probabilistic Chekov (p-Chekov)

- Fast reactive to severe plan disturbances that necessitate plan adjustment
- Risk-aware: under small disturbances, failure rate is guaranteed to be within a user-specified risk bound
- Account for process noises and observation noises
- Solutions are locally optimal or near-optimal according to a specified objective function

Outline

Model Definitions

Variables

- Robot State Space: $\mathcal{X} = \mathbb{R}^{n_x}$
- Control Input Space: $\mathcal{U} = \mathbb{R}^{n_u}$
- Discretized Time Series: $t = 0, 1, 2, \ldots, T$
- Fixed Time Interval: ΔT
- Robot State at time step t: $\mathbf{x}_t \in \mathcal{X}$
- Control Input at time step t: $\mathbf{u}_t \in \mathcal{U}$
- Measurement at time step t: $\mathbf{z}_t$

Model Definitions

Variables

- Robot State Space: $\mathcal{X} = \mathbb{R}^{n_x}$
- Control Input Space: $\mathcal{U} = \mathbb{R}^{n_u}$
- Discretized Time Series: $t = 0, 1, 2, \ldots, T$
- Fixed Time Interval: ΔT
- Robot State at time step t: $\mathbf{x}_t \in \mathcal{X}$
- Control Input at time step t: $\mathbf{u}_t \in \mathcal{U}$
- Measurement at time step t: $\mathbf{z}_t$

System Dynamics and Observation Model

$$\begin{aligned} \mathbf{x}_t &= f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{m}_t), & \mathbf{m}_t &\sim \mathcal{N}(0, M_t) \\ \mathbf{z}_t &= h(\mathbf{x}_t, \mathbf{n}_t), & \mathbf{n}_t &\sim \mathcal{N}(0, N_t) \end{aligned} \tag{1}$$

Model Definitions

Initial and Goal Conditions

- Initial Condition: $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}^{\mathrm{start}}, \mathbf{\Sigma}_{\mathbf{x}_0})$
- Goal Condition: a convex goal region $\mathcal{X}^{\mathrm{goal}}$

Model Definitions

Initial and Goal Conditions

- Initial Condition: $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}^{\text{start}}, \mathbf{\Sigma}_{\mathbf{x}_0})$
- Goal Condition: a convex goal region $\mathcal{X}^{\text{goal}}$

Trajectories

- Trajectory Π: $(\mathbf{x}_0^*, \mathbf{u}_0^*, \ldots, \mathbf{x}_T^*, \mathbf{u}_T^*)$, a sequence of nominal robot states and control inputs, satisfies the deterministic dynamics model $\mathbf{x}_t^* = f(\mathbf{x}_{t-1}^*, \mathbf{u}_{t-1}^*, 0)$ for $0 < t \leq T$.

Model Definitions

Initial and Goal Conditions

- Initial Condition: $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}^{\mathrm{start}}, \mathbf{\Sigma}_{\mathbf{x}_0})$
- Goal Condition: a convex goal region $\mathcal{X}^{\mathrm{goal}}$

Trajectories

- Trajectory Π: $(\mathbf{x}_0^*, \mathbf{u}_0^*, \ldots, \mathbf{x}_T^*, \mathbf{u}_T^*)$, a sequence of nominal robot states and control inputs, satisfies the deterministic dynamics model $\mathbf{x}_t^* = f(\mathbf{x}_{t-1}^*, \mathbf{u}_{t-1}^*, 0)$ for $0 < t \leq T$.

Objectives

- Objective Function: $J(\Pi)$

Constraint Definitions

No-collision Constraints

- A set of no-collision constraints for each obstacle:

$$C_i, \quad \forall i = 1, \ldots, N \tag{2}$$

Chance Constraints

- A joint chance constraint Δ_c:

$$P\left(\bigvee_{i=1}^{N} \overline{C_i}\right) \leq \Delta_c \tag{3}$$

Constraint Definitions

Temporal Constraints

- A temporal constraint τ:

$$T \times \Delta T \leq \tau \tag{4}$$

Goal State Constraints

- In the last time step of a trajectory, robot state should satisfy:

$$\mathbf{x}_T^* \to \mathcal{X}^{\mathrm{goal}} \tag{5}$$

Constraint Definitions

System Dynamics Constraints

- Robot states are within the robot state space $\mathcal{X}$, and the state transitions satisfy the deterministic system dynamics model:

$$\mathbf{x}_t^* = f(\mathbf{x}_{t-1}^*, \mathbf{u}_{t-1}^*, 0) \in \mathcal{X}, \quad \forall t = 1, \ldots, T \tag{6}$$

Control Input Constraints

- Control inputs are within the control input space $\mathcal{U}$:

$$\mathbf{u}_t^* \in \mathcal{U}, \quad \forall t = 1, \ldots, T \tag{7}$$

Problem Definition

$$
\begin{aligned}
\underset{\Pi}{\text{minimize}} \quad & J(\Pi) \\
\text{subject to} \quad & \mathbf{x}_0 \sim \mathcal{N}(\bar{\mathbf{x}}_0, \mathbf{\Sigma}_{\mathbf{x}_0}) \\
& \mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{m}_t), && 0 < t \leq T \\
& \mathbf{z}_t = h(\mathbf{x}_t, \mathbf{n}_t), && 0 < t \leq T \\
& \mathbf{m}_t \sim \mathcal{N}(0, M_t), && 0 < t \leq T \\
& \mathbf{n}_t \sim \mathcal{N}(0, N_t), && 0 < t \leq T \\
& \mathbf{x}_t \in \mathcal{X}, && 0 < t \leq T \\
& \mathbf{u}_t \in \mathcal{U}, && 0 < t \leq T \\
& \mathbf{x}_T^* \to \mathcal{X}^{\text{goal}} \\
& P\left(\bigvee_{i=1}^{N} \overline{C_i}\right) \leq \Delta_c \\
& T \times \Delta T \leq \tau
\end{aligned} \tag{8}
$$

Problem Definition

Input

- Robot state space: $\mathcal{X}$
- Control input space: $\mathcal{U}$
- System dynamics model: $\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{m}_t)$
- Observation model: $\mathbf{z}_t = h(\mathbf{x}_t, \mathbf{n}_t)$
- Covariance matrix of process noise: M_t
- Covariance matrix of observation noise: N_t
- Environment model containing obstacles
- Initial condition: $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}^{\text{start}}, \mathbf{\Sigma}_{\mathbf{x}_0})$
- Convex goal region: $\mathcal{X}^{\text{goal}}$
- Objective function: $J(\Pi)$
- Temporal constraint: τ
- Chance constraint: Δ_c

Problem Definition

Output

- A valid trajectory Π that is locally optimal or near optimal

Assumptions

- Collision environments are practical, not overly complex.

Assumptions

- Collision environments are practical, not overly complex.
- Both controller uncertainties (process noises) and sensor uncertainties (observation noises) have Gaussian distribution.

Assumptions

- Collision environments are practical, not overly complex.
- Both controller uncertainties (process noises) and sensor uncertainties (observation noises) have Gaussian distribution.
- Both the system dynamics model and observation model are either linear or can be well approximated locally by its linearization.

Outline

Chekov Roadmap + TrajOpt

- Roadmap represents static collision-free space: reused across planning instances

Chekov Roadmap + TrajOpt

- Roadmap represents static collision-free space: reused across planning instances
- Sparse roadmap: fast queries, but solutions might be sub-optimal

Chekov Roadmap + TrajOpt

- Roadmap represents static collision-free space: reused across planning instances
- Sparse roadmap: fast queries, but solutions might be sub-optimal
- TrajOpt: high failure rate when provided naïve straight-line seed

Chekov Roadmap + TrajOpt

- Roadmap represents static collision-free space: reused across planning instances
- Sparse roadmap: fast queries, but solutions might be sub-optimal
- TrajOpt: high failure rate when provided naïve straight-line seed
- Roadmap + TrajOpt: pass roadmap solutions as seed trajectories to TrajOpt
- The combined Chekov planner: fast reactive to disturbances, low failure rate

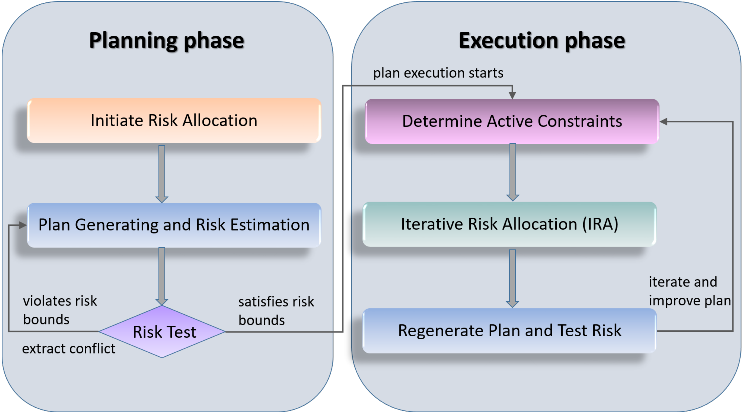
Linear-quadratic Gaussian Motion Planning

- Kalman filter combined with linear-quadratic regulator (LQR)

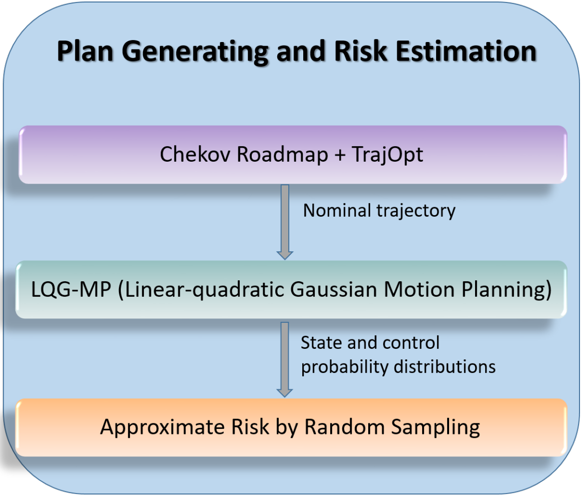
Linear-quadratic Gaussian Motion Planning

- Kalman filter combined with linear-quadratic regulator (LQR)
- Input: *a priori* probability distributions (Gaussian) of sensors and controllers
- Output: *a priori* probability distributions of robot states and control inputs for a given path

Approach


Planning phase
Initiate Risk Allocation
Plan Generating and Risk Estimation
violates risk bounds
Risk Test
satisfies risk bounds
extract conflict
Execution phase
plan execution starts
Determine Active Constraints
Iterative Risk Allocation (IRA)
Regenerate Plan and Test Risk
iterate and improve plan

Approach



Plan Generating and Risk Estimation
Chekov Roadmap + TrajOpt
Nominal trajectory
LQG-MP (Linear-quadratic Gaussian Motion Planning)
State and control
probability distributions
Approximate Risk by Random Sampling

Outline

Potential Extensions

- Semantic-information-guided risk allocation

Potential Extensions

- Semantic-information-guided risk allocation
- Incorporate non-Gaussian noises with Moment-Sum-Of-Squares-based state estimation method

Potential Extensions

- Semantic-information-guided risk allocation
- Incorporate non-Gaussian noises with Moment-Sum-Of-Squares-based state estimation method
- Combine with stochastic roadmaps to consider environmental uncertainties

Potential Extensions

- Semantic-information-guided risk allocation
- Incorporate non-Gaussian noises with Moment-Sum-Of-Squares-based state estimation method
- Combine with stochastic roadmaps to consider environmental uncertainties
- Integrate with activity planners through temporal constraints

Summary

P-Chekov vs TrajOpt

- P-Chekov is a risk-aware global planning and execution system; TrajOpt is its local trajectory optimization part
- P-Chekov learns from previous planning trials and add configuration penalties to TrajOpt

Summary

P-Chekov vs TrajOpt

- P-Chekov is a risk-aware global planning and execution system; TrajOpt is its local trajectory optimization part
- P-Chekov learns from previous planning trials and add configuration penalties to TrajOpt

P-Chekov vs LQG-MP

- P-Chekov is a stand-alone real-time risk-aware planner. It finds a solution that satisfies the risk bound.
- LQG-MP is a path selection method. It selects the best path from candidate paths generated by other motion planners in order to minimize risk.

Summary

P-Chekov vs Other Risk-aware Planners

- P-Chekov can solve the planning tasks with high-dimensional robot, 3D non-convex environment that current risk-aware planners can't solve in real time
- P-Chekov can incorporate differential constraints such as torque and velocity limits
- P-Chekov improves the plan during execution (any-time planning) and can react to disturbances quickly

For Further Reading

John Schulman, Jonathan Ho, Alex X Lee, Ibrahim Awwal, Henry Bradlow, and Pieter Abbeel.
Finding locally optimal, collision-free trajectories with sequential convex optimization.
In *Robotics: science and systems*, volume 9, pages 1–10, 2013.

Jur Van Den Berg, Pieter Abbeel, and Ken Goldberg.
LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information.
The International Journal of Robotics Research, 30(7):895–913, 2011.