Efficient Collision Modelling for Numerical Optimal Control

Final Presentation of Master’s Thesis Project

Christian Dietz

April 5, 2022

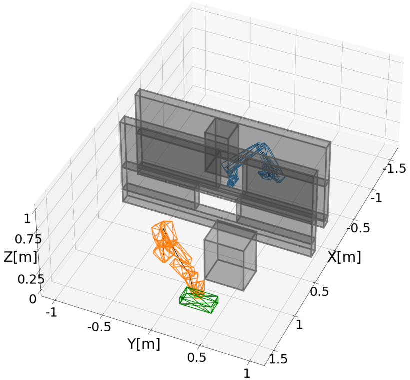
Background

- University: Technical University of Munich (TUM)
- Study course: Master Mathematics in Science and Engineering
- Supervisor: Prof. Dr. Michael Ulbrich (TUM)
- Advisors: Dr. Sebastian Albrecht (Siemens),
  Armin Nurkanović (University of Freiburg)

SIEMENS

Motivational example

Kuka LBR iiwa 7 & Ur5 execute an item passing task



SIEMENS

Contribution of the master's thesis

Master's thesis is based on:

X. Zhang, A. Liniger, and F. Borrelli. **Optimization-based collision avoidance.** *IEEE Transactions on Control Systems Technology*, 29:972-983, 2021.

Contributions:

- Derivation of different forms of collision constraints
- Performance evaluation of the constraints through numerical experiments
- Application to manipulator models

SIEMENS

Outline

1. Optimal Control Problem
2. Derivation of Collision Constraints
3. Numerical Benchmarks
4. Trajectory Planning for Manipulators: Methodology and Challenges

Outline

1. Optimal Control Problem
2. Derivation of Collision Constraints
3. Numerical Benchmarks
4. Trajectory Planning for Manipulators: Methodology and Challenges

Starting point is a dynamic model of a robot

Dynamic equations of motion:

$$\tau = M(q)\ddot{q} + C(q, \dot{q}) + G(q)$$

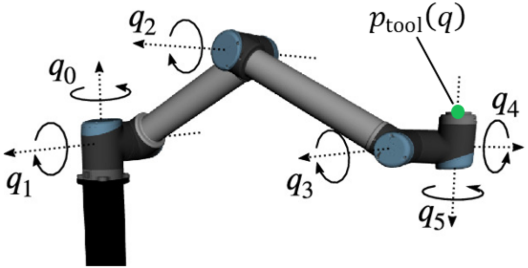
where

q_i - joint angles

$\dot{q}_i$ - joint velocities

$\ddot{q}_i$ - joint accelerations

τ_i - torques



From: https://github.com/giusenso/robot-manipulator-dynamics

Starting point is a dynamic model of a robot

Rewrite as first-order ODE

$$\frac{d}{dt}\begin{pmatrix} q \\ \dot{q} \end{pmatrix} = \begin{pmatrix} \dot{q} \\ M(q)^{-1}(\tau - C(q,\dot{q}) - G(q)) \end{pmatrix}$$

set

$$z = (q, \dot{q}), \quad u = \tau$$

Compact form:

$$\dot{z} = f(z, u)$$

Starting point is a dynamic model of a robot

Rewrite as first-order ODE

$$\frac{d}{dt}\begin{pmatrix} q \\ \dot{q} \end{pmatrix} = \begin{pmatrix} \dot{q} \\ M(q)^{-1}(\tau - C(q,\dot{q}) - G(q)) \end{pmatrix}$$

set

$$z = (q, \dot{q}), \quad u = \tau$$

Compact form:

$$\dot{z} = f(z, u)$$

Alternatively:

Double integrator dynamic

$$\frac{d}{dt}\begin{pmatrix} q \\ \dot{q} \end{pmatrix} = \begin{pmatrix} \dot{q} \\ \ddot{q} \end{pmatrix}$$

set

$$z = (q, \dot{q}), \quad u = \ddot{q}$$

Compact form:

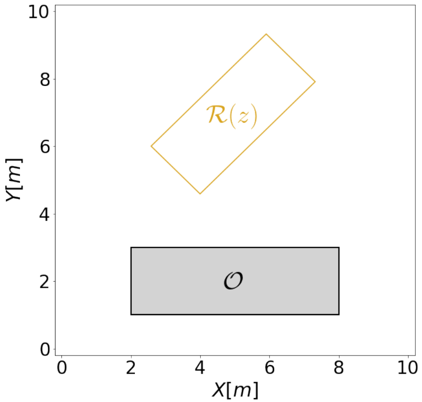
$$\dot{z} = f(z, u)$$

A discrete optimal control problem (OCP)

$$
\begin{aligned}
\min_{z,u,\lambda,\mu} \quad & \sum_{k=0}^{N-1} L(z_k, u_k) + M(z_N) & & \text{(objective term)} \\
\text{s.t.} \quad & z_0 = \bar{z}_0, & & \text{(initial condition)} \\
& p_{\text{tool}}(z_N) = \bar{p}_N, & & \text{(terminal constraint)} \\
& z_{k+1} = F(z_k, u_k), \quad k = 0, \dots, N-1, & & \text{(dynamical conditions)} \\
& g_{\text{Path}}(z_k, u_k) \leq 0, \quad k = 0, \dots, N-1, & & \text{(path constraints)} \\
& g_{\text{Coll}}(z_k, \lambda_k, \mu_k) \leq 0, \quad k = 0, \dots, N-1. & & \text{(collision constraints)}
\end{aligned}
$$

Outline

Squared distance function: Definition



Squared distance between robot $\mathcal{R}(z)$ and obstacle $\mathcal{O}$:

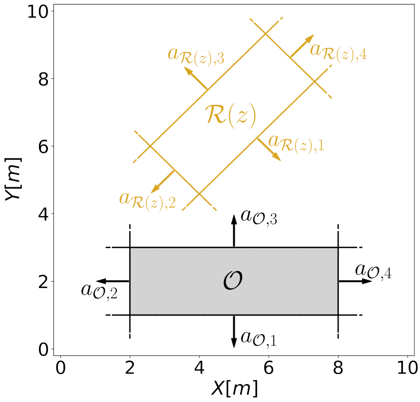
$$\begin{aligned} \Delta^2(\mathcal{R}(z), \mathcal{O}) := \min_{x,y} \quad & \|x - y\|_2^2 \\ \text{s.t.} \quad & x \in \mathcal{R}(z), \\ & y \in \mathcal{O}. \end{aligned}$$

Theoretically: Enforce

$$\Delta^2(\mathcal{R}(z_k), \mathcal{O}) \geq \Delta^2_{\min}, \quad k = 1, \ldots, N,$$

in the discrete OCP.

Squared distance function: Half-space representation

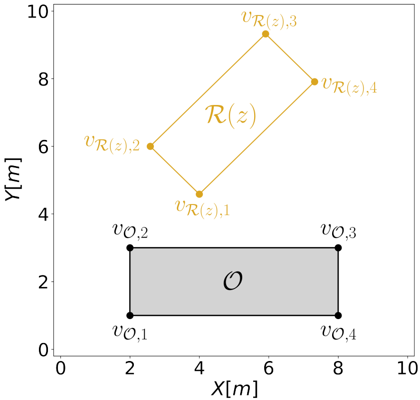


$$A_{\mathcal{R}(z)} = \begin{bmatrix} a^T_{\mathcal{R}(z),1} \\ a^T_{\mathcal{R}(z),2} \\ a^T_{\mathcal{R}(z),3} \\ a^T_{\mathcal{R}(z),4} \end{bmatrix} \in \mathbb{R}^{4\times 2}, \ b_{\mathcal{R}(z)} = \begin{bmatrix} b_{\mathcal{R}(z),1} \\ b_{\mathcal{R}(z),2} \\ b_{\mathcal{R}(z),3} \\ b_{\mathcal{R}(z),4} \end{bmatrix} \in \mathbb{R}^4,$$

$A_{\mathcal{O}} \in \mathbb{R}^{4\times 2}$ and $b_{\mathcal{O}} \in \mathbb{R}^4$ similarly structured.

$$\begin{aligned} \Delta^2(\mathcal{R}(z), \mathcal{O}) = \min_{x,y} \quad & \|x - y\|_2^2 \\ \text{s.t.} \quad & A_{\mathcal{R}(z)} x \leq b_{\mathcal{R}(z)}, \\ & A_{\mathcal{O}} y \leq b_{\mathcal{O}}. \end{aligned}$$

Squared distance function: Vertex representation



$$V_{\mathcal{R}(z)} = \begin{bmatrix} v_{\mathcal{R}(z),1} & v_{\mathcal{R}(z),2} & v_{\mathcal{R}(z),3} & v_{\mathcal{R}(z),4} \end{bmatrix} \in \mathbb{R}^{2\times 4},$$

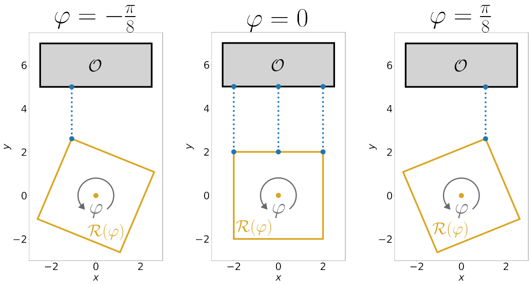
$V_{\mathcal{O}} \in \mathbb{R}^{2\times 4}$ similarly structured.

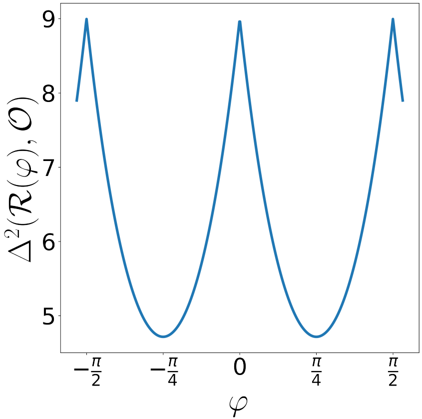
$$\begin{aligned} \Delta^2(\mathcal{R}(z), \mathcal{O}) = \min_{\theta,\eta} \quad & \|V_{\mathcal{R}(z)}\theta - V_{\mathcal{O}}\eta\|_2^2 \\ \text{s.t.} \quad & 1^T\theta = 1,\ 1^T\eta = 1, \\ & \theta \geq 0,\ \eta \geq 0. \end{aligned}$$

SIEMENS

The squared distance function is not differentiable

- $\Delta^2(\mathcal{R}(\varphi), \mathcal{O})$: squared distance between square and rectangle
- square depends on orientation φ





SIEMENS

Resolving non-differentiability[1]: Deriving the dual problem

Primal distance problem:

$$
\begin{aligned}
\min_{x,y,w} \quad & \|w\|_2^2 \\
\text{s.t.} \quad & w = x - y, \\
& A_{\mathcal{R}(z)} x \leq b_{\mathcal{R}(z)}, \\
& A_{\mathcal{O}} y \leq b_{\mathcal{O}}.
\end{aligned}
$$

Dual function:

$$
\begin{aligned}
d_z(\nu, \lambda, \mu) &= \inf_{x,y,w} L_z(x, y, w, \nu, \lambda, \mu) \\
&= \inf_{x,y,w} \Big(\|w\|_2^2 + \nu^T (x - y - w) \\
&\quad + \lambda^T (A_{\mathcal{R}(z)} x - b_{\mathcal{R}(z)}) + \mu^T (A_{\mathcal{O}} y - b_{\mathcal{O}}) \Big)
\end{aligned}
$$

[1] X. Zhang, A. Liniger, and F. Borrelli. Optimization-based collision avoidance. *IEEE Transactions on Control Systems Technology*, 29:972- 983, 2021.

SIEMENS

Resolving non-differentiability[1]: Deriving the dual problem

Primal distance problem:

$$
\begin{aligned}
\min_{x,y,w} \quad & \|w\|_2^2 \\
\text{s.t.} \quad & w = x - y, \\
& A_{\mathcal{R}(z)} x \leq b_{\mathcal{R}(z)}, \\
& A_{\mathcal{O}} y \leq b_{\mathcal{O}}.
\end{aligned}
$$

Dual function:

$$
\begin{aligned}
d_z(\nu, \lambda, \mu) &= \inf_{x,y,w} L_z(x, y, w, \nu, \lambda, \mu) \\
&= \inf_{x,y,w} \big(\|w\|_2^2 + \nu^T(x - y - w) \\
&\quad + \lambda^T(A_{\mathcal{R}(z)} x - b_{\mathcal{R}(z)}) + \mu^T(A_{\mathcal{O}} y - b_{\mathcal{O}})\big) \\
&= \inf_{w} \big(\|w\|_2^2 - \nu^T w\big) + \inf_{x} \big((\lambda^T A_{\mathcal{R}(z)} + \nu^T)x\big) \\
&\quad + \inf_{y} \big((\mu^T A_{\mathcal{O}} - \nu^T)y\big) - \lambda^T b_{\mathcal{R}(z)} - \mu^T b_{\mathcal{O}}
\end{aligned}
$$

[1] X. Zhang, A. Liniger, and F. Borrelli. Optimization-based collision avoidance. *IEEE Transactions on Control Systems Technology*, 29:972- 983, 2021.

SIEMENS

Resolving non-differentiability[1]: Deriving the dual problem

Dual function:

$$d_z(\nu, \lambda, \mu) = \inf_w \left(\|w\|_2^2 - \nu^T w\right) + \inf_x \left((\lambda^T A_{\mathcal{R}(z)} + \nu^T)x\right) \\ + \inf_y \left((\mu^T A_{\mathcal{O}} - \nu^T)y\right) - \lambda^T b_{\mathcal{R}(z)} - \mu^T b_{\mathcal{O}}$$

Infima values:

$$\inf_w \left(\|w\|_2^2 - \nu^T w\right) = -\frac{1}{4}\|\nu\|_2^2,$$

$$\inf_x \left((\lambda^T A_{\mathcal{R}(z)} + \nu^T)x\right)$$

$$= \begin{cases} 0, & \text{if } A_{\mathcal{R}(z)}^T \lambda + \nu = 0, \\ -\infty, & \text{if } A_{\mathcal{R}(z)}^T \lambda + \nu \neq 0. \end{cases}$$

$\inf_y \left((\mu^T A_{\mathcal{O}} - \nu^T)y\right)$ similarly.

[1] X. Zhang, A. Liniger, and F. Borrelli. Optimization-based collision avoidance. *IEEE Transactions on Control Systems Technology*, 29:972- 983, 2021.

Resolving non-differentiability[1]: Deriving the dual problem

Dual problem:

$$\begin{aligned} \max_{\nu,\lambda,\mu} \quad & -\frac{1}{4}\|\nu\|_2^2 - \lambda^T b_{\mathcal{R}(z)} - \mu^T b_{\mathcal{O}} \\ \text{s.t.} \quad & A_{\mathcal{R}(z)}^T \lambda + \nu = 0, \\ & A_{\mathcal{O}}^T \mu - \nu = 0, \\ & \lambda \geq 0, \ \mu \geq 0. \end{aligned}$$

Reduced dual problem:

$$\begin{aligned} \max_{\lambda,\mu} \quad & -\frac{1}{4}\|A_{\mathcal{O}}^T \mu\|_2^2 - \lambda^T b_{\mathcal{R}(z)} - \mu^T b_{\mathcal{O}} \\ \text{s.t.} \quad & A_{\mathcal{R}(z)}^T \lambda + A_{\mathcal{O}}^T \mu = 0, \\ & \lambda \geq 0, \ \mu \geq 0. \end{aligned}$$

[1] X. Zhang, A. Liniger, and F. Borrelli. Optimization-based collision avoidance. *IEEE Transactions on Control Systems Technology*, 29:972- 983, 2021.

Resolving non-differentiability[1]: Deducing differentiable constraints

Dual distance constraints:

$$\Delta^2(\mathcal{R}(z_k), \mathcal{O}) \geq \Delta^2_{\min}$$

$$\Delta^2(\mathcal{R}(z_k), \mathcal{O}) :=$$

$$\begin{aligned} \max_{\lambda,\mu} \quad & -\frac{1}{4}\|A^T_{\mathcal{O}}\mu\|^2_2 - \lambda^T b_{\mathcal{R}(z_k)} - \mu^T b_{\mathcal{O}} \\ \text{s.t.} \quad & A^T_{\mathcal{R}(z)}\lambda + A^T_{\mathcal{O}}\mu = 0, \\ & \lambda \geq 0, \ \mu \geq 0. \end{aligned}$$

Differentiable distance constraints:

$$-\frac{1}{4}\|A^T_{\mathcal{O}}\mu\|^2_2 - \lambda^T b_{\mathcal{R}(z_k)} - \mu^T b_{\mathcal{O}} \geq \Delta^2_{\min},$$

$$A^T_{\mathcal{R}(z)}\lambda + A^T_{\mathcal{O}}\mu = 0,$$

$$\lambda \geq 0, \ \mu \geq 0.$$

[1] X. Zhang, A. Liniger, and F. Borrelli. Optimization-based collision avoidance. *IEEE Transactions on Control Systems Technology*, 29:972- 983, 2021.

SIEMENS

Different possible primal formulations; and resulting dual formulations

Polytopic Robot	$\mathcal{R}(z) = \{V_{\mathcal{R}(z)}\theta \mid 1^T\theta = 1, \theta \geq 0\}$	$\mathcal{R}(z) = \{x \mid A_{\mathcal{R}(z)}x \leq b_{\mathcal{R}(z)}\}$
Polytopic Obstacles	$\mathcal{O} = \{V_{\mathcal{O}}\eta \mid 1^T\eta = 1, \eta \geq 0\}$ *(V-Pol)*[1]	$\mathcal{O} = \{y \mid A_{\mathcal{O}}y \leq b_{\mathcal{O}}\}$ *(H-Pol)*[2]

[1] New contributions

[2] X. Zhang, A. Liniger, and F. Borrelli. Optimization-based collision avoidance. *IEEE Transactions on Control Systems Technology*, 29:972- 983, 2021.

SIEMENS

Different possible primal formulations; and resulting dual formulations

Polytopic Robot	$\mathcal{R}(z) = \{V_{\mathcal{R}(z)}\theta \mid 1^T\theta = 1, \theta \geq 0\}$	$\mathcal{R}(z) = \{x \mid A_{\mathcal{R}(z)}x \leq b_{\mathcal{R}(z)}\}$
Polytopic Obstacles	$\mathcal{O} = \{V_{\mathcal{O}}\eta \mid 1^T\eta = 1, \eta \geq 0\}$ *(V-Pol)*[1]	$\mathcal{O} = \{y \mid A_{\mathcal{O}}y \leq b_{\mathcal{O}}\}$ *(H-Pol)*[2]
Ellipsoidal Obstacles	$\mathcal{O} = \{y \mid \hat{A}_{\mathcal{O}}y \preceq_{\mathcal{K}_{\text{so}}} \hat{b}_{\mathcal{O}}\}$ *(V-GI)*[1]	$\mathcal{O} = \{y \mid \hat{A}_{\mathcal{O}}y \preceq_{\mathcal{K}_{\text{so}}} \hat{b}_{\mathcal{O}}\}$ *(H-GI)*[2]

[1] New contributions

[2] X. Zhang, A. Liniger, and F. Borrelli. Optimization-based collision avoidance. *IEEE Transactions on Control Systems Technology*, 29:972- 983, 2021.

Different possible primal formulations; and resulting dual formulations

Polytopic Robot	$\mathcal{R}(z) = \{V_{\mathcal{R}(z)}\theta \mid 1^T\theta = 1, \theta \geq 0\}$	$\mathcal{R}(z) = \{x \mid A_{\mathcal{R}(z)}x \leq b_{\mathcal{R}(z)}\}$
Polytopic Obstacles	$\mathcal{O} = \{V_{\mathcal{O}}\eta \mid 1^T\eta = 1, \eta \geq 0\}$ *(V-Pol)*[1]	$\mathcal{O} = \{y \mid A_{\mathcal{O}}y \leq b_{\mathcal{O}}\}$ *(H-Pol)*[2]
Ellipsoidal Obstacles	$\mathcal{O} = \{y \mid \hat{A}_{\mathcal{O}}y \preceq_{\mathcal{K}_{\text{so}}} \hat{b}_{\mathcal{O}}\}$ *(V-GI)*[1]	$\mathcal{O} = \{y \mid \hat{A}_{\mathcal{O}}y \preceq_{\mathcal{K}_{\text{so}}} \hat{b}_{\mathcal{O}}\}$ *(H-GI)*[2]
Ellipsoidal Obstacles	$\mathcal{O} = \{y \mid (y - c_{\mathcal{O}})^T P_{\mathcal{O}}(y - c_{\mathcal{O}}) \leq 1\}$ *(V-Sqrt)*[1], *(V-MD)*[1]	$\mathcal{O} = \{y \mid (y - c_{\mathcal{O}})^T P_{\mathcal{O}}(y - c_{\mathcal{O}}) \leq 1\}$ *(H-Sqrt)*[1], *(H-Sqrt+L)*[1], *(H-MD)*[1]

[1] New contributions

[2] X. Zhang, A. Liniger, and F. Borrelli. Optimization-based collision avoidance. *IEEE Transactions on Control Systems Technology*, 29:972- 983, 2021.

SIEMENS

Different possible primal formulations; and resulting dual formulations

Additional variables/constraints - Scaling w.r.t. number (#) of vertices / half-spaces:

Formulation Type	# Optimization Variables	# Inequality Constraints	# Equality Constraints
"V-"	Constant	Linear	None
"H-"	Linear	Linear	Constant / None

SIEMENS

Outline

1. Optimal Control Problem
2. Derivation of Collision Constraints
3. Numerical Benchmarks
4. Trajectory Planning for Manipulators: Methodology and Challenges

Computational framework

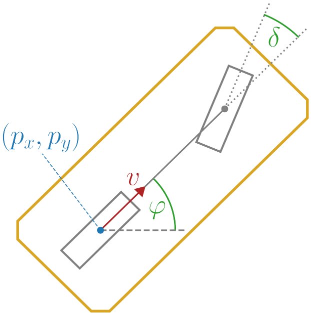
- Code is implemented in **Python**
- OCP problems are formulated via **CasADi**[1]
- Numerical solver used is **IPOPT**[2]

[1] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. CasADi - A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11:1-36, 2019.

[2] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25-57, 2006.

Considered robotic systems for benchmark problems

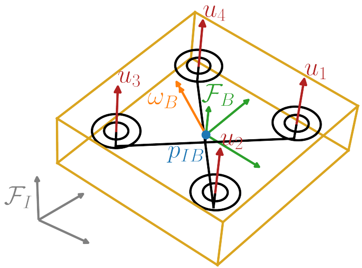
Bicycle-based Car Model:



State variables: $z = (p_x, p_y, \varphi, v, \dot{v}, \delta)$

Control variables: $u = (\ddot{v}, \dot{\delta})$

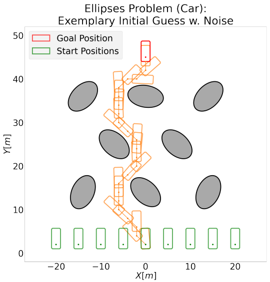
Drone Model:

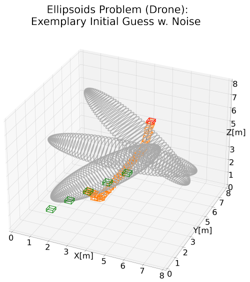


State variables: $z = (p_{IB}, \dot{p}_{IB}, q_{IB}, \omega_B)$

Control variables: $u = (u_1, u_2, u_3, u_4)$

Initial guesses generated by an A* algorithm





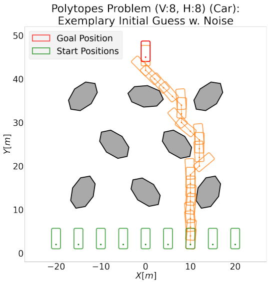
- A* samples through nodes in discrete space
- Next node to explore is chosen based on lowest cost
- cost $\hat{=}$ path length so far + remaining path length estimation (lower bound)

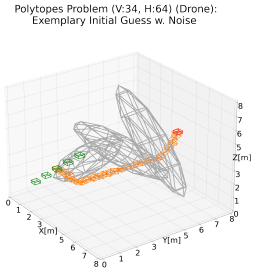
$\Rightarrow$ A* provides initial guess for positions

- Initializing orientation with noise provides a **family of initial guesses**

SIEMENS

Initial guesses generated by an A* algorithm





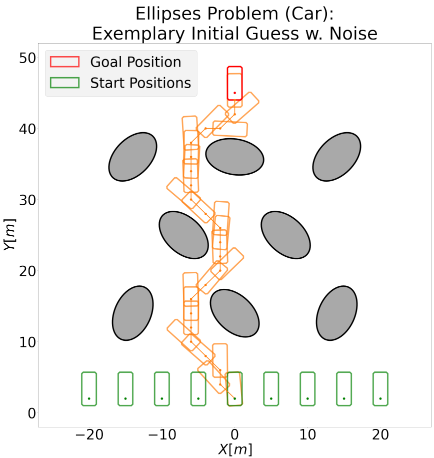
- A^* samples through nodes in discrete space
- Next node to explore is chosen based on lowest cost
- cost $\hat{=}$ path length so far + remaining path length estimation (lower bound)

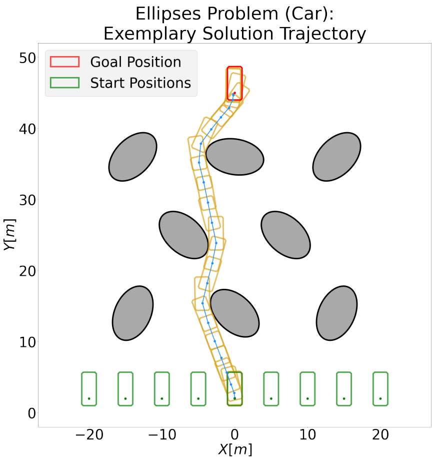
$\Rightarrow$ A^* provides initial guess for positions

- Initializing orientation with noise provides a **family of initial guesses**

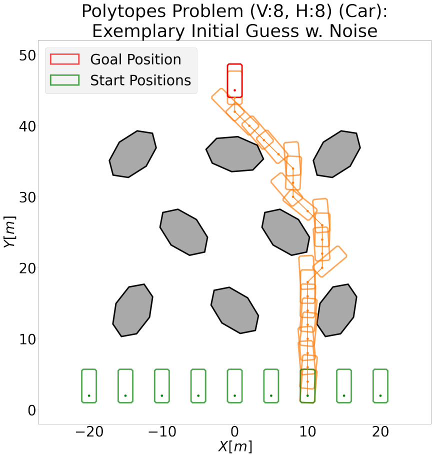
SIEMENS

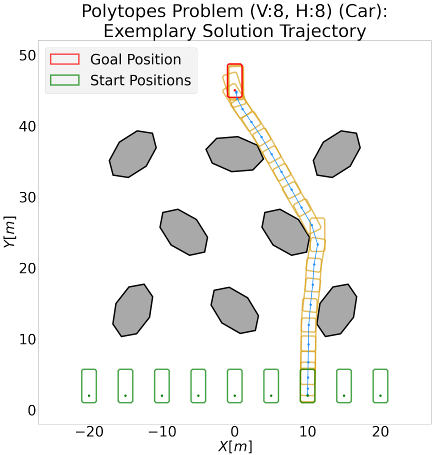
Example solutions of the benchmark problems





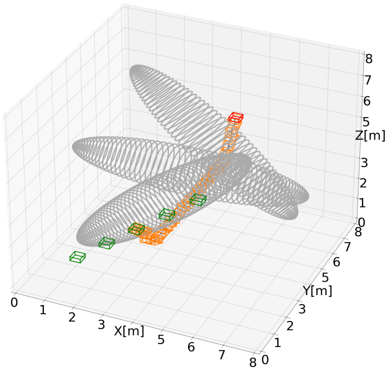
Example solutions of the benchmark problems



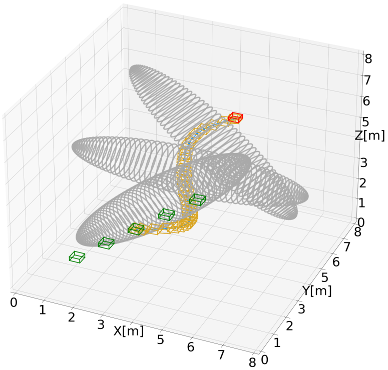


Example solutions of the benchmark problems

Ellipsoids Problem (Drone):
Exemplary Initial Guess w. Noise

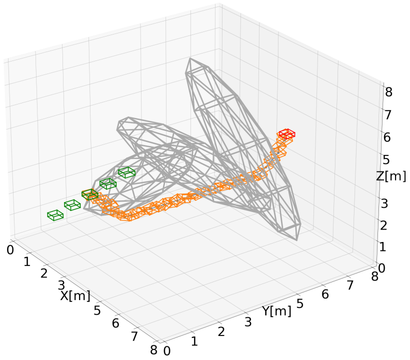


Ellipsoids Problem (Drone):
Exemplary Solution Trajectory

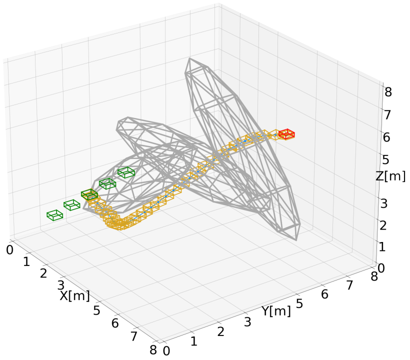


Example solutions of the benchmark problems

Polytopes Problem (V:34, H:64) (Drone):
Exemplary Initial Guess w. Noise

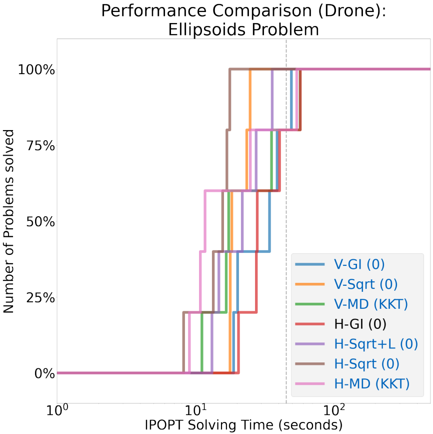


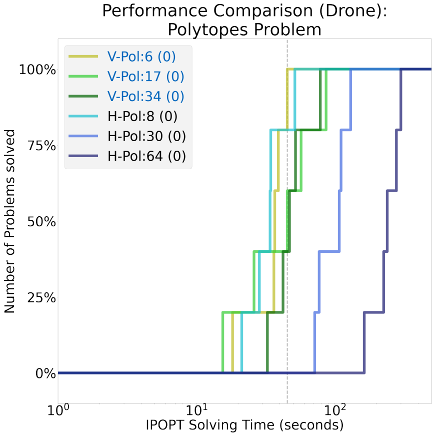
Polytopes Problem (V:34, H:64) (Drone):
Exemplary Solution Trajectory



SIEMENS

Computational performance comparison

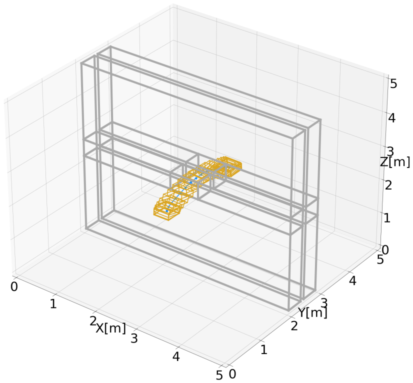




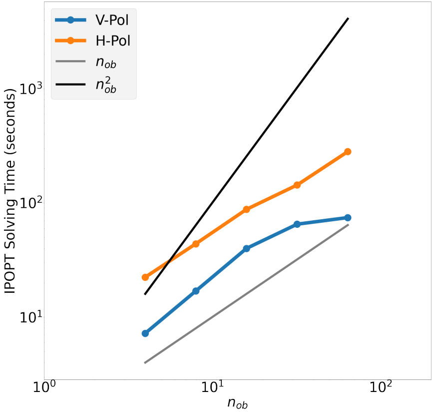
SIEMENS

Scaling w.r.t. number of obstacles

Exemplary Problem



Computation Time according to Number of Obstacles

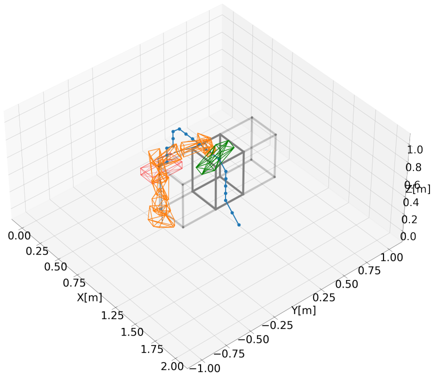


Outline

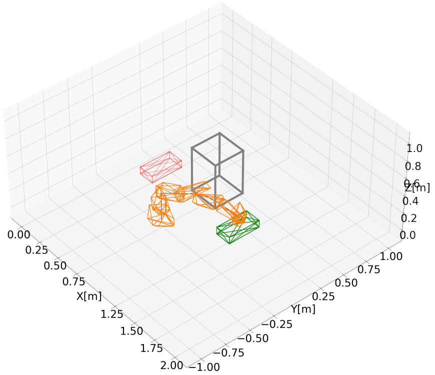
1. Optimal Control Problem
2. Derivation of Collision Constraints
3. Numerical Benchmarks
4. Trajectory Planning for Manipulators: Methodology and Challenges

Generating initial guesses

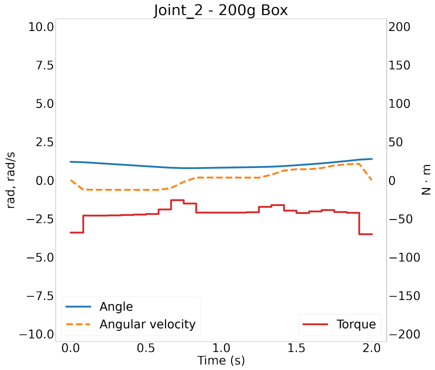
Solving Inverse Kinematics for A-star points:
Solve $p_{\text{tool}}(q) = p_{\text{Astar},\,i}$

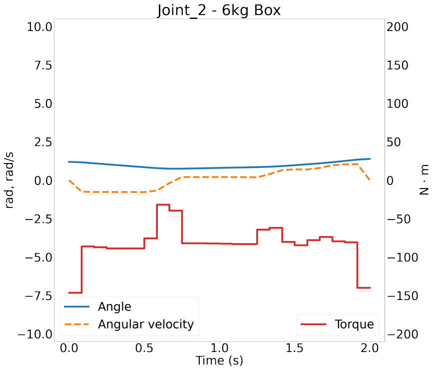


Solving with Full Dynamic:
Box weighs 200g



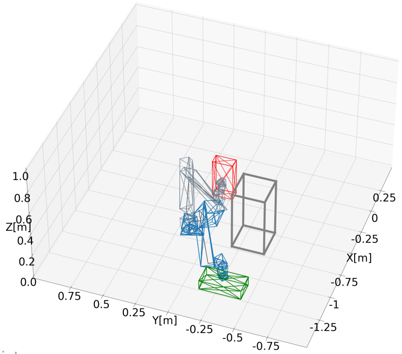
Using the "full" dynamic yields realistic controls



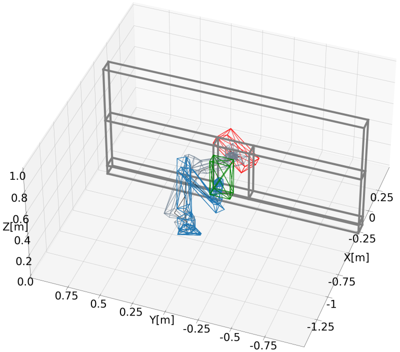


Motivational example revisited: Each manipulator has four phases

Ur5 - Phase 1
Checking Collision for Parts: 3-8
Checking 10 Self-Collisions



Ur5 - Phase 2
Checking Collision for Parts: 4-8
Checking 0 Self-Collisions



SIEMENS

Summary

- Using problem specific formulations increases computational performance
  - $\rightarrow$ Vertex representation for polytopes
  - $\rightarrow$ Quadratic inequality representation for ellipsoids
- Representing polytopes through vertices increases computational performance especially for intricate polytopes
- Considered constraints are applicable together with different robotic systems:
  - $\rightarrow$ Car model
  - $\rightarrow$ Drone model
  - $\rightarrow$ Manipulator models

References

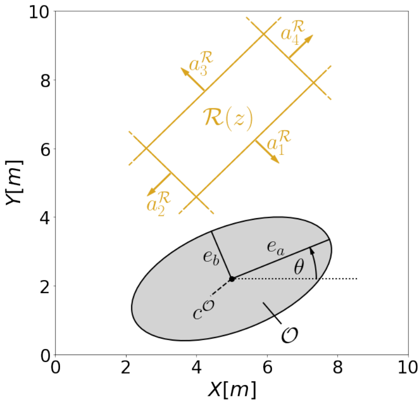
[1] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. **CasADi – A software framework for nonlinear optimization and optimal control.** *Mathematical Programming Computation*, 11:1–36, 2019.

[2] L. M. G. Johannessen, M. H. Arbo, and J. T. Gravdahl. **Robot dynamics with URDF & CasADi.** In *2019 7th International Conference on Control, Mechatronics and Automation*. IEEE, 2019.

[3] A. Wächter and L. T. Biegler. **On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming.** *Mathematical Programming*, 106:25–57, 2006.

[4] X. Zhang, A. Liniger, and F. Borrelli. **Optimization-based collision avoidance.** *IEEE Transactions on Control Systems Technology*, 29:972–983, 2021.

Representation for ellipsoidal obstacles



Let

$$P_{\mathcal{O}} = R(\theta) \begin{pmatrix} \frac{1}{e_a^2} & 0 \\ 0 & \frac{1}{e_b^2} \end{pmatrix} R(\theta)^T$$

where

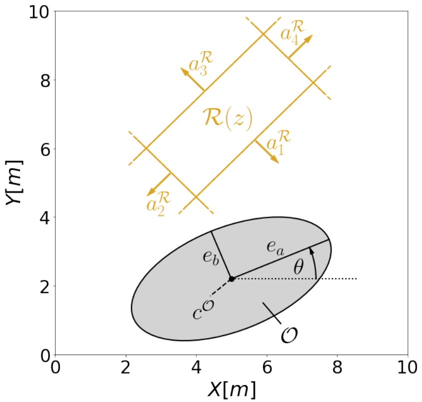
$$R(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

Then

$$\mathcal{O} = \{y \mid (y - c_{\mathcal{O}})^T P_{\mathcal{O}} (y - c_{\mathcal{O}}) \leq 1\}$$

and e.g. $\mathcal{R}(z) = \{x \mid A_{\mathcal{R}(z)} x \leq b_{\mathcal{R}(z)}\}$

Representation for ellipsoidal obstacles



Possible distance function:

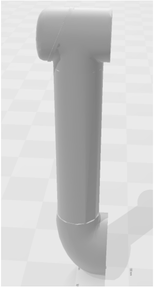
$$\Delta^2(\mathcal{R}(z), \mathcal{O}) = \min_{x,y} \quad \|x - y\|_2^2$$

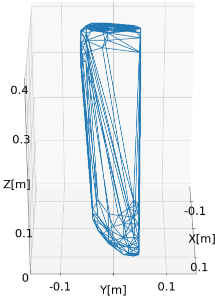
$$\text{s.t.} \quad A_{\mathcal{R}(z)} x \leq b_{\mathcal{R}(z)},$$

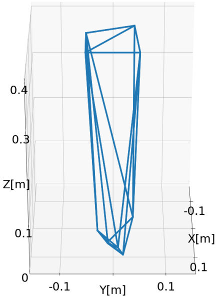
$$(y - c_{\mathcal{O}})^T P_{\mathcal{O}} (y - c_{\mathcal{O}}) \leq 1.$$

SIEMENS

Representing manipulator components as convex sets







Urdf files come with .stl meshes.

By taking the convex hull of the .stl mesh we get a convex set.

Randomly sampling subsets of vertices and taking the best volume match.

SIEMENS

IPOPT warm start behaviour (V-Pol)