TECHNION
Israel Institute of Technology

Time Optimal Robot Navigation with Navigation Functions

Leeor A. Ravina / GSC'25

Supervisor: Elon D. Rimon

Faculty of Mechanical Engineering, Technion

Navigation with Artificial Potential Fields

Path Planning

- Find a path between the start and target.
- Path must be safe – avoid obstacle collision.

Control

- Tracking the planned path.
- Control inputs must comply with the system constraints.

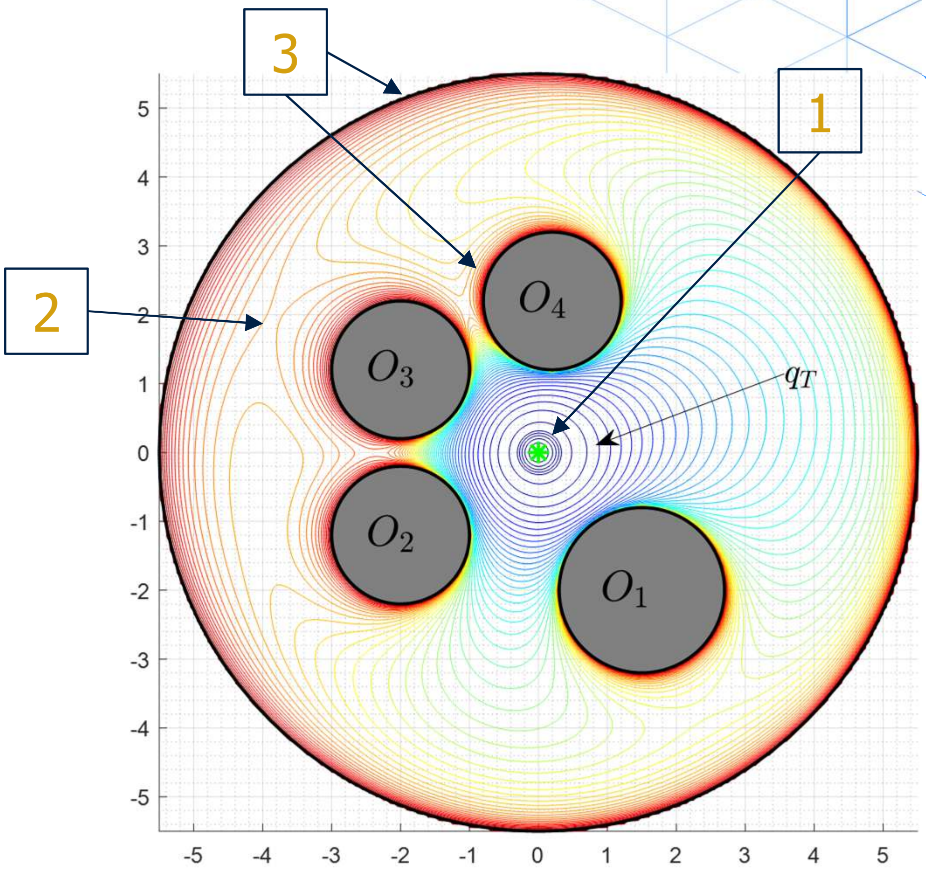
Artificial Potential Fields

- Encompasses complete information about the obstacle-free space.
- Used to form a feedback controller.

- Robot navigation using artificial potential fields unifies the path planning problem with the feedback controller design.

Navigation Functions

- Let q_T be a target point in the obstacle free space $\mathcal{W}$. A $\mathcal{C}^2$-smooth function$\varphi : \mathcal{W} \to [0,1]$ *navigation function* if
  - φ polar with unique minimum at target
  - All other critical points of φ are non-degenerate saddle points
  - φ admissible with uniform maximal value on boundary of $\mathcal{W}$



TECHNION
Israel Institute of Technology

Navigation Function Construction

- Using *tuning parameter* k the navigation function is

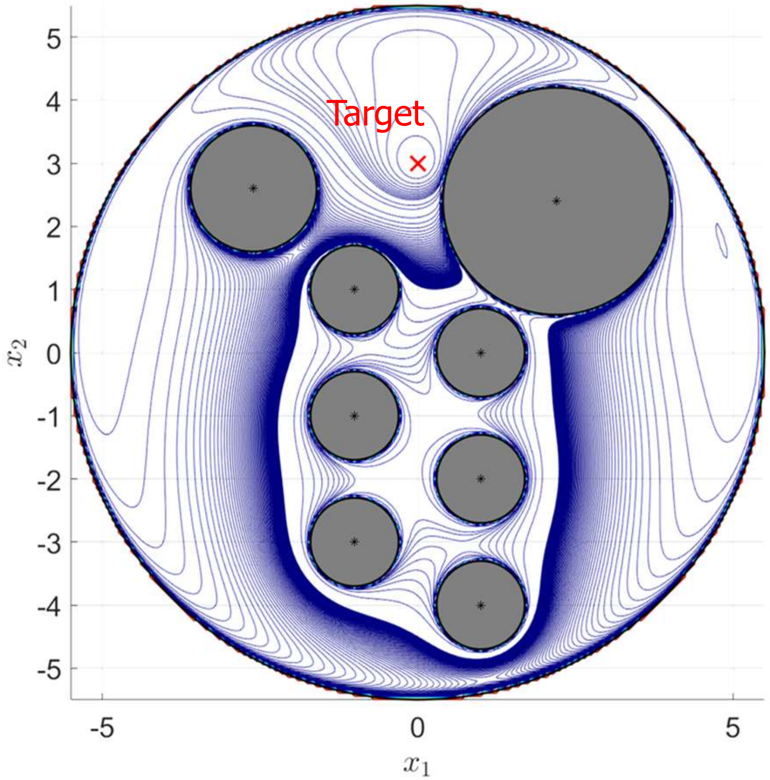
$$\varphi(q) = \frac{\gamma(q)}{\left(\gamma^{k}(q) + \beta(q)\right)^{\frac{1}{k}}}$$

Distance to Target

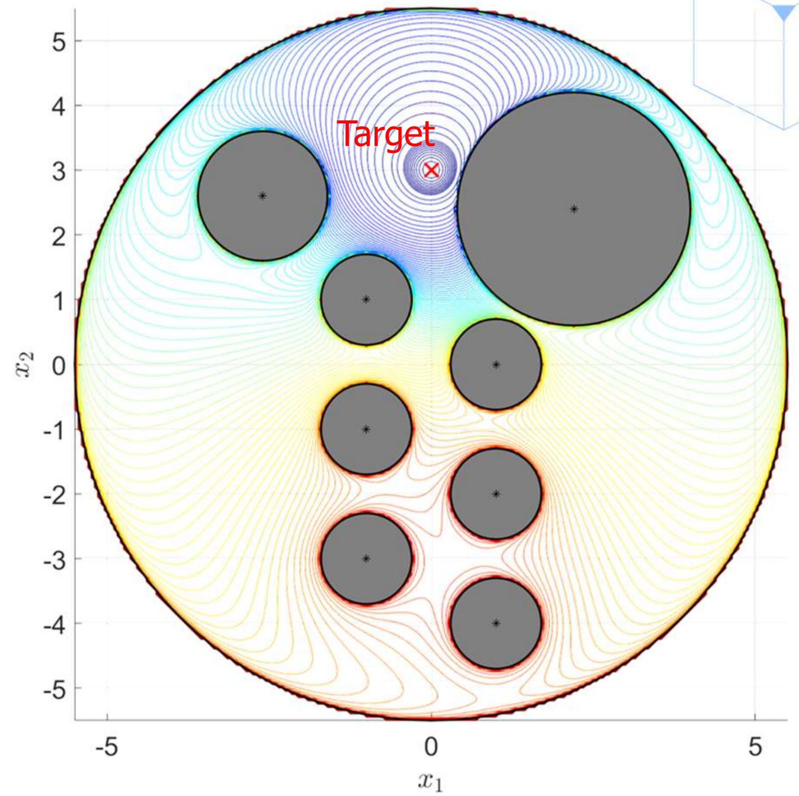
Obstacle Product
$\beta(q) = \prod \beta_i(q)$

Tuned Navigation Functions

$k = 2$



$k = 8$



TECHNION
Israel Institute of Technology

Navigation Function Classical Control Law

- Classical control law : $u(t) = -\nabla\varphi(x) - C\dot{x} \qquad C \succ 0$
- Asymptotic convergence to target using total mechanical energy as a Lyapunov function $V(x, \dot{x})$:

$$V(x, \dot{x}) = \varphi(x) + \frac{1}{2} m \left|\left|\dot{x}\right|\right|^2$$

$$\frac{\partial V}{\partial t} = \nabla\varphi \cdot \dot{x} + m \cdot (\dot{x} \cdot \ddot{x}) = -\dot{x}^T C \dot{x} \leq 0$$

- LaSalle's invariance principle: robot trajectories converge to target from almost all initial points in $\mathcal{W}$ (zero speed at target)

Bounded Control Input Problem

- Ensure **bounded control inputs** using *robot navigation functions*
- Preserve the navigation function **collision safety** and **arrival to target**
- Problem Statement :

> Construct feedback control law that guides robot system $m\ddot{x}(t) = u(t)$ to the target while avoiding collision with known obstacles under control input bound $||u|| \leq F_{max}$.

TECHNION
Israel Institute of Technology

Bounded Control Law

- Bounded control navigation function law

$$u(t) = -h \cdot \nabla\varphi - F_T \cdot \left(\widehat{\nabla\varphi} \cdot \hat{v}\right)\widehat{\nabla\varphi} - F_N \cdot \left(\widehat{\nabla\varphi}^{\perp} \cdot \hat{v}\right)\widehat{\nabla\varphi}^{\perp}$$

- Based on a conservative rectangular bound $(2F_T)^2 + F_N \leq F_{max}$.
- $\hat{v}$ denotes robot velocity direction - $\hat{v} = \frac{\dot{x}}{||\dot{x}||}$
- $h > 0$ is a scaling factor – used to enforce control input bound.
- Using the modified Lyapunov function

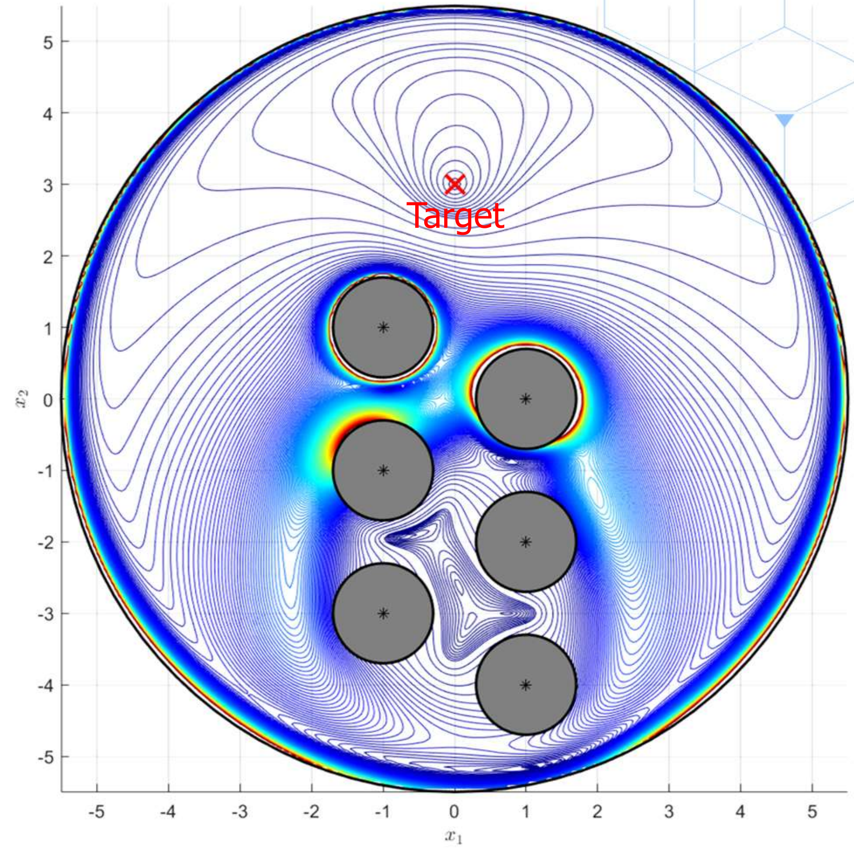
$$V(x, \dot{x}) = h \cdot \varphi(x) + \frac{1}{2} m \left|\left|\dot{x}\right|\right|^2$$

Gradient Scaling Caveat

- Gradient scaling by h requires <u>global information</u> about the environment:

$$h = \frac{F_T}{\max_{x \in \mathcal{W}} ||\nabla \varphi||}$$

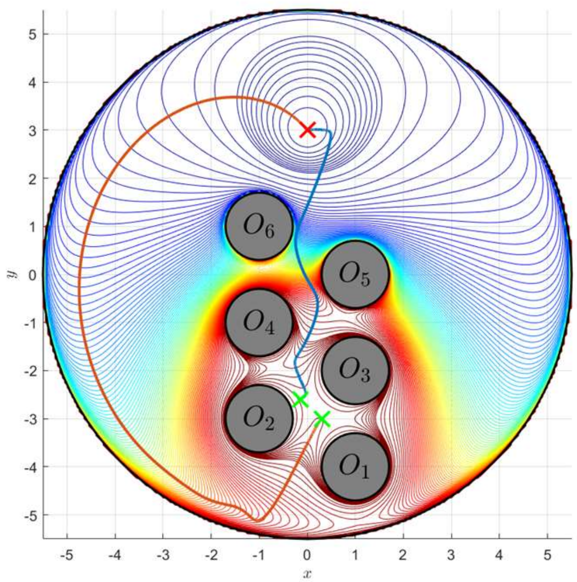
- Computation of h currently done numerically over the environment
- Adaptive on-line scaling law?



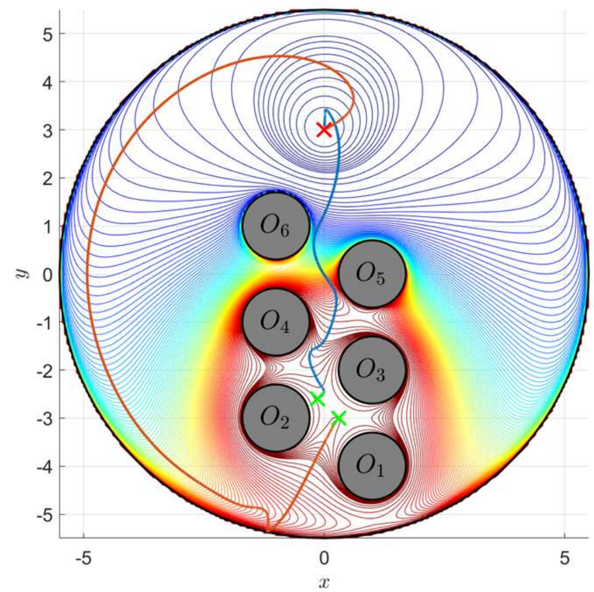
Gradient Magnitude $||\nabla \varphi||$

TECHNION
Israel Institute of Technology

Simulation Results – Classical Law Vs Bounded Law



Classical law trajectory



bounded control law trajectory

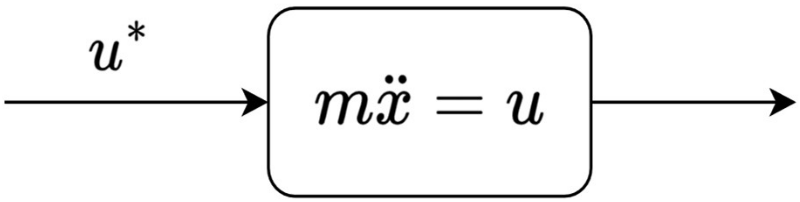
TECHNION
Israel Institute of Technology

Navigation Function Based Time Optimal Control

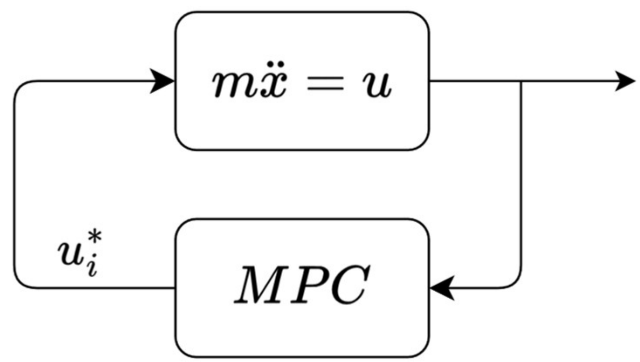
- Problem Statement :

> Construct feedback control law that guides robot system $m\ddot{x}(t) = u(t)$ to the target while avoiding collision with known obstacles under control input bound $||u|| \leq F_{max}$ in minimal navigation time t_f.

- Solution Approaches



Open-loop



Closed-loop

TECHNION
Israel Institute of Technology

Time Optimal Navigation – Open Loop

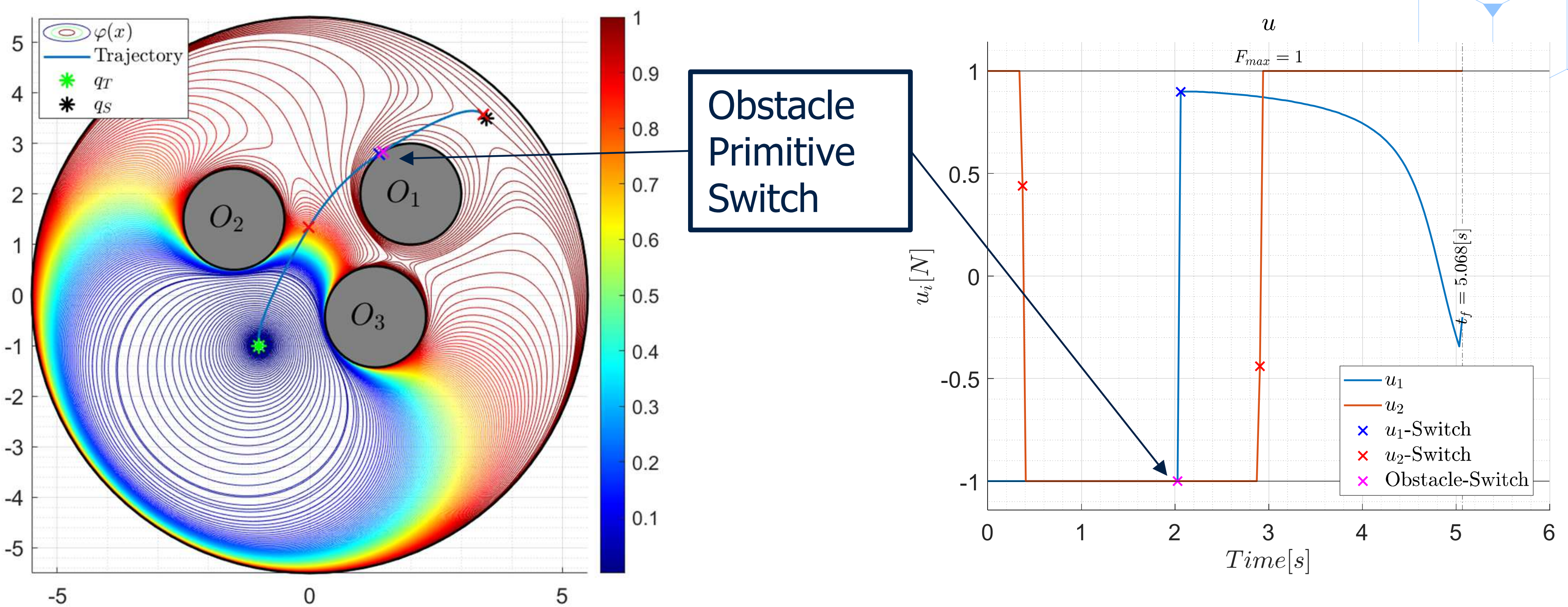
- Considers the optimal control problem

$$\begin{cases} \min_{x,y,u,t_f} t_f \ \ subjet\ to: \\ \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ \varphi(x,y) < 1 \\ |u_1| \le F_1, |u_2| \le F_2 \\ \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} = \begin{bmatrix} x_S \\ y_S \end{bmatrix} \ \ \begin{bmatrix} x(t_f) \\ y(t_f) \end{bmatrix} = \begin{bmatrix} x_S \\ y_S \end{bmatrix} \end{cases}$$

- The uniform maximal height of φ is used as a **single safety constraint.**

TECHNION
Israel Institute of Technology

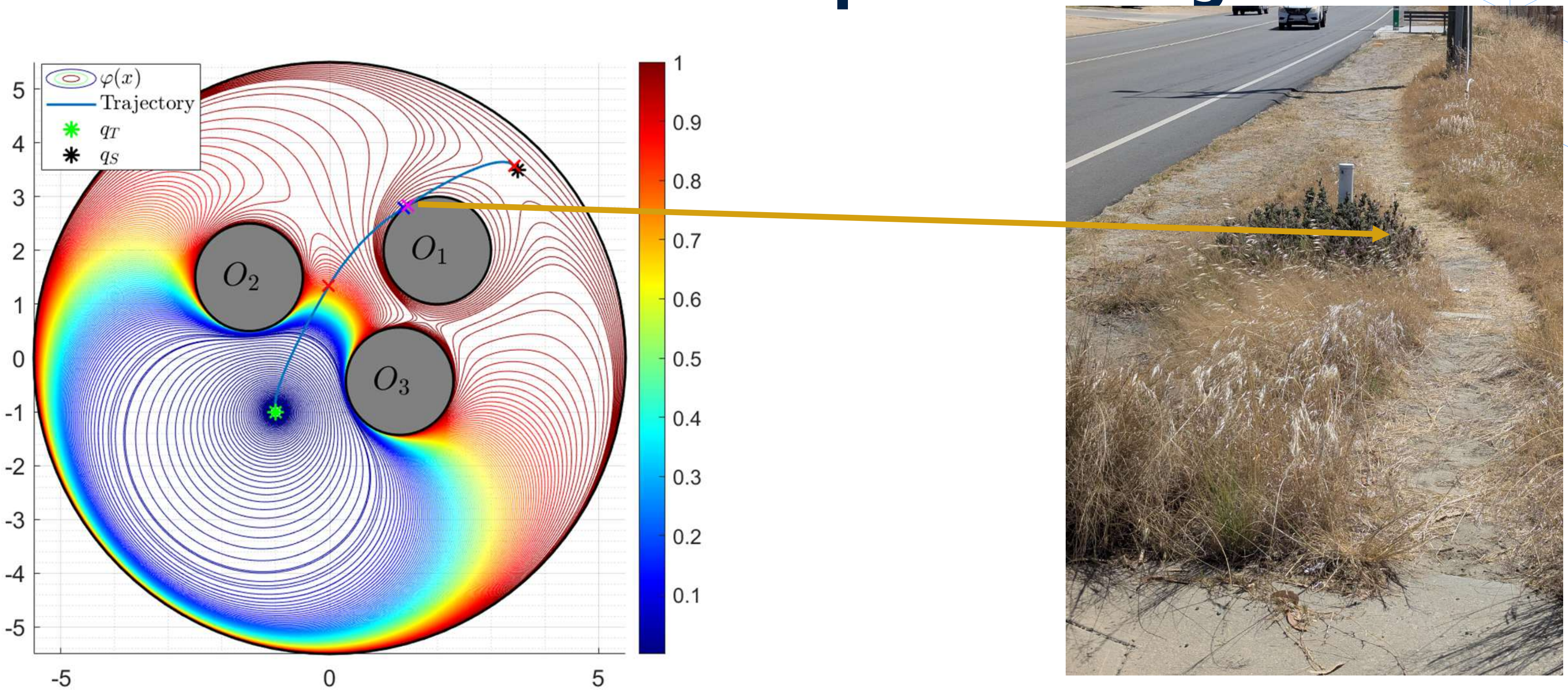
Simulation Results - Time Optimal Navigation



- Optimal control constructed by the primitives $\mathcal{B}^{+-}\mathcal{B}^{--}\mathcal{S}_2^-\mathcal{S}_2^+$.

TECHNION
Israel Institute of Technology

Simulation Results - Time Optimal Navigation



- Resulting optimal trajectory reminds of desire paths!

TECHNION
Israel Institute of Technology

Closed Loop Pseudo-Time Optimal Navigation

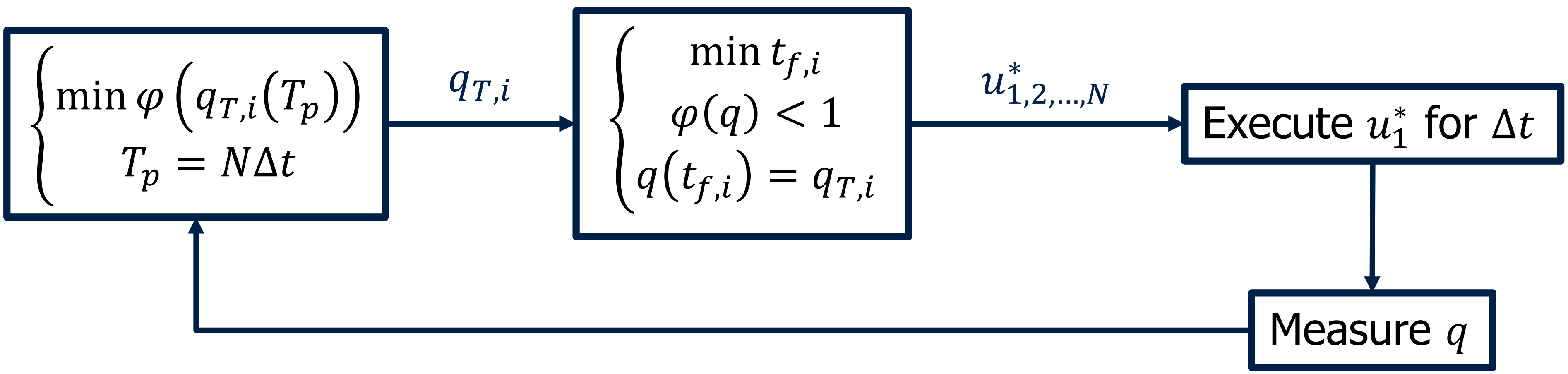
- Problem:
  - Minimum time solution requires global solution – invalid for real-time.
- Suggested Solution:
  - Choose virtual target that minimizes the navigation function value at the end of the prediction time T_p.

$$\begin{cases} \min\limits_{z_j,u_j} \varphi(x_N, y_N) \;\; subjet\ to: \\ z_{j+1} = (I + \Delta t A) z_j + \Delta t B u_j \\ \varphi\left(x_j, y_j\right) < 1 \\ |u_1| \le F_1, |u_2| \le F_2 \\ z_1 = z_{measured} \end{cases} \qquad \begin{gathered} 1 \le j \le N - 1 \\ T_p = N\Delta t \end{gathered}$$

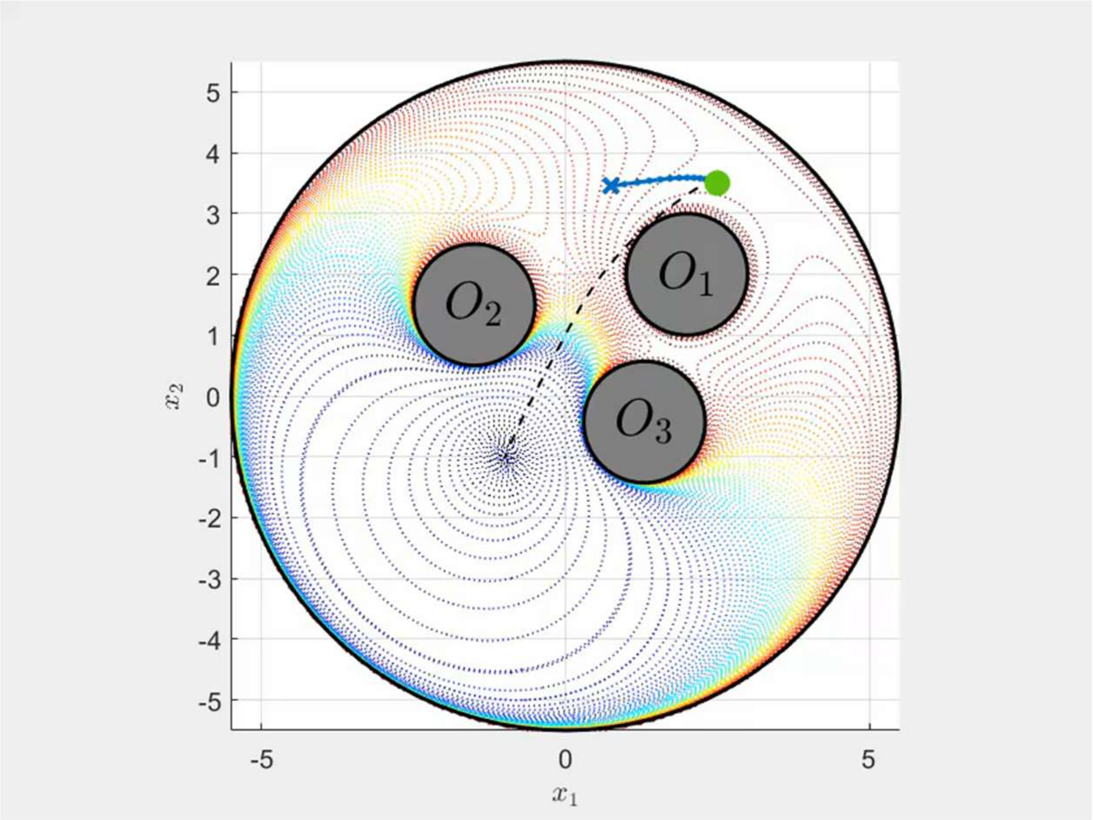
TECHNION
Israel Institute of Technology

Closed Loop Pseudo-Time Optimal Navigation

- Closed loop implementation of the solution of (1) using MPC.
- (1) requires global solution – an intermediate problem is solved.
- A virtual target $q_{T,i}$ is chosen to minimize $\varphi(q_{T,i})$.



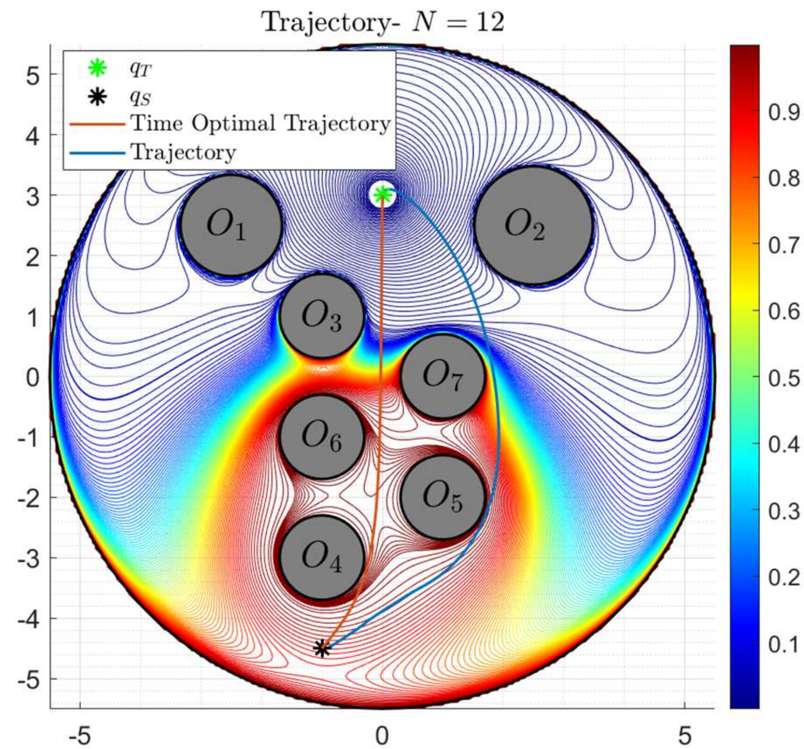
Closed Loop Pseudo-Time Optimal Navigation



TECHNION
Israel Institute of Technology

Simulation Results – MPC Navigation

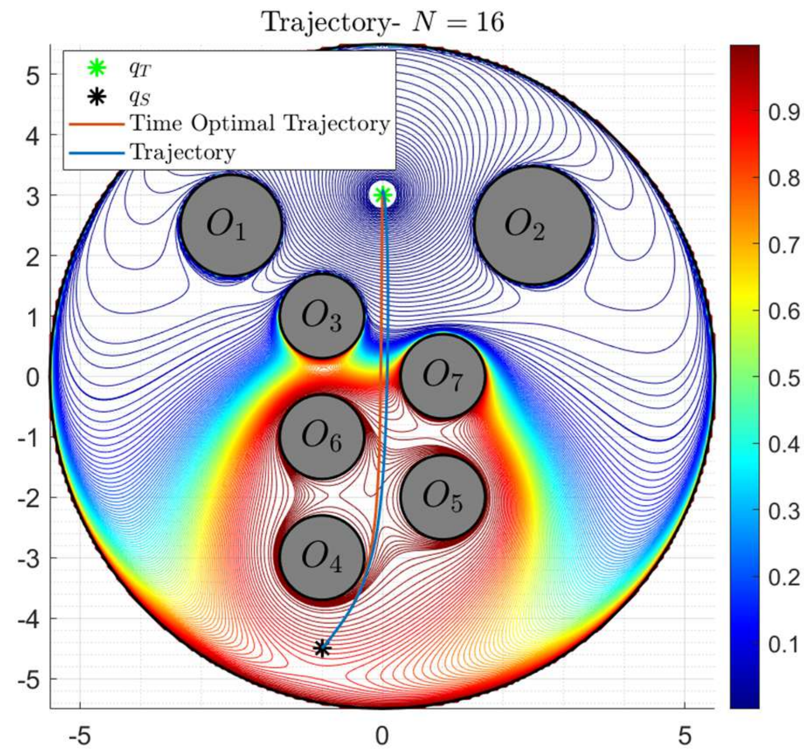
Prediction Steps N	Navigation Time t_f $[s]$	Solve time $\bar{t}_{solve}$ $[s]$
12	7.7	0.076
16	7.2	0.174
20	6.5	0.186
24	6.6	0.228
28	6.3	0.246
Time-optimal	5.4	-
Bounded	14.9	-



TECHNION
Israel Institute of Technology

Simulation Results – MPC Navigation

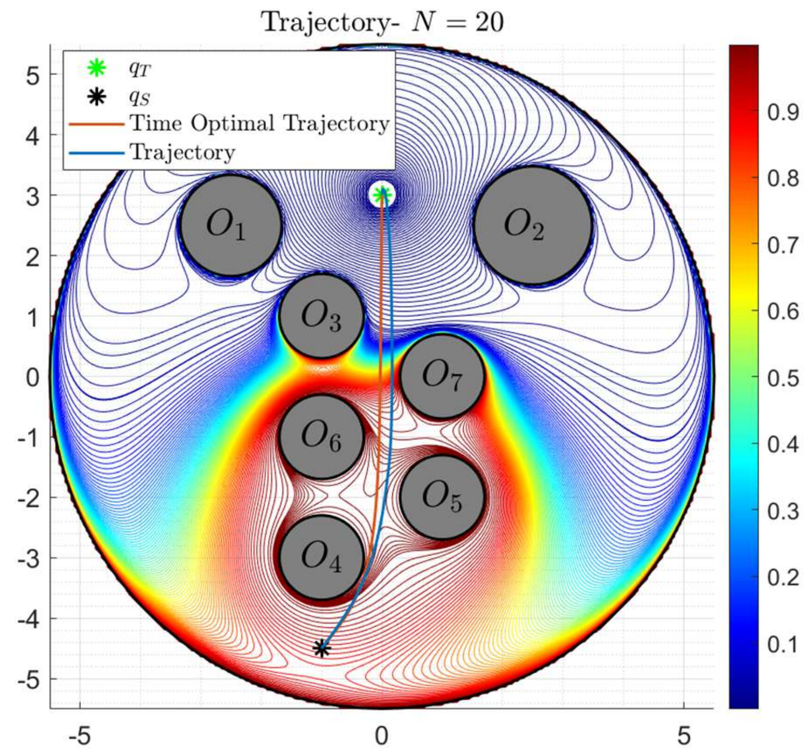
Prediction Steps N	Navigation Time t_f $[s]$	Solve time $\bar{t}_{solve}$ $[s]$
12	7.7	0.076
16	7.2	0.174
20	6.5	0.186
24	6.6	0.228
28	6.3	0.246
Time-optimal	5.4	-
Bounded	14.9	-



TECHNION
Israel Institute of Technology

Simulation Results – MPC Navigation

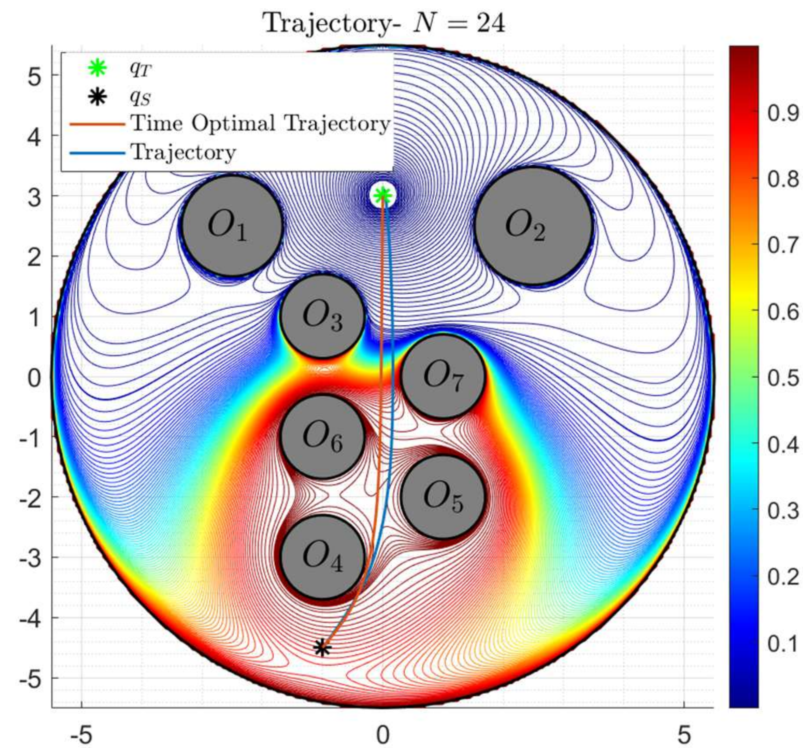
Prediction Steps N	Navigation Time t_f $[s]$	Solve time $\bar{t}_{solve}$ $[s]$
12	7.7	0.076
16	7.2	0.174
20	6.5	0.186
24	6.6	0.228
28	6.3	0.246
Time-optimal	5.4	-
Bounded	14.9	-



TECHNION
Israel Institute of Technology

Simulation Results – MPC Navigation

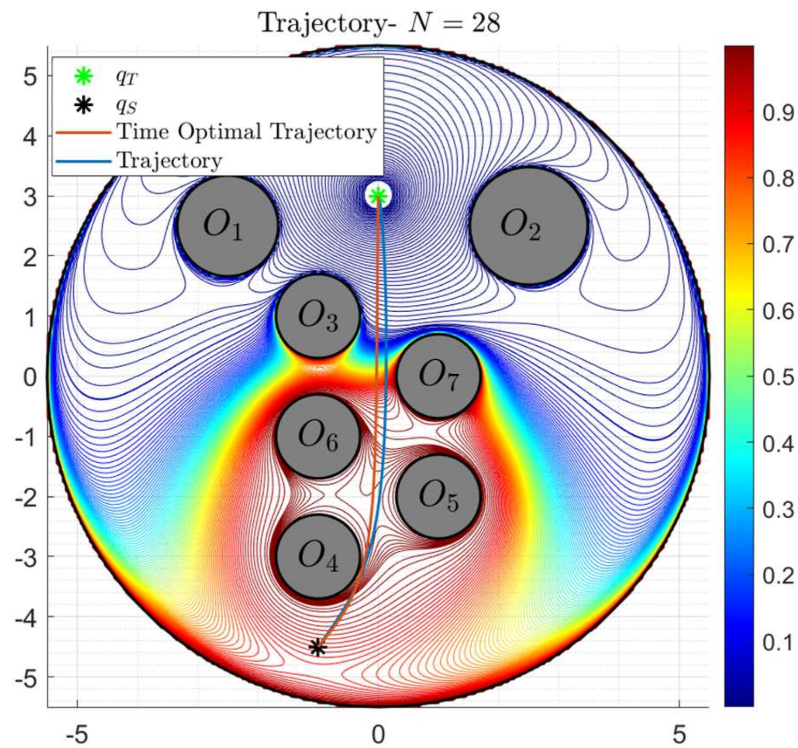
Prediction Steps N	Navigation Time t_f $[s]$	Solve time $\bar{t}_{solve}$ $[s]$
12	7.7	0.076
16	7.2	0.174
20	6.5	0.186
24	6.6	0.228
28	6.3	0.246
Time-optimal	5.4	-
Bounded	14.9	-



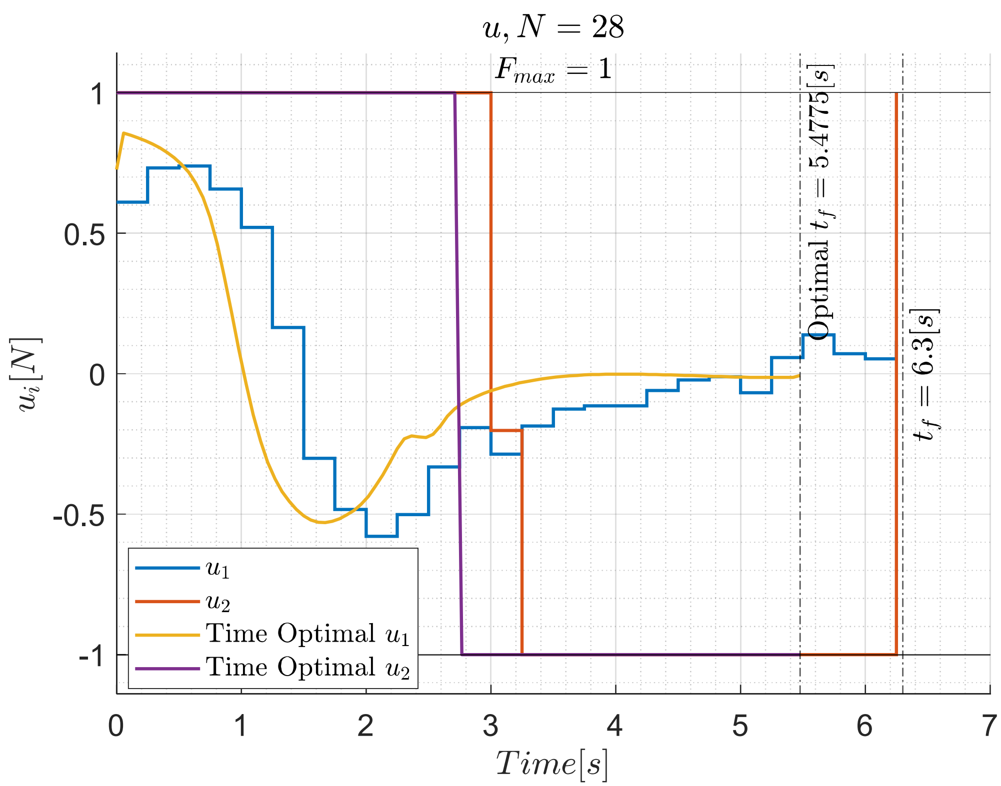
TECHNION
Israel Institute of Technology

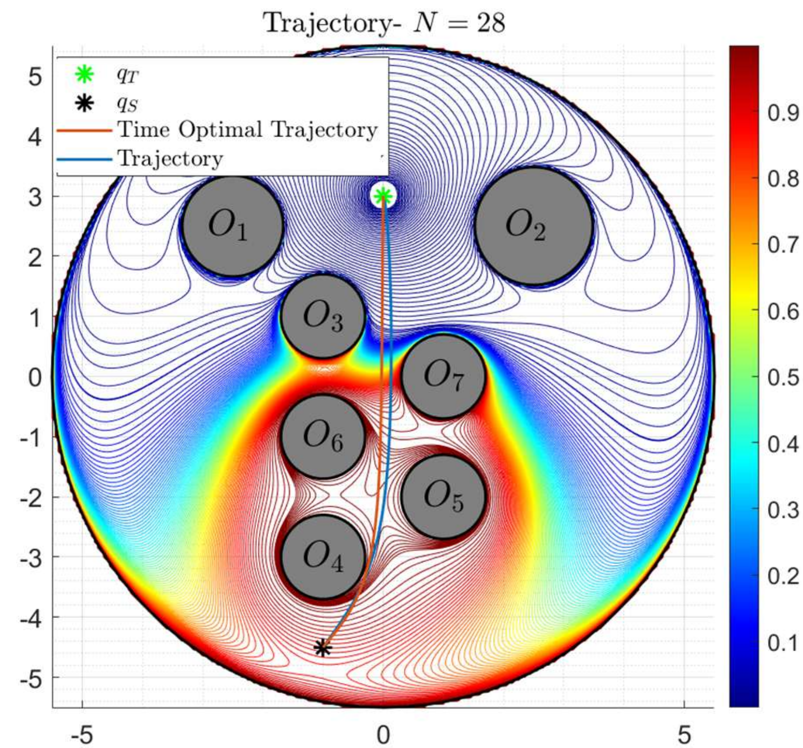
Simulation Results – MPC Navigation

Prediction Steps N	Navigation Time t_f $[s]$	Solve time $\bar{t}_{solve}$ $[s]$
12	7.7	0.076
16	7.2	0.174
20	6.5	0.186
24	6.6	0.228
28	6.3	0.246
Time-optimal	5.4	-
Bounded	14.9	-

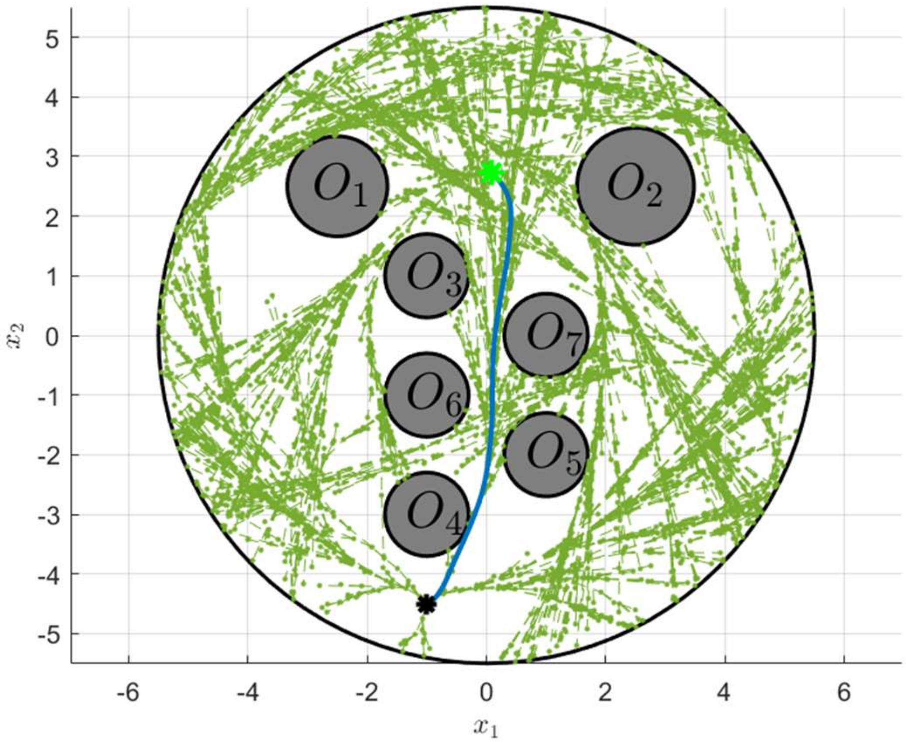


Simulation Results – MPC Navigation

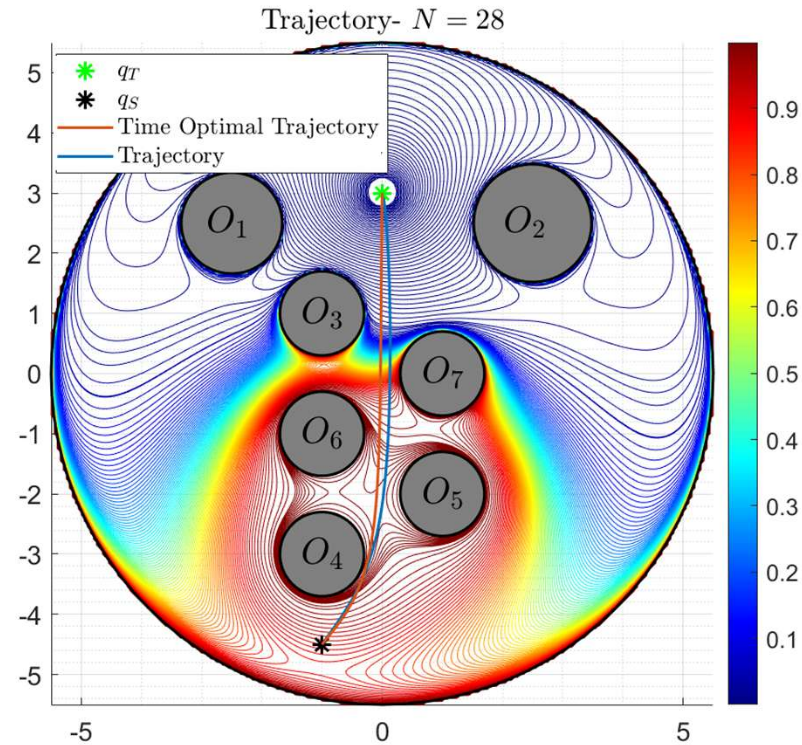




MPC Navigation Vs. Kinodynamic Planner



$t_f = 7.2[s]$[1]



$t_f = 6.3[s]$

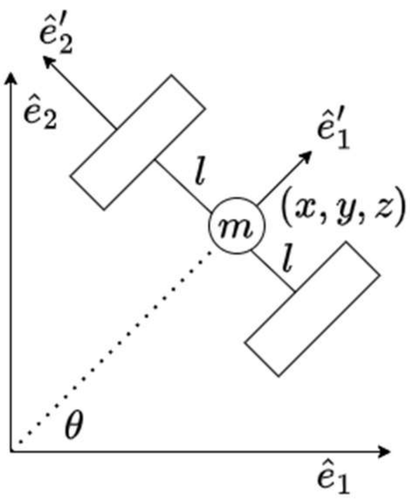
TECHNION
Israel Institute of Technology

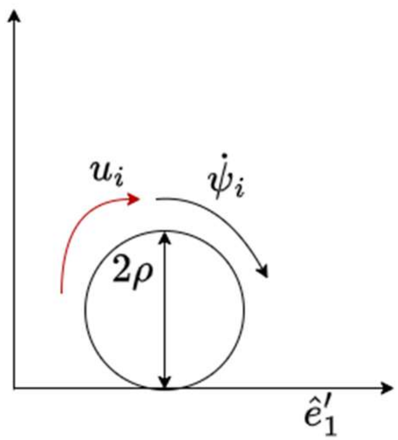
[1]SST* Planner from OMPL by Sucan et al. **ompl.kavrakilab.org**

Future Research Goals

- Integrate newer and faster solvers to increase prediction horizon.
- Replace double-integrator model with non-holonomic vehicle models.

$$\boxed{\begin{bmatrix}\ddot{x}\\ \ddot{y}\end{bmatrix} = \begin{bmatrix}u_1\\ u_2\end{bmatrix}}$$





Thank You!

Questions?