Real-Time Collision-Free Modification of Reference Trajectories for Vehicle Formations

Šimon Lehký, Tomáš Haniš, Kristian Hengster-Movrić

Fakulta elektrotechnická, ČVUT
Karlovo náměstí 13, Praha 2

MESAS 2026
October 21-23, 2026



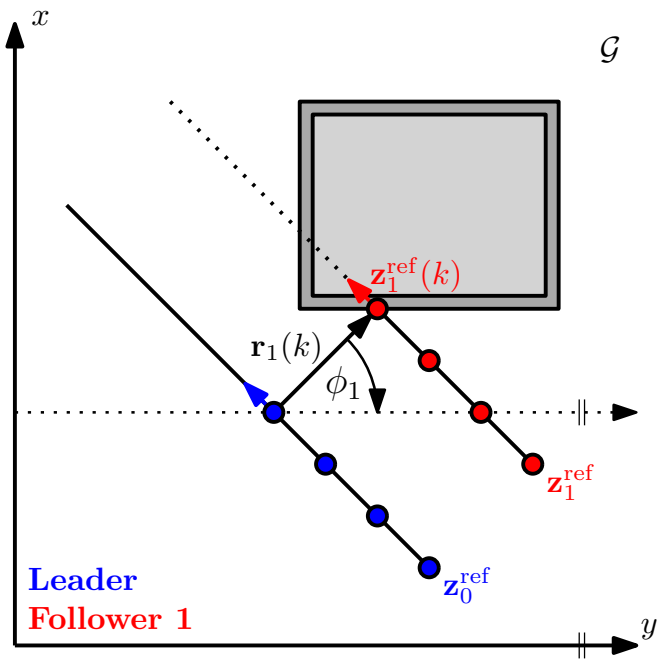


Motivation: loyal wingman formations

- The leader plans a **collision-free** trajectory $\boldsymbol{z}_0^{\mathrm{ref}}$
- Followers track geometric references

$$\boldsymbol{z}_i^{\mathrm{ref}}(k) = \boldsymbol{z}_0^{\mathrm{ref}}(k) + \boldsymbol{r}_i(k)$$

- These are **not guaranteed to be collision-free**
- Needed: a **local, real-time** correction on the follower's occupancy grid that
  - keeps the formation whenever possible
  - handles nonconvex obstacles
  - reports when no feasible reference exists



The nominal reference $\boldsymbol{z}_1^{\mathrm{ref}}(k)$ of the follower enters an obstacle.

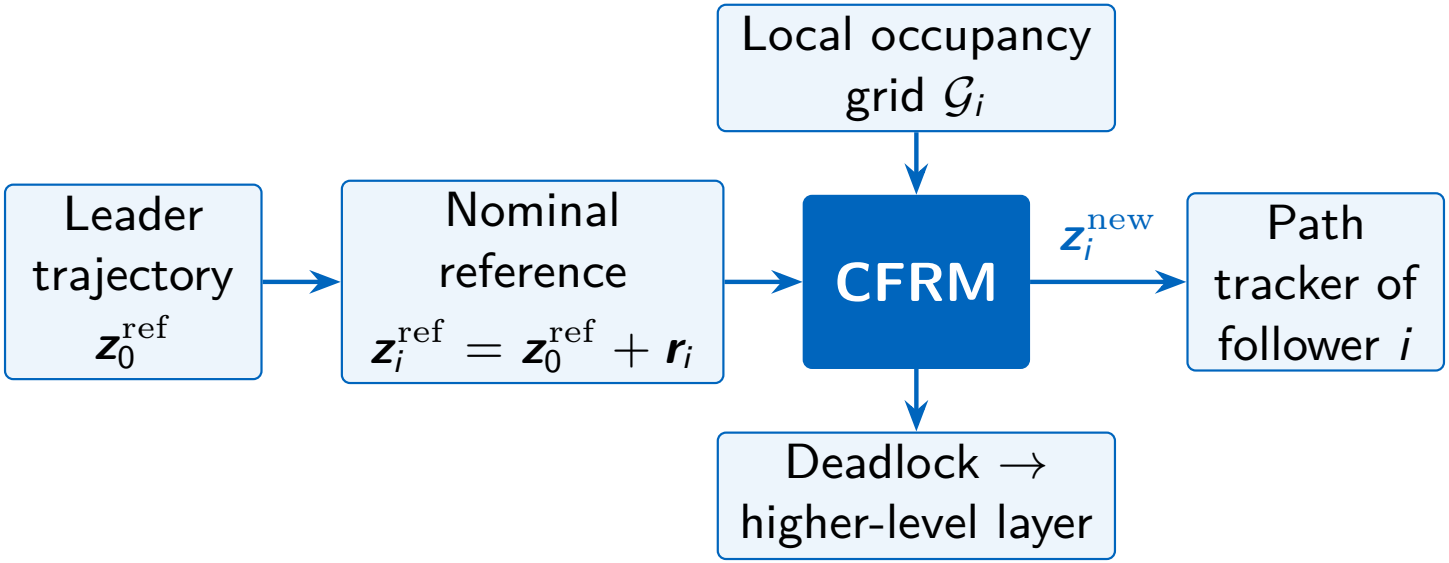
Existing approaches and the gap

Approach	**Limitation for the follower**
Joint replanning of the formation (sampling- and optimization-based)	computationally demanding, needs a global map
Local replanning of the follower (e.g., A*)	leaves the formation: when to deviate, how to rejoin?
Reactive methods (potential fields, dynamic window)	local minima near nonconvex obstacles
Safety filters, reference governors	need a dynamic model and an analytic safe set

Gap

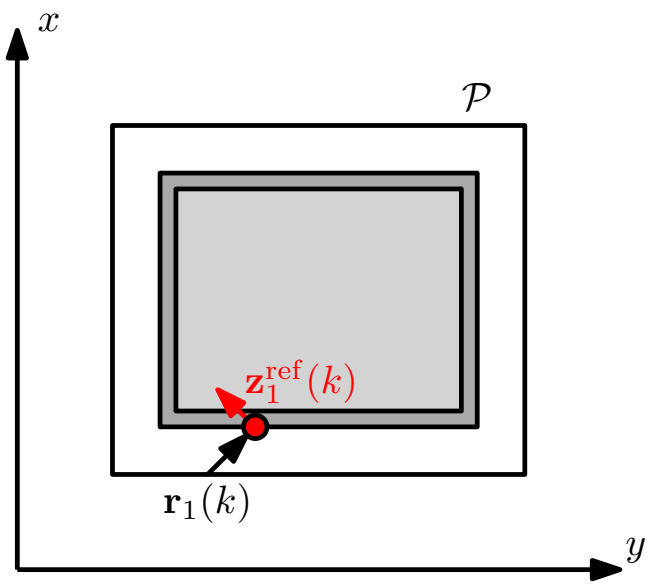
Correct the follower's **reference** directly on its **local occupancy grid**: collision-free, abreast of the leader whenever possible, at a small and bounded cost, and report when no such reference exists.

Collision-free reference modification (CFRM)

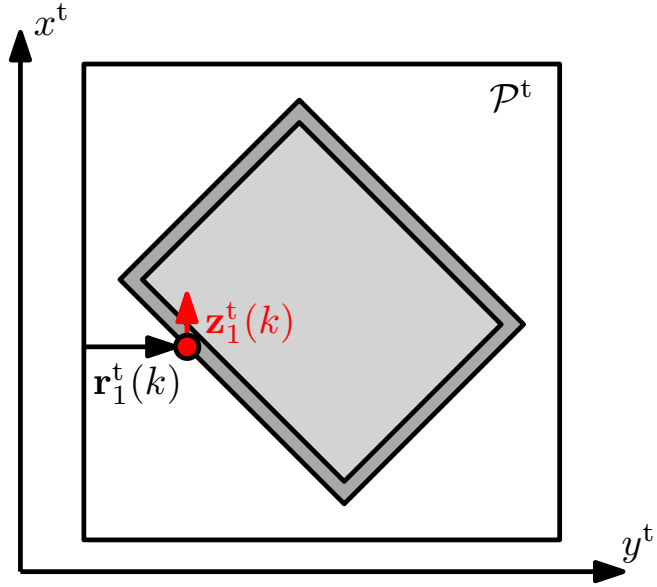


- **Lateral relocation** of a colliding reference to a nearby free cell: the follower stays **abreast of the leader**
- **One-dimensional search** after rotating the obstacle: no global map, no dynamic model, no gains to tune
- **Safe by construction:** collision-free references on a static grid; deadlocks are detected

Step 1: extract and rotate the obstacle



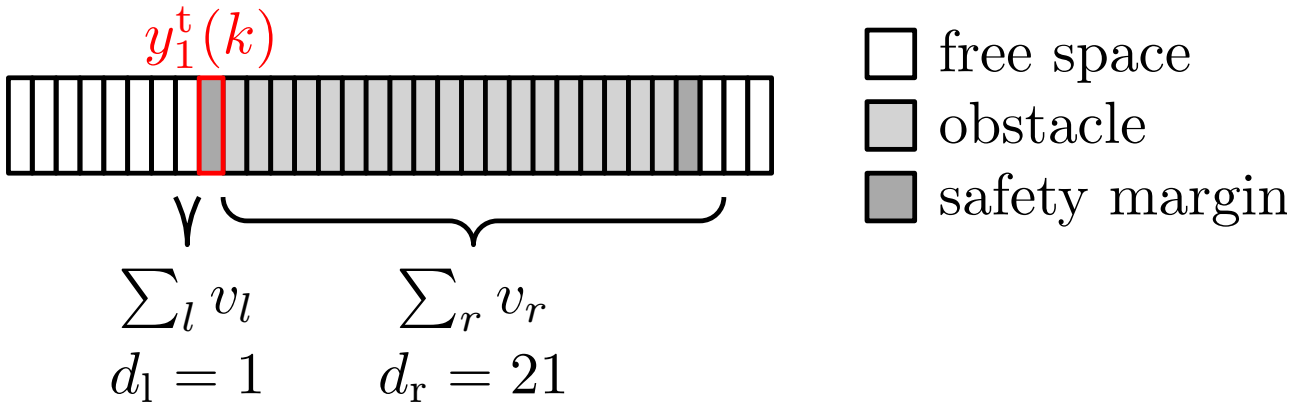
Subgrid $\mathcal{P}$ around the obstacle



Rotated subgrid $\mathcal{P}^{\mathrm{t}}$

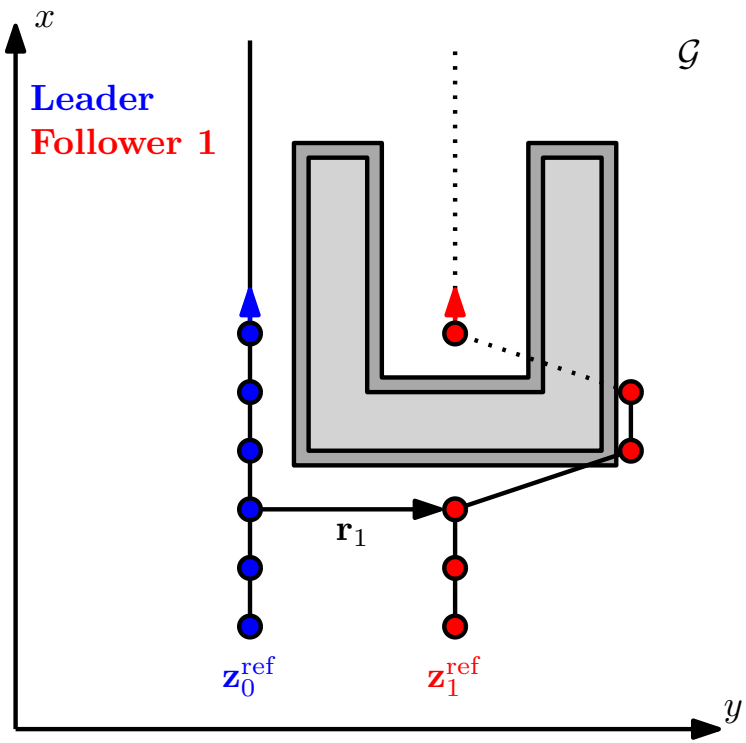
- **Subgrid $\mathcal{P}$:** the 4-connected obstacle component with the colliding cell, plus a padding
- **Rotation** by $\phi_1 = \mathrm{atan2}(r_{1,x}, r_{1,y})$: the offset $\boldsymbol{r}_1$ points along the rows
- Nearest-neighbour resampling; cells from outside $\mathcal{P}$ count as obstacles

Step 2: one-dimensional row search



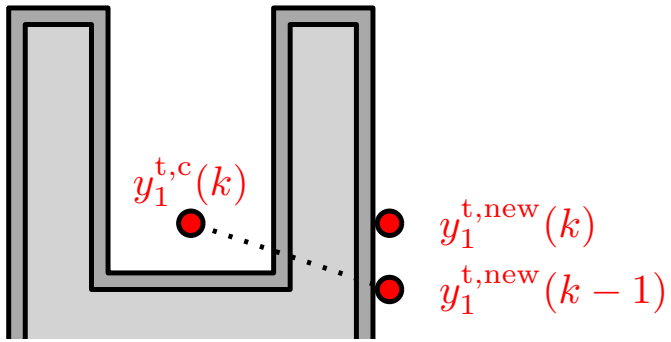
- Search the row through $y_1^{\mathrm{t}}(k)$ to the left (towards the leader) and to the right
- **Cost:** sum of the cell values up to the first **admissible** free cell (the next cell is free as well)
- **Lower cost wins** (left if equal): $y_1^{\mathrm{t,c}}(k) = y_1^{\mathrm{t}}(k) \pm d$, then transformed back by $\boldsymbol{R}_1^{-1}(\phi_1)$
- $\mathcal{P}$ and s are stored: the obstacle is passed on the same side

Nonconvex obstacles



The candidate is free, but the follower cannot reach it.

- **Problem:** the segment from the last reference to the candidate **crosses the obstacle**
- **Detection:** exact cell traversal of the segment $[\boldsymbol{z}_1^{\text{new}}(k-1), \boldsymbol{z}_1^{\text{c}}(k)]$
- **Resolution:** move along the row in direction s to the first free cell connected to the previous reference by free cells

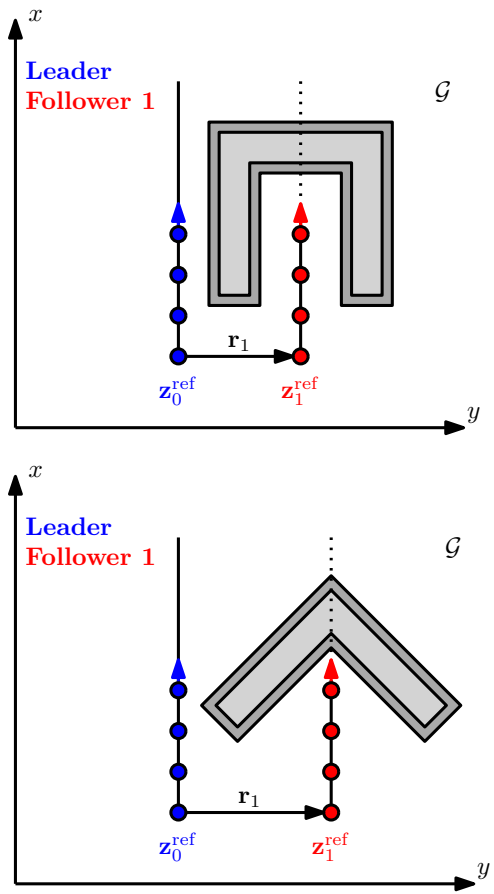


Safety guarantee and deadlocks

- Every new reference $\boldsymbol{z}$ is **validated**: it lies in a free cell, and the segment from $\boldsymbol{z}_1^{\text{new}}(k-1)$ touches no obstacle cell
- Otherwise, **lateral-first transition**:
  $\boldsymbol{z}_1^{\text{lat}}(k) = \boldsymbol{z}_1^{\text{new}}(k-1) + \hat{\boldsymbol{r}}_1 \hat{\boldsymbol{r}}_1^{\top} \left(\boldsymbol{z} - \boldsymbol{z}_1^{\text{new}}(k-1) \right)$
- If neither is valid, the reference is kept; **deadlock** after H retentions

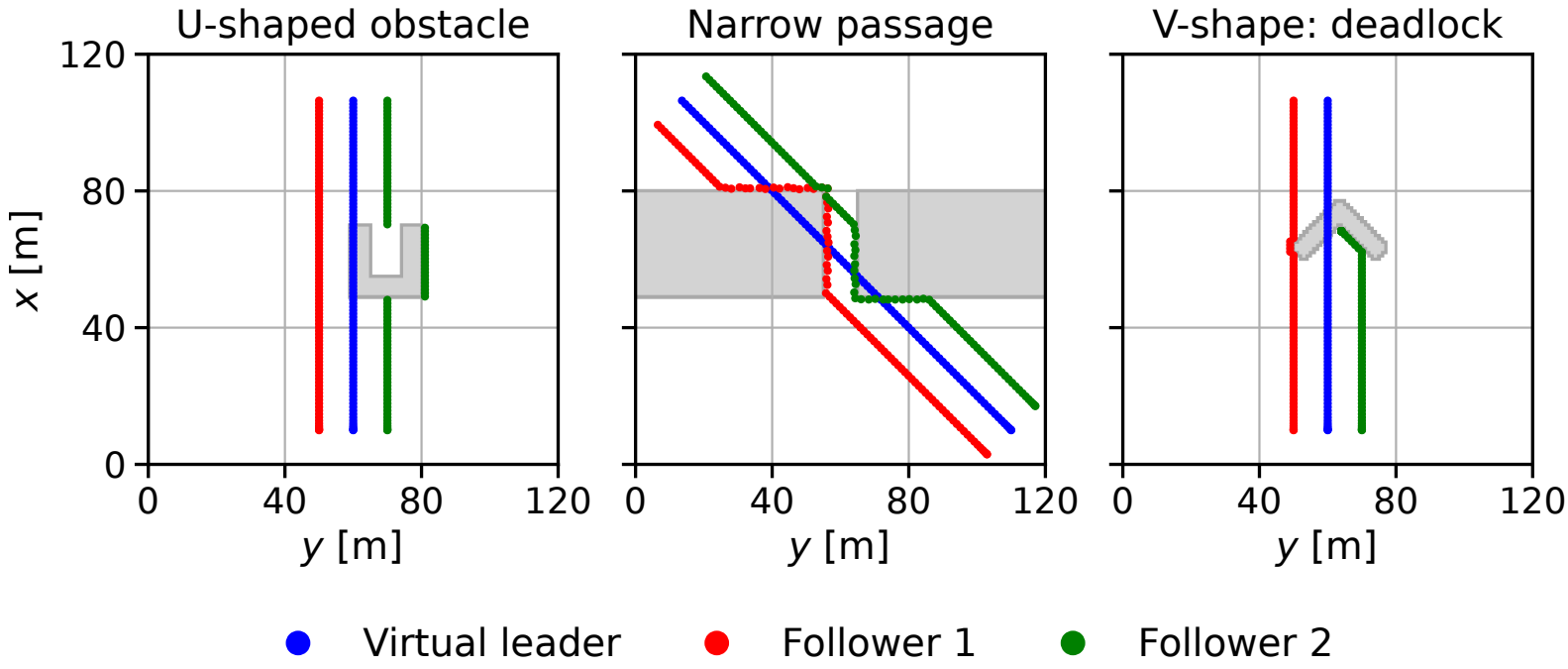
Guarantee on a static grid

Every reference lies in a free cell, and no segment between consecutive references touches an obstacle cell.



No feasible local reference: the deadlock is reported.

Demonstration in simulation



- Standalone C++ library; a virtual leader and two followers
- The references pass convex and nonconvex obstacles and narrow passages
- A deadlock in a concavity is detected and reported

Integration into the NATO AVT-408 autonomy stack

- CFRM runs as a **ROS 2 node** in the NATO AVT-408 autonomy stack
- Realistic simulation: the follower modifies its reference around obstacles while following the leader



Simulation environment (left) and RViz (right), by the NATO AVT-408 project team

Randomized benchmark

6000 scenarios

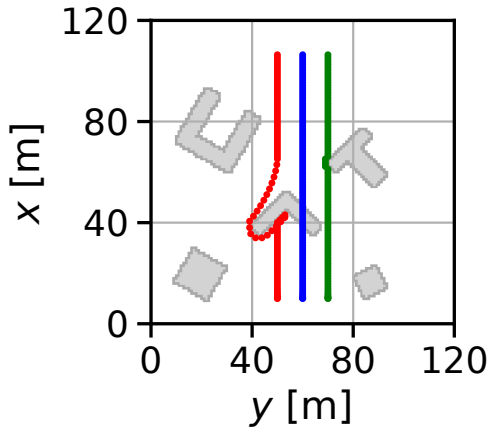
- Leader path of 195 m, random heading and curvature
- Two followers, offset $\pm d$, $d \sim \mathcal{U}(8, 20)$ m
- 1–6 obstacles: rectangles, parallelograms, convex polygons, L-, T-, U- and V-shapes
- Grid 1 m per cell, 0.5 m per update

9954 challenged follower runs

- **Success:** safe, ends at the formation position
- **Unsafe:** reference in, or segment through, an obstacle cell
- **Detected** or **undetected** stop

Compared methods

- **PROJ:** nearest free cell
- **APF:** artificial potential field
- **A*:** leaves the formation, replans locally
- **CFRM+R:** CFRM, A* after a deadlock



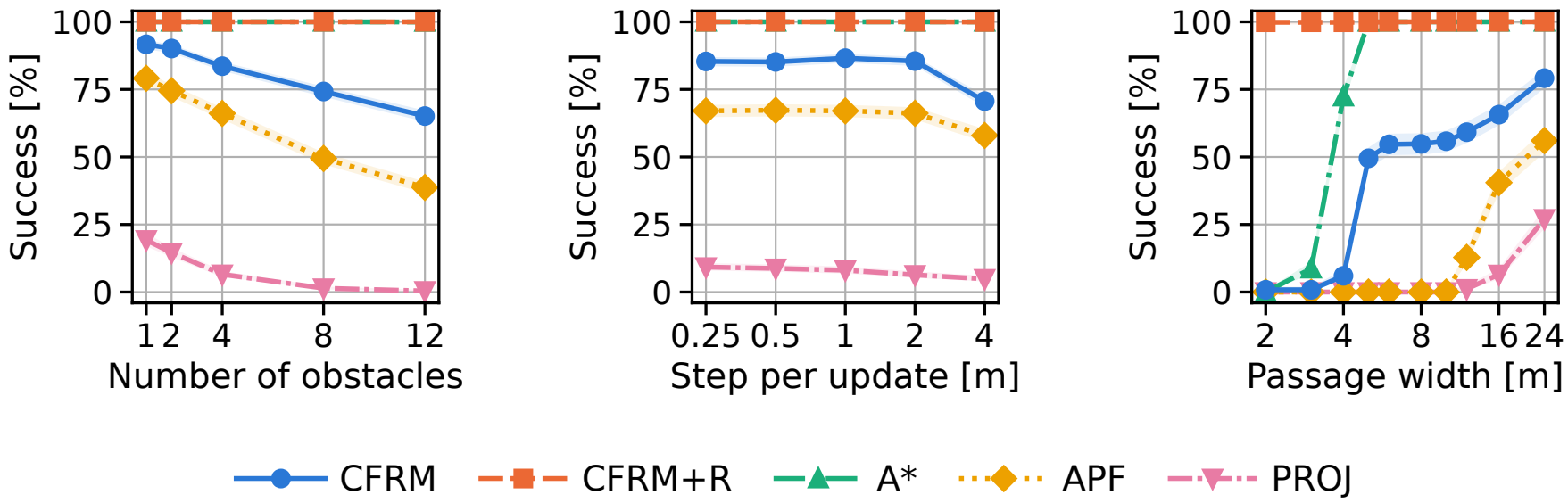
CFRM+R recovers from a deadlock.

Comparison with baseline methods

Method	Success [%]	Unsafe [%]	Detected stop [%]	Undetected stop [%]	Along-track dev. [m]
CFRM	**85.0**	**0.0**	15.0	**0.0**	**0.4**
CFRM+R	99.99	0.0	–	0.01	0.5
PROJ	8.6	91.4	–	0.0	1.9
APF	67.6	0.0	–	32.4	23.2
A*	100.0	0.0	–	0.0	5.9

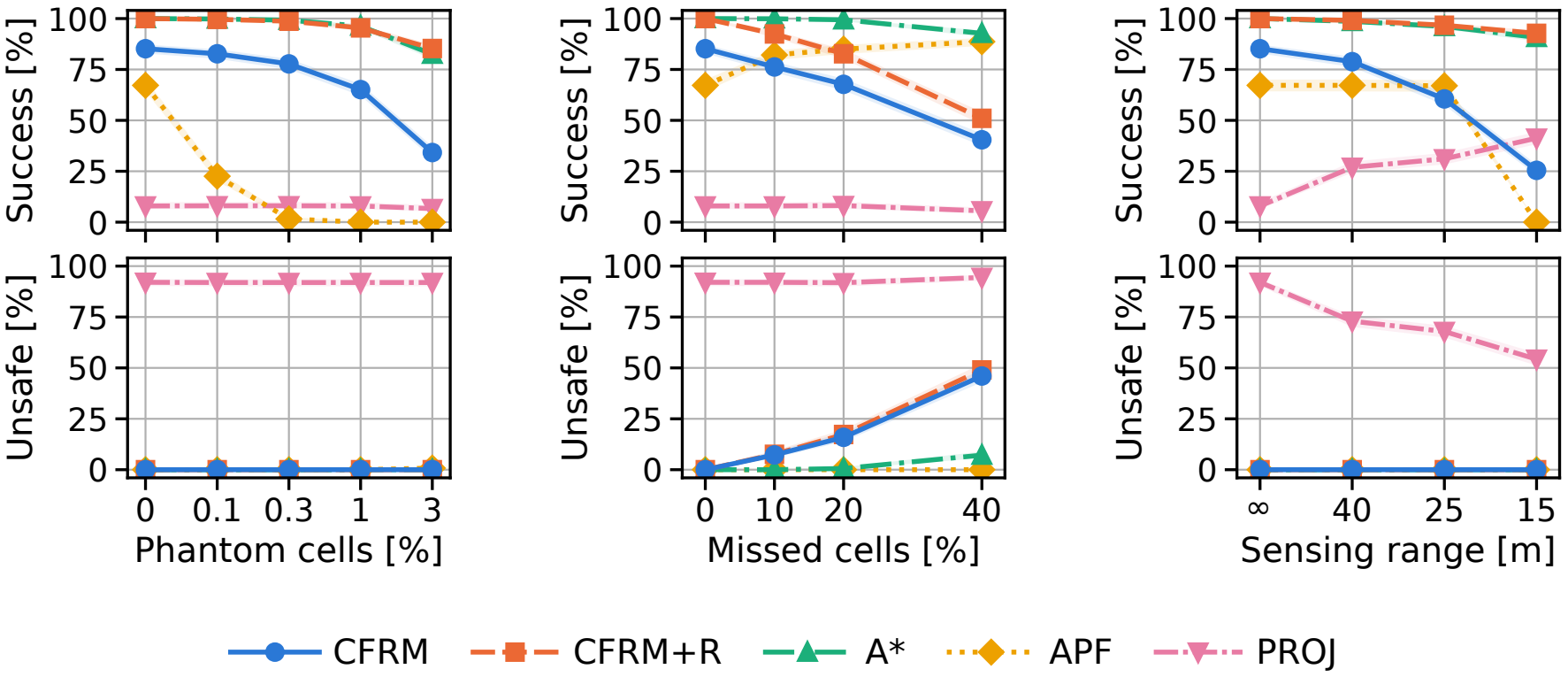
- **CFRM:** never unsafe, every failure is a detected deadlock, the follower stays abreast
- **APF:** stops without indication in a third of the runs (local minima)
- **A*:** always succeeds, but leaves the formation (5.9 vs. 0.4 m along the track)
- **PROJ:** jumps across obstacles
- Without the nonconvexity resolution, CFRM reaches only 70.2 %

Obstacle complexity and parameters



- One obstacle: **99.7–100 %** if convex, **81.8–89.3 %** if nonconvex (L, T, U, V)
- CFRM beats APF for every shape (by 8–17 points) and in all sweeps but one
- Resolution 0.25–1 m: little influence; step per update up to 2 m: no effect
- CFRM and CFRM+R: no unsafe reference in any of these experiments

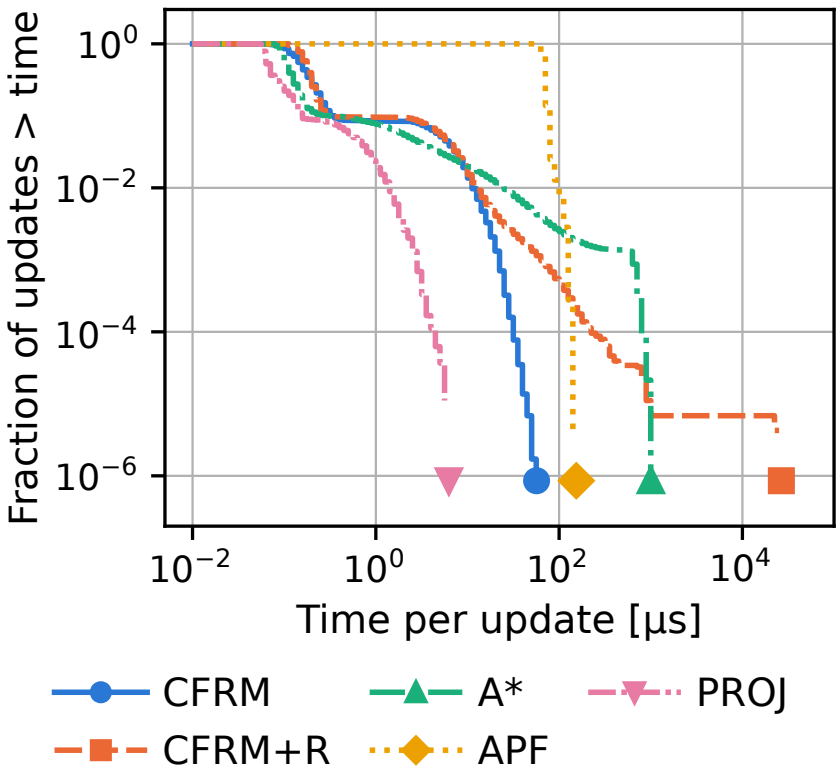
Imperfect occupancy grids



- **Phantom cells:** CFRM stays safe, but stops more often; APF fails already at 0.3 %
- **Missed cells:** critical, since CFRM trusts the grid
- **Limited range** (right): safe if unobserved cells count as occupied

⇒ **Filtered map:** suppress missed detections, unobserved = occupied, look far enough ahead

Computation time



Method	Mean [µs]	Max. [µs]
CFRM	**0.72**	**56.8**
CFRM+R	1.11	26 611
APF	66.22	154.0
A*	1.87	1003
PROJ	0.13	6.3

One core, Intel Xeon at 2.1 GHz

- Mean **90× faster** than APF; longest update 3× and 18× shorter than APF and A*
- Worst case 2.0 ms (0.25 m per cell, 80 m obstacles): 2 % of a 100 ms period

Conclusion

- **CFRM** corrects follower references in real time on the follower's own occupancy grid
- Collision-free references on a static grid; deadlocks are detected and reported
- Randomized evaluation with a correct grid:
  - never unsafe; 85.0 % success vs. 67.6 % for a potential field
  - follower stays abreast: 0.4 m vs. 5.9 m along the track for A*
  - 99.99 % success with a simple recovery layer
  - longest update 57 µs at 1 m per cell
- Integrated into the NATO AVT-408 autonomy stack

Outlook

Path tracker that respects the vehicle constraints and the grid (e.g., MPC), coordination between followers, filtered occupancy grids

Thank you for your attention.

Šimon Lehký
lehkysim@fel.cvut.cz
Faculty of Electrical Engineering, Czech Technical University in Prague