Distributed Motion Coordination Using Convex Feasible Set Based Model Predictive Control

Hongyu Zhou[1] and Changliu Liu[2]

1. Work done during Hongyu's internship at Intelligent Control Lab

2. Intelligent Control Lab, Robotics Institute, Carnegie Mellon University

Outline

- Motivation
- Contributions
- Introduction to Convex Feasible Set Algorithm
- Convex Feasible Set Based Distributed Model Predictive Control
- Conclusions and Future Work

Outline

- **Motivation**
- Contributions
- Introduction to Convex Feasible Set Algorithm
- Convex Feasible Set Based Distributed Model Predictive Control
- Conclusions and Future Work

Motivation

- Trajectory planning for connected autonomous vehicles remains challenging
- Optimization-based methods can generate smoother trajectories and take into account the interaction among vehicles, but suffer from high computational complexity and potential deadlocks
- Need to propose an efficient, safe, and coordinated multi-vehicle trajectory planning method

Outline

- Motivation
- **Contributions**
- Introduction to Convex Feasible Set Algorithm
- Convex Feasible Set Based Distributed Model Predictive Control
- Conclusions and Future Work

Contributions

- Extend convex feasible set (CFS) algorithm in a distributed fashion to solve multi-vehicle trajectory planning problem
- Propose a deadlock resolution by changing vehicle’s desire speed
- Simulate typical driving scenarios to validate our method

Outline

- Motivation
- Contributions
- **Introduction to Convex Feasible Set Algorithm**
- Convex Feasible Set Based Distributed Model Predictive Control
- Conclusions and Future Work

Introduction to Convex Feasible Set Algorithm

- An optimization algorithm for real time motion planning
- Handle motion planning problems with convex objective functions and non-convex inequality constraints
- Idea: obtain convex feasible sets within the non-convex inequality constraints
- Solve the convex subproblems iteratively until solutions converge

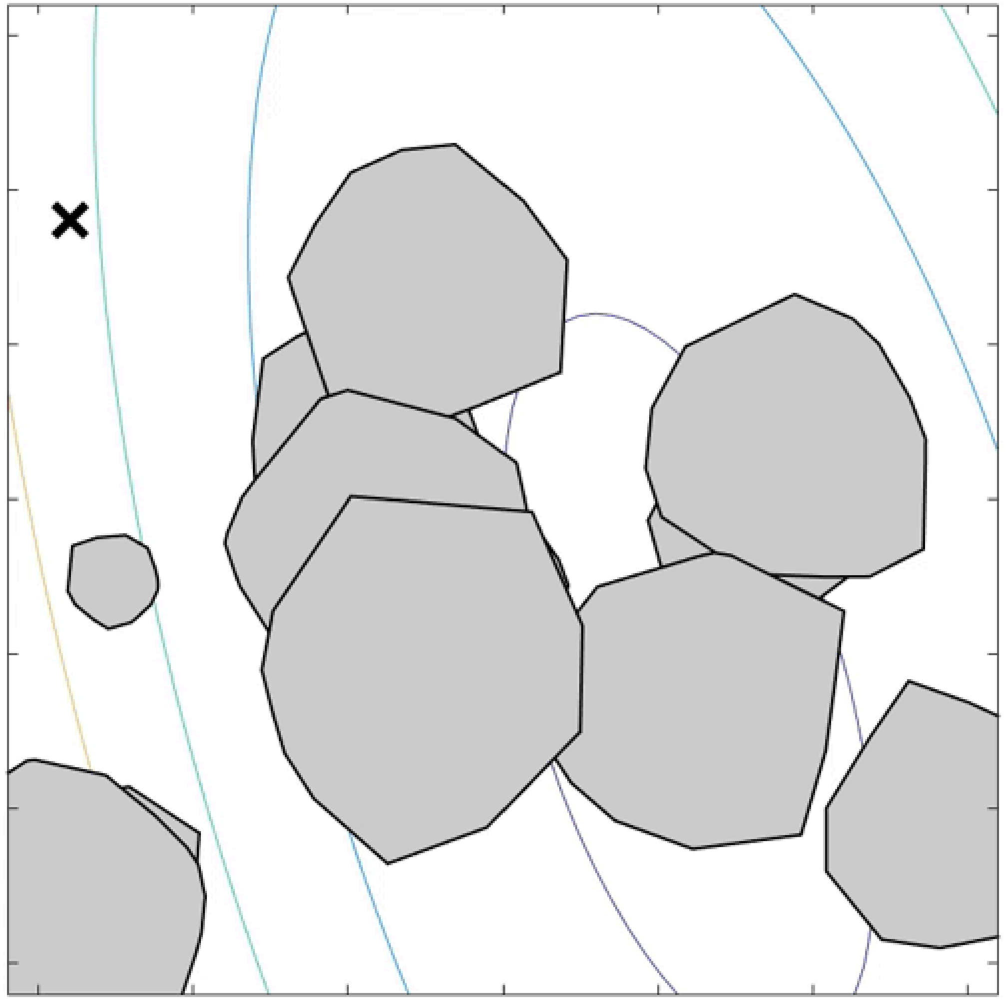
C. Liu, C. Lin, and M. Tomizuka, "The convex feasible set algorithm for real time optimization in motion planning", in *SIAM Journal on Control and Optimization*, vol. 56, no. 4, pp. 2712-2733, Jul. 2018

Introduction to Convex Feasible Set Algorithm

Pseudocode:

Algorithm 1: The Convex Feasible Set Algorithm

```
Initialize initial guess x^(0), k := 0;
while True do
    Find a convex feasible set F^(k) ⊂ Γ for x^(k) ;
    Solve the convex optimization problem for x^(k+1);
    if Terminal condition is satisfied then
        Break the while loop;
    end
    k := k + 1;
end
return x^(k+1);
```

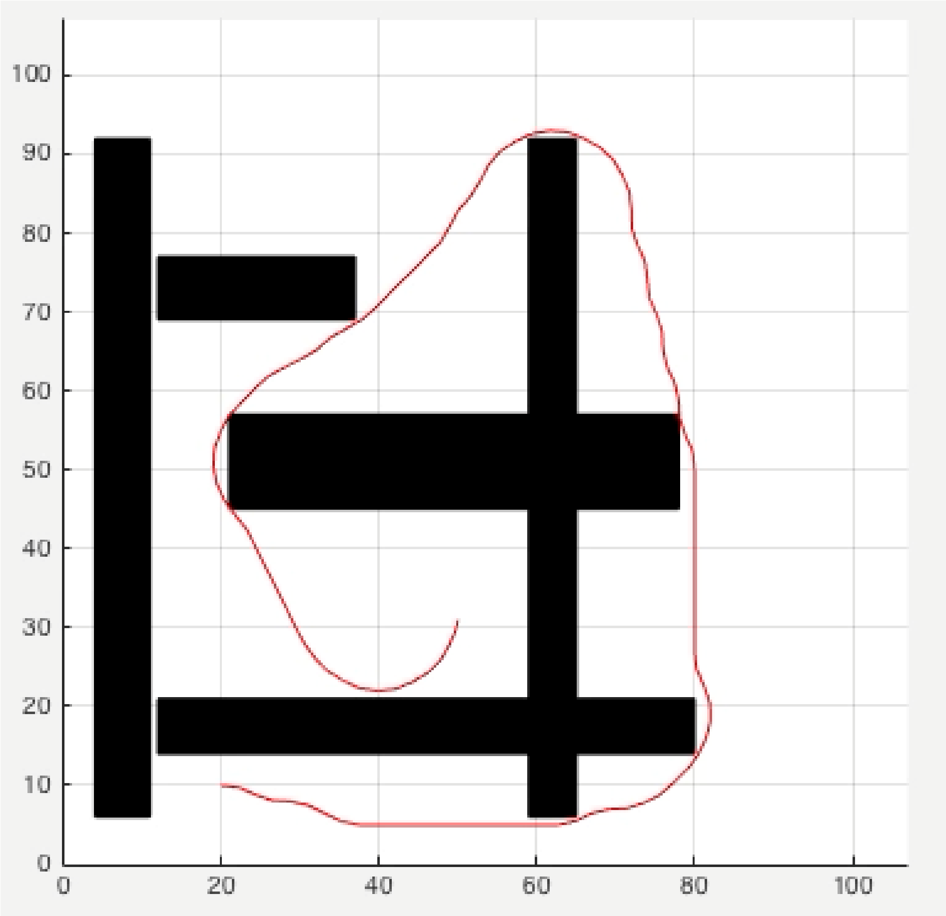


Courtesy of C. Liu

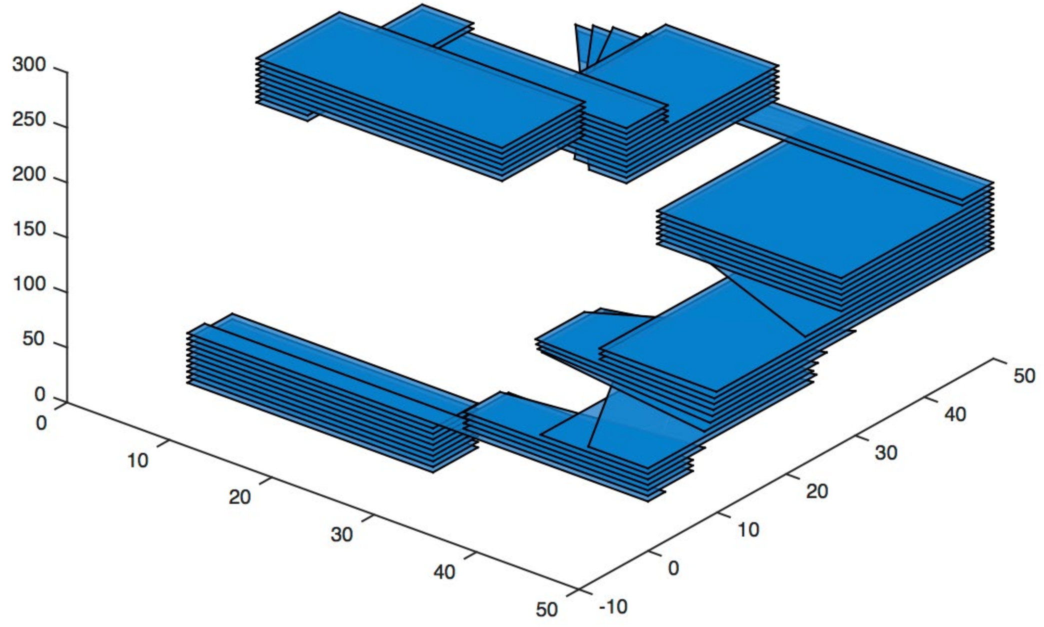
C. Liu, C. Lin, and M. Tomizuka, "The convex feasible set algorithm for real time optimization in motion planning", in *SIAM Journal on Control and Optimization*, vol. 56, no. 4, pp. 2712-2733, Jul. 2018

Introduction to Convex Feasible Set Algorithm

CFS for Efficient Long Term Planning



- Stack the set of safe control (half spaces) for all time steps
- Reduce the non-convex optimization problem to a convex problem



C. Liu, C. Lin, and M. Tomizuka, "The convex feasible set algorithm for real time optimization in motion planning", in *SIAM Journal on Control and Optimization*, vol. 56, no. 4, pp. 2712-2733, Jul. 2018

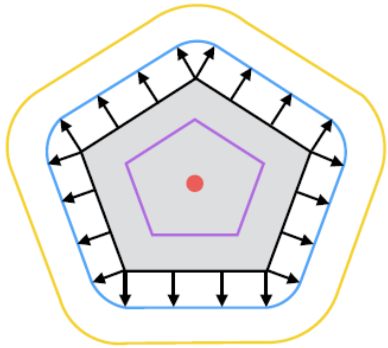
Outline

- Motivation
- Contributions
- Introduction to Convex Feasible Set Algorithm
- **Convex Feasible Set Based Distributed Model Predictive Control**
- Conclusions and Future Work

CFS-DMPC

Notation:

- i: the index of ego vehicle
- j: the index of surrounding vehicles
- H: the planning horizon
- $\mathrm{x}_i = [x_i^1; \dots; x_i^H]$: the trajectory of vehicle i with x_i^h as 2D position at time step h
- s_i: the slack variable
- $J_i(\mathrm{x}_i, s_i)$: the objective function for vehicle i
- $d(\cdot)$: the signed distance function, e.g., $d\left(x_i^h, x_j^h\right)$ is the distance between x_i^h and x_j^h at time step h
- O_j^h: boundary of vehicle j at time step h as an obstacle



C. Liu, C. Lin, and M. Tomizuka, "The convex feasible set algorithm for real time optimization in motion planning", in *SIAM Journal on Control and Optimization*, vol. 56, no. 4, pp. 2712-2733, Jul. 2018

CFS-DMPC

Cost function:

$J_i(\mathrm{x}_i, s_i)$

Total cost

- **Cost of trajectory deviation**: $J_i^o(\mathrm{x}_i) = \frac{1}{2} c_o (\mathrm{x}_i - \mathrm{x}_i^{ref})^{\mathrm{T}} (\mathrm{x}_i - \mathrm{x}_i^{ref})$
- **Cost of trajectory acceleration**: $J_i^a(\mathrm{x}_i) = \frac{1}{2} c_a (\mathbf{A}_i \mathrm{x}_i)^{\mathrm{T}} (\mathbf{A}_i \mathrm{x}_i)$
- **Cost of magnitude of slack variable**: $J_i^s(s_i) = c_s {s_i}^{\mathrm{T}} s_i$

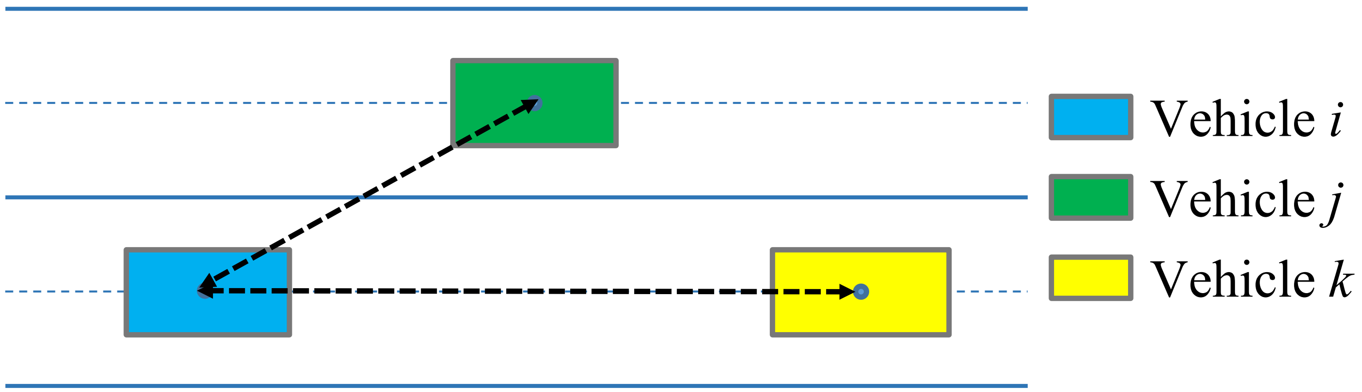
x_i^{ref} : the reference trajectory according to vehicle's desired speed

$\mathbf{A}_i$: linear operator that maps x_i to the accelerations along the trajectory

CFS-DMPC

Constraints:

- **Safety constraint**: force each vehicle pair (i, j) to maintain a safety distance at every time step



- **Initial position**: make the planned trajectories to start as close to the vehicle' current position as possible

CFS-DMPC

Safety constraint:

Assume vehicles are connected, i.e., are able to share low-bandwidth information, this is constant

$$\phi(x_i^h, \bar{x}_j^h) = d(x_i^h, \bar{x}_j^h) - d_{min} \geq 0$$

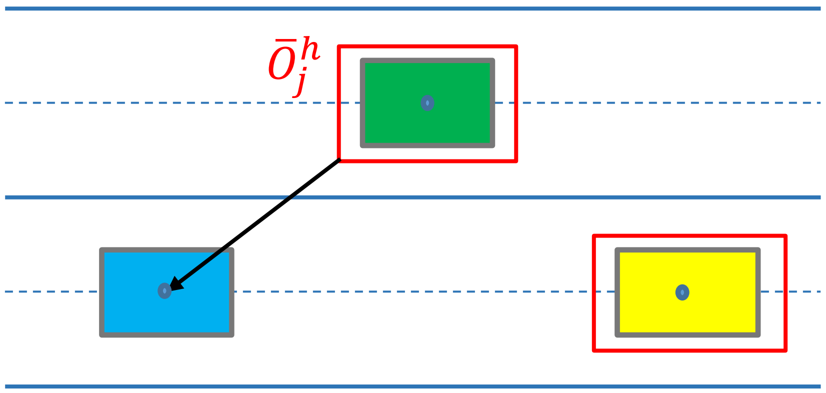
with d_{min} as the safety margin

Use CFS to convexify the safety constraint

$$\phi_{i,j}^{h,k} + \nabla\phi_{i,j}^{h,k} \cdot \left(x_i^h - x_i^{h(k)}\right) \geq 0$$

with $\phi_{i,j}^{h,k} := \phi\left(x_i^{h(k)}, \bar{O}_j^h\right) = d\left(x_i^{h(k)}, \bar{O}_j^h\right) - d_{min}$

for all surrounding vehicle j and time step h



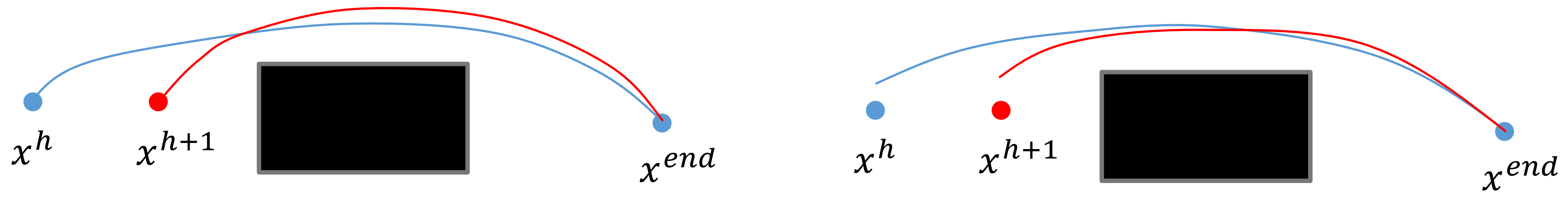
CFS-DMPC

Initial position:

$$x_i^1 = x_i^c + s_i$$

with x_i^c as the current position of vehicle i

- Adding slack variable can minimize the difference between the planned trajectories of adjacent time steps

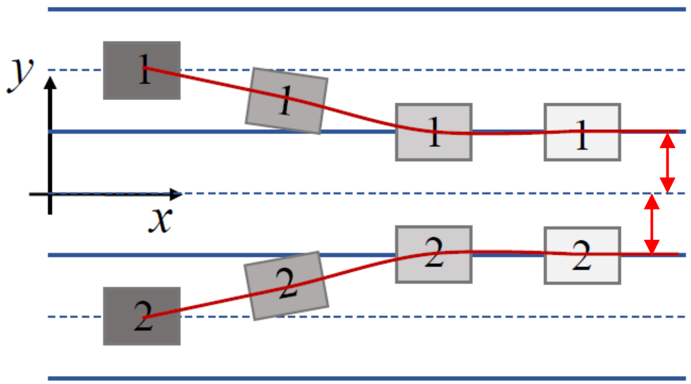


Without slack variable *With slack variable*

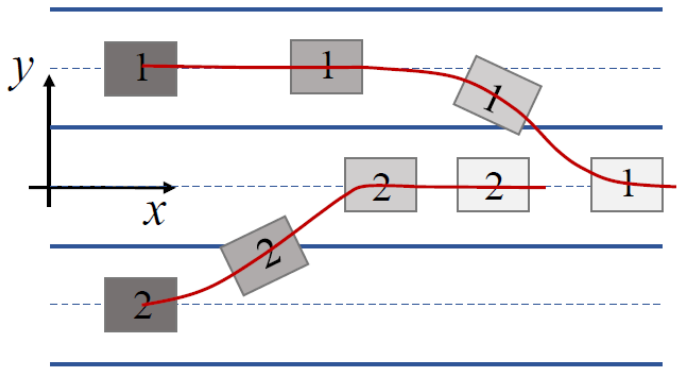
J. Chen, C. Liu, and M. Tomizuka, "Foad: Fast optimization-based autonomous driving motion planner," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 4725–4732.

CFS-DMPC

Deadlock resolution:



(a) Two vehicles in a deadlock situation.



(b) Forming a platoon with the proposed deadlock resolution.

Criteria to change desired speed:

$$\left|\max\{d(\mathbf{x}_i^{-n}, \mathbf{x}_i^{ref})\} - \min\{d(\mathbf{x}_i^{-n}, \mathbf{x}_i^{ref})\}\right| \le \epsilon_1 \wedge \left|\mathrm{mean}\{d(\mathbf{x}_i^{-n}, \mathbf{x}_i^{ref})\}\right| \ge \epsilon_2$$

where $x_i^{-n} = [x_i^{H-n+1}; x_i^{H-n+2}; \ldots; x_i^{H}]$ is the last n points of the planned trajectory, and ϵ_1 and ϵ_2 are tunable thresholds.

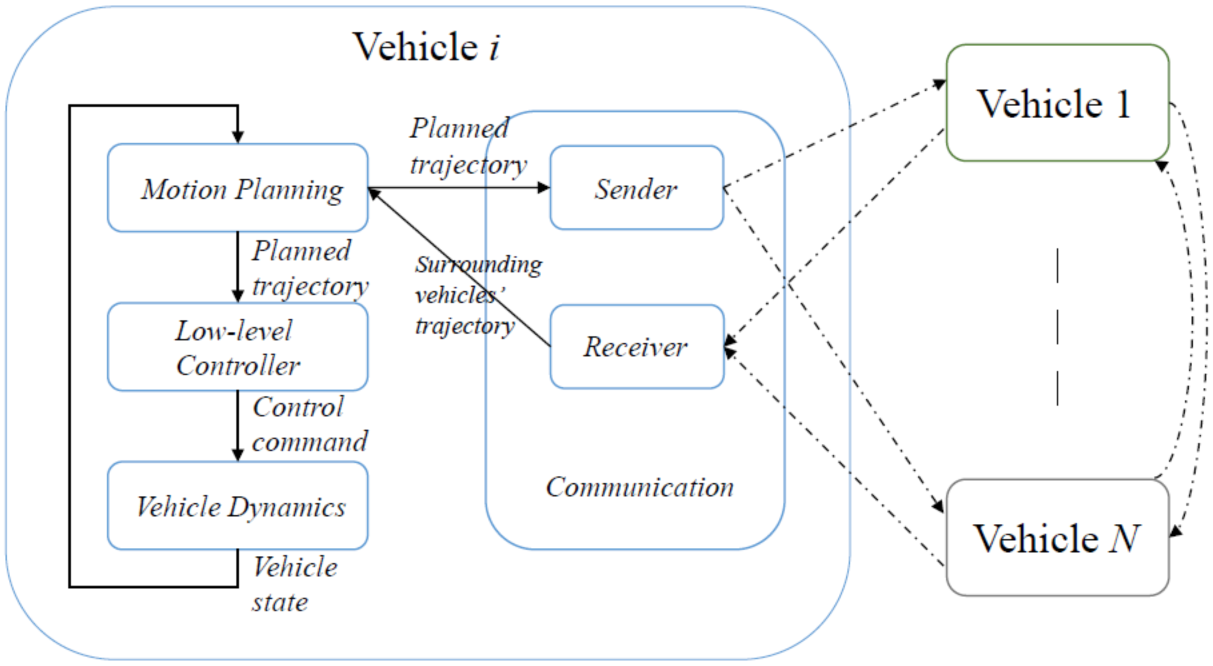
CFS-DMPC

Algorithm and system architecture:

Algorithm 1: The CFS-DMPC design for vehicle i

Input: x_i^c, $\mathbf{x}_j$, $\forall j \in \mathcal{V}\backslash\{i\}$
Parameter: c_o, c_a, c_s, T_r, T_s, H, l, w, r, n, ϵ_1, ϵ_2
Output: $\mathbf{x}_i$

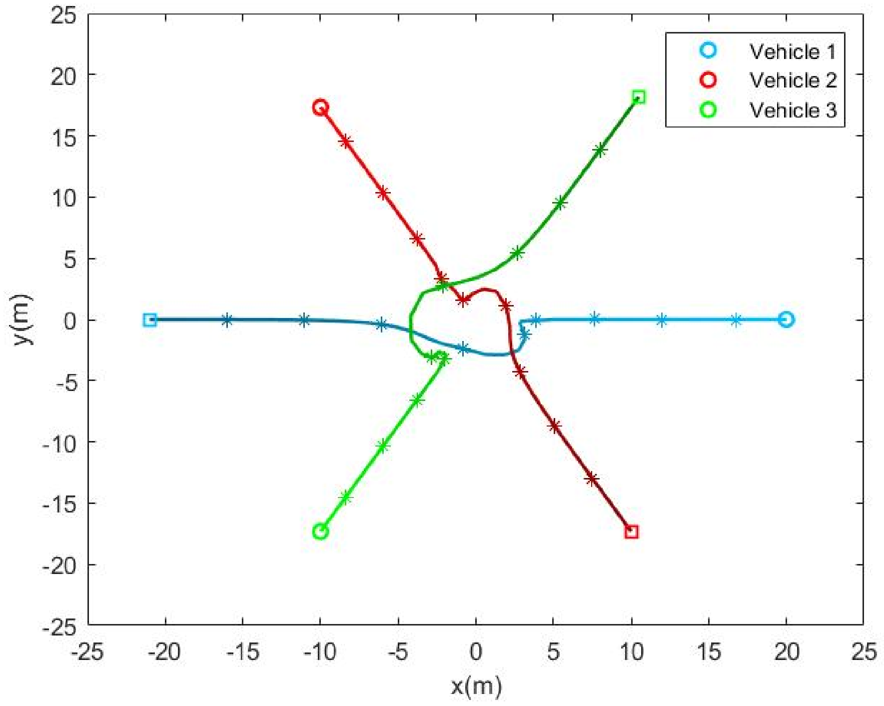
```
Initialize x_i, x_i^ref;
for t = 0, T_r, 2T_r, ..., ∞ do
    Communication with vehicle j, ∀j ∈ V\{i} : send x_i
      and receive x_j;
    Check deadlocks and change the desired speed
      accordingly;
    Modify x_i^ref according to x_i^c and the desired speed;
    Initialize x_i^(0) with x_i from the previous planning;
    Solve optimization problem (9) for x_i.
end
```



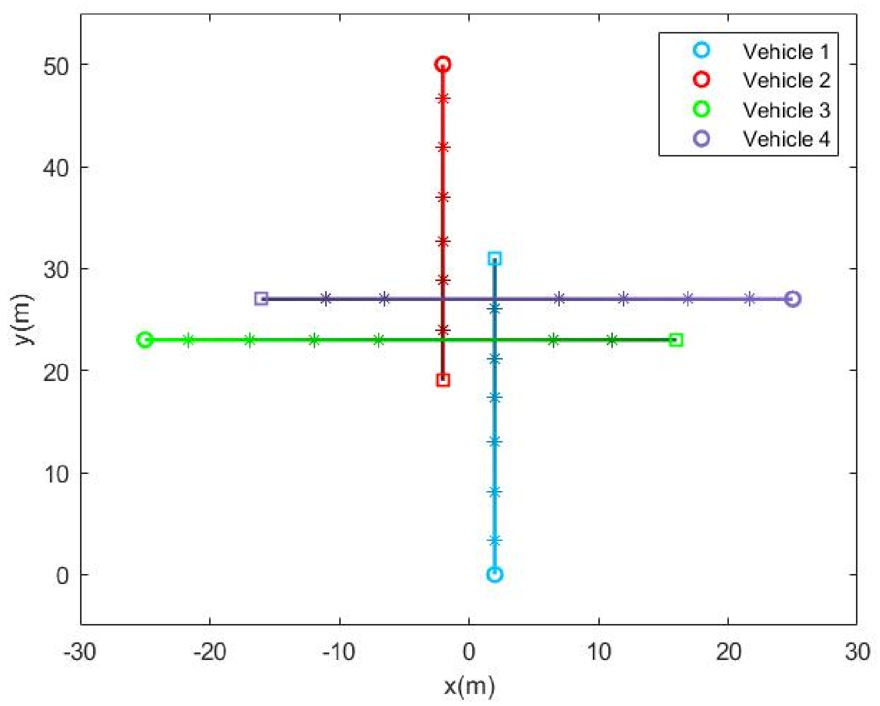
CFS-DMPC

Simulation (without tracking control):

- Unstructured env.
  (point-to-point transition on a circle)



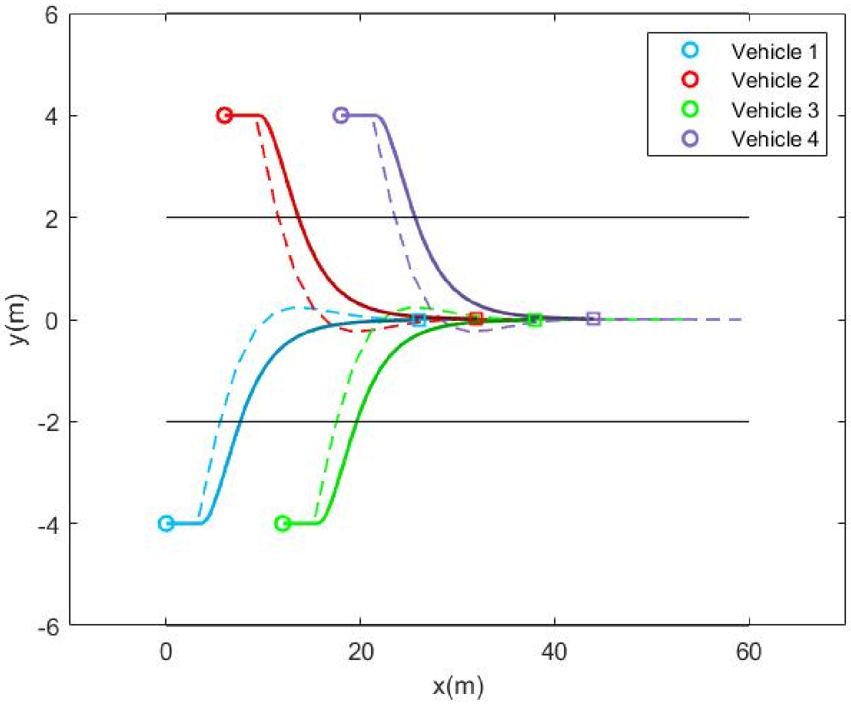
- Intersection



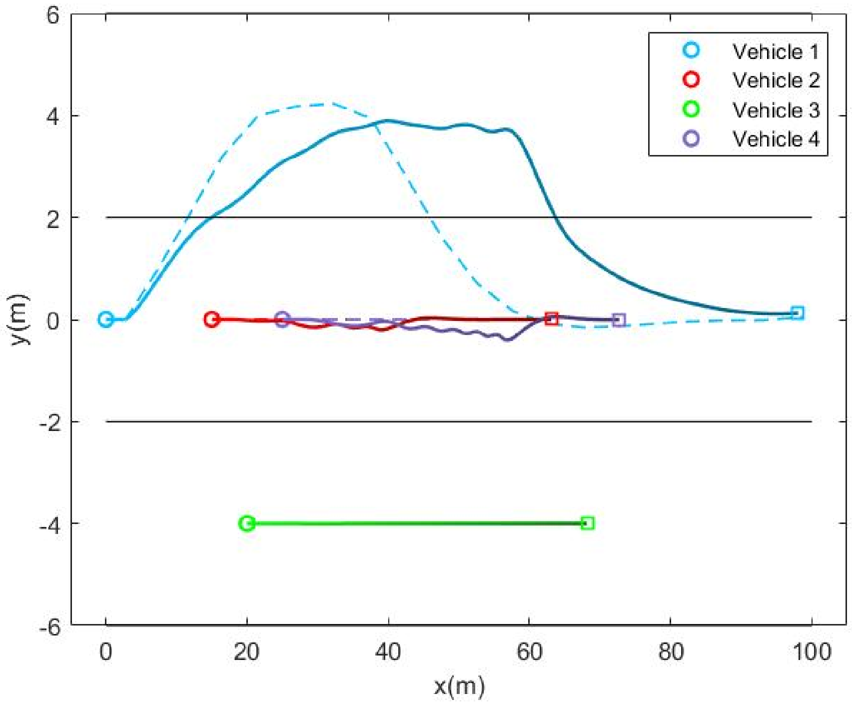
CFS-DMPC

Simulation (with tracking control):

- Platoon formation



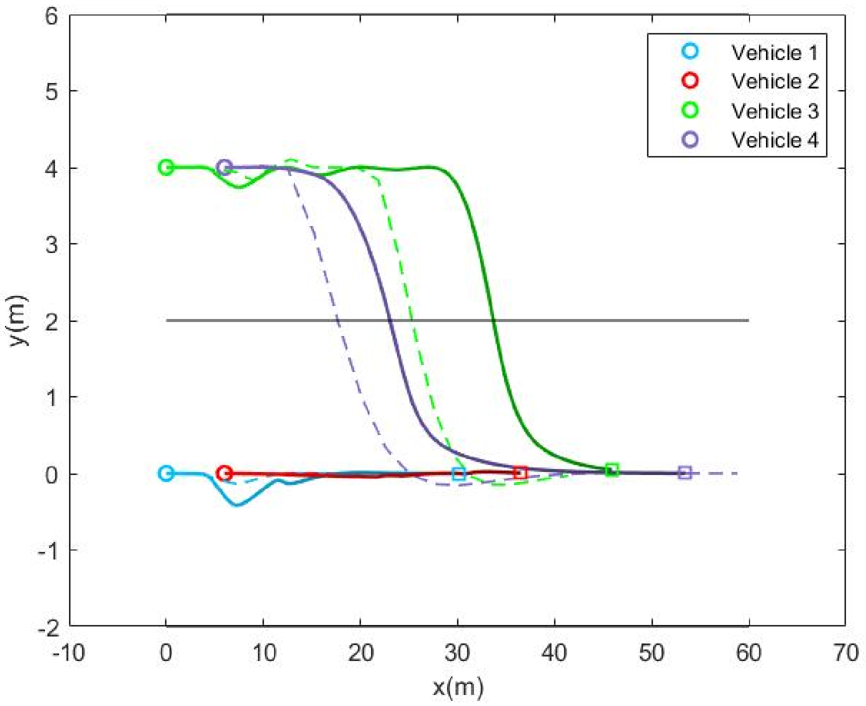
- Overtaking



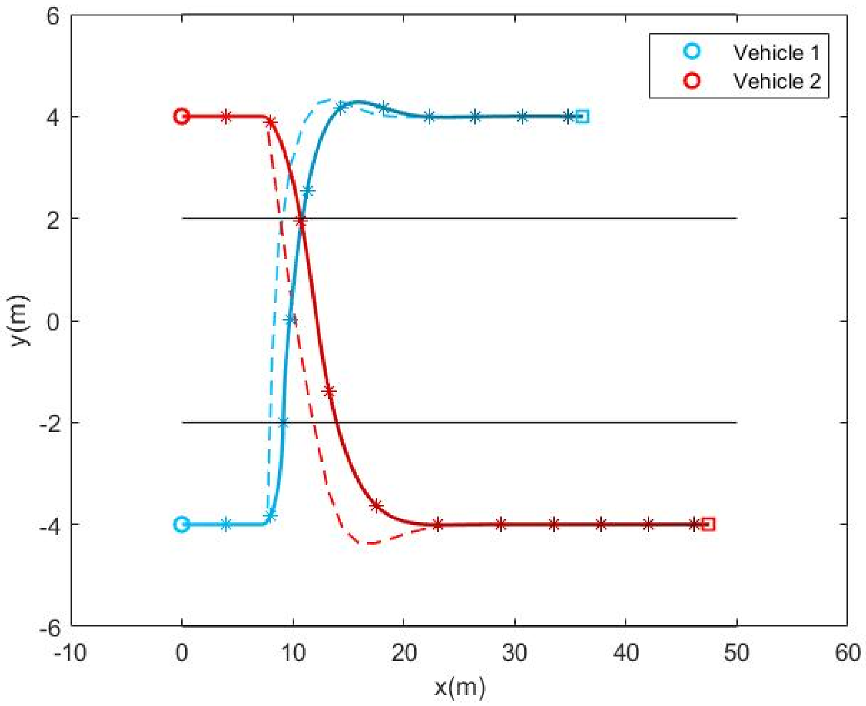
CFS-DMPC

Simulation (with tracking control):

- Merging



- Crossing



CFS-DMPC

Comparison on efficiency:

TABLE I

COMPUTATION TIME (IN SECOND) FOR CENTRALIZED AND DISTRIBUTED APPROACHES.

No.	Centralized		Distributed		
	Avg.	Max.	Avg. (Each)	Avg. (Total)	Max. (Total)
2	0.1853	0.2293	0.0048	0.0096	0.0124
3	0.4313	0.4726	0.0091	0.0272	0.0381
4	0.7768	1.1257	0.0129	0.0514	0.0700
5	1.2780	1.7829	0.0183	0.0913	0.1667

J. Huang and C. Liu, "Multi-car convex feasible set algorithm in trajectory planning," in *Dynamic Systems and Control Conference. American Society of Mechanical Engineers, 2020.*

CFS-DMPC

Comparison on optimality:

TABLE II

THE TOTAL COST FOR BOTH CENTRALIZED AND DISTRIBUTED APPROACHES WITH AND WITHOUT TRACKING ERRORS.

No.	w/		w/o		Distr.		
	Centr.	Distr.	Centr.	Distr.	w/	w/o	Ratio ($\frac{w/}{w/o}$)
2	-19.41	-16.32	-18.25	-9.13	-11.22	-18.12	61.93%
3	-34.99	-28.21	-30.35	-15.17	-25.82	-37.98	67.99%
4	-55.44	-43.06	-44.71	-22.34	-49.30	-67.92	72.58%
5	-81.49	-61.23	-61.56	-30.75	-83.80	-110.10	76.12%

Distributed MPC is more time-efficient but sacrifices optimality

Robustness to tracking error, which causes lost of optimality

J. Huang and C. Liu, "Multi-car convex feasible set algorithm in trajectory planning," in *Dynamic Systems and Control Conference American Society of Mechanical Engineers, 2020.*

CFS-DMPC

Comparison with RVO in unstructured env.:

TABLE III

COMPARISON BETWEEN RVO AND CFS-DMPC.

No.	Avg. Length (m)		Time Duration (s)		Avg. Computation Time (s)	
	RVO	MPC	RVO	MPC	RVO (Each)	MPC (Each)
2	42.57	41.61	5.5	5.3	0.0015	0.0022
4	44.85	48.63	8.2	6.4	0.0022	0.0052
6	43.97	45.85	7.2	5.7	0.0027	0.0083

+: *more time optimal*

-: *longer computation time*

J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 1928–1935.

Outline

- Motivation
- Contributions
- Introduction to Convex Feasible Set Algorithm
- Convex Feasible Set Based Distributed Model Predictive Control
- **Conclusions and Future Work**

Conclusions and Future Work

Conclusions

- Implemented CFS in distributed model predictive control for multi-vehicle coordination
- Proposed a deadlock resolution by changing a vehicle’s desired speed
- Simulation results showed the efficiency and robustness

Future work

- Conduct real-work experiment
- Analyze theoretical stability and robustness

Thank you!

Q&A