Reciprocal Safety Velocity Cones for Decentralized Collision Avoidance in Multi-Agent Systems

Soulaimane Berkane[1] **Dimos V. Dimarogonas**[2]

[1]Depart. of Computer Sci. and Eng., University of Quebec in Outaouais

[2]Division of Decision and Control Systems, KTH Royal Institute of Technology

9–14 July 2023

Motivation & Introduction

Motivation & Introduction



- Multi-agent systems are able to **efficiently carry out tasks that are either impossible or inefficient to carry out** using single-agent systems.
- **Inter-agent collision avoidance** is a central problem
- **Objective:** propose a **sensor-based** collision avoidance strategy.

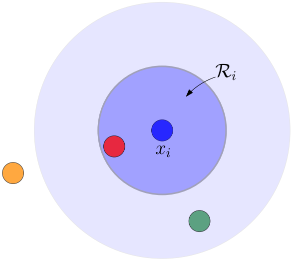
Problem Formulation

Problem Formulation

Geometry and Dynamics

Consider a set of N **spherical agents** $\mathcal{O}_i$ evolving in the Euclidean space $\mathbb{R}^n$ with first-order dynamics:

$$\dot{x}_i = u_i$$



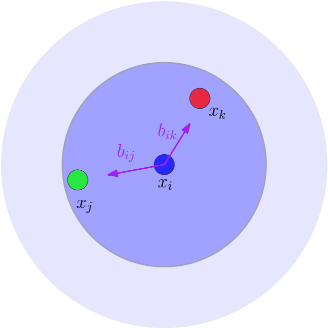
- $\mathcal{R}_i$: avoidance region (where the **avoidance maneuver is activated**)
- Sensing region in light blue

Problem Formulation

Measurements

Neighbouring agents

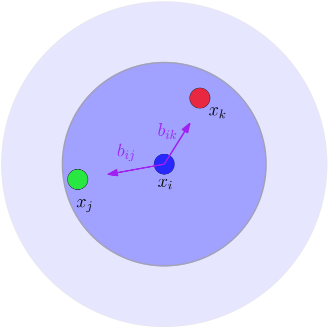
$$\mathcal{N}_i := \{j \in \mathbb{I} : \mathcal{O}_j \cap \mathcal{R}_i \neq \emptyset\}$$



Assumption

Each agent has access to the **inter-agent bearings**

$$b_{ij} := \frac{x_i - x_j}{\|x_i - x_j\|}, \quad j \in \mathcal{N}_j.$$



Assumption

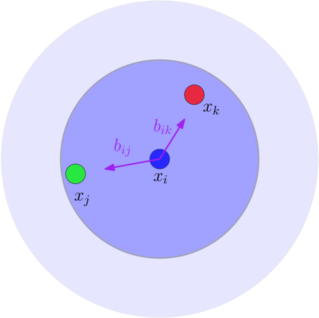
Each agent has access to the **inter-agent bearings**

$$b_{ij} := \frac{x_i - x_j}{\|x_i - x_j\|}, \quad j \in \mathcal{N}_j.$$

Problem Formulation

Objectives

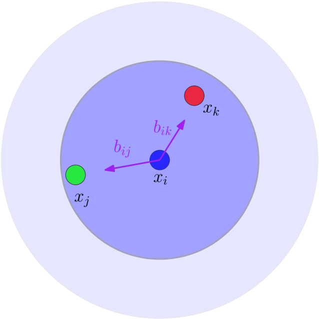
- Purely **decentralized** scheme
- **No explicit communication** with other agents
- Neighbours' velocities are **unknown**
- All agents execute the same control strategy (cooperation).



Problem Formulation

Objectives

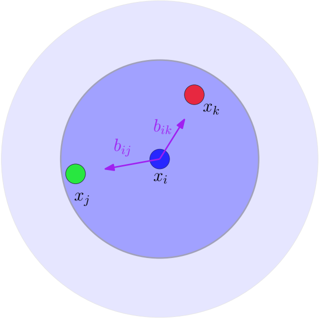
- Purely **decentralized** scheme
- **No explicit communication** with other agents
- Neighbours' velocities are **unknown**
- All agents execute the same control strategy (cooperation).



Problem Formulation

Objectives

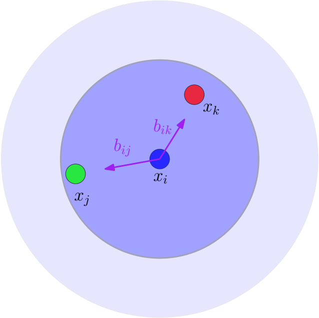
- Purely **decentralized** scheme
- **No explicit communication** with other agents
- Neighbours' velocities are **unknown**
- All agents execute the same control strategy (cooperation).



Problem Formulation

Objectives

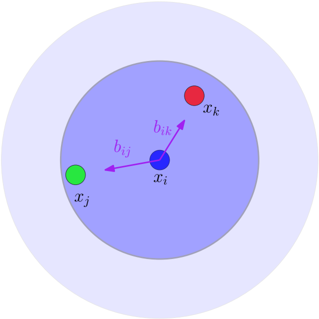
- Purely **decentralized** scheme
- **No explicit communication** with other agents
- Neighbours' velocities are **unknown**
- All agents execute the same control strategy (cooperation).



Problem Formulation

Objectives

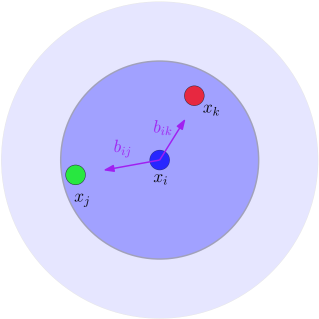
- Purely **decentralized** scheme
- **No explicit communication** with other agents
- Neighbours' velocities are **unknown**
- All agents execute the same control strategy (cooperation).



Problem Formulation

Objectives

- Purely **decentralized** scheme
- **No explicit communication** with other agents
- Neighbours' velocities are **unknown**
- All agents execute the same control strategy (cooperation).



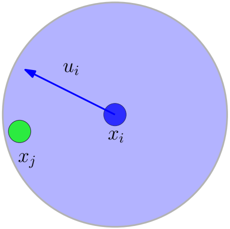
Objective

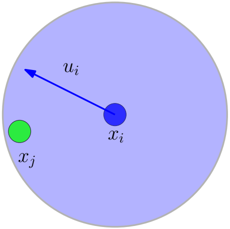
Asymptotically stabilize *the position of each agent i to a desired position* x_i^d *while* **avoiding collision** *with other agents.*

Proposed Approach

Proposed Approach

Velocity Constraints





- Assuming $\dot{x}_j = 0$ (**static neighbour**)

$$\frac{1}{2}\frac{d}{dt}\|x_i - x_j\|^2 = (x_i - x_j)^\top \mathbf{u_i}$$

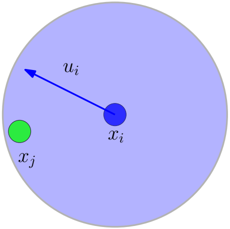
- We need $(x_i - x_j)^\top \mathbf{u_i} \geq 0$ to **ensure safety**.
- The following **velocity constraint** is imposed each agent i:

$$(x_i - x_j)^\top \mathbf{u_i} \geq 0, \quad j \in \mathcal{N}_i$$

$$\iff \boxed{b_{ij}^\top \mathbf{u_i} \geq 0, \quad j \in \mathcal{N}_i}$$

Proposed Approach

Velocity Constraints



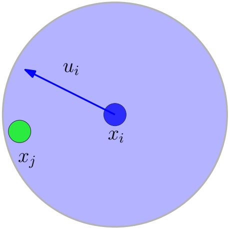
- Assuming $\dot{x}_j = 0$ (**static neighbour**)

$$\frac{1}{2}\frac{d}{dt}\|x_i - x_j\|^2 = (x_i - x_j)^\top \mathbf{u_i}$$

- We need $(x_i - x_j)^\top \mathbf{u_i} \geq 0$ to **ensure safety**.
- The following **velocity constraint** is imposed each agent i:

$$(x_i - x_j)^\top \mathbf{u_i} \geq 0, \quad j \in \mathcal{N}_i$$

$$\iff \boxed{b_{ij}^\top \mathbf{u_i} \geq 0, \quad j \in \mathcal{N}_i}$$



- Assuming $\dot{x}_j = 0$ (**static neighbour**)

$$\frac{1}{2}\frac{d}{dt}\|x_i - x_j\|^2 = (x_i - x_j)^\top \mathbf{u_i}$$

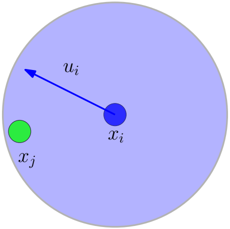
- We need $(x_i - x_j)^\top \mathbf{u_i} \geq 0$ to **ensure safety**.
- The following **velocity constraint** is imposed each agent i:

$$(x_i - x_j)^\top \mathbf{u_i} \geq 0, \quad j \in \mathcal{N}_i$$

$$\iff \boxed{b_{ij}^\top \mathbf{u_i} \geq 0, \quad j \in \mathcal{N}_i}$$

Proposed Approach

Velocity Constraints



- If $i \in \mathcal{N}_j$ and $j \in \mathcal{N}_i$ (*i.e.*,**mutual neighbours**)

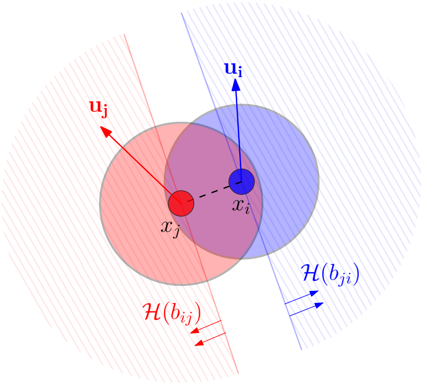
$$\frac{1}{2}\frac{d}{dt}\|x_i - x_j\|^2 = \underbrace{(x_i - x_j)^\top \mathbf{u_i}}_{\geq 0} + \underbrace{(x_j - x_i)^\top \mathbf{u_j}}_{\geq 0}$$

- Velocity constraint is **reciprocally** imposed among mutually neighbouring agents $\Longrightarrow$ **safety is guaranteed.**

Definition (Halfspace)

Let $\nu \in \mathbb{R}^n \setminus \{0\}$, we define the (closed) halfspace normal to ν as

$$\mathcal{H}(\nu) := \{x \in \mathbb{R}^n : \nu^\top x \leq 0\}.$$



$$\boxed{u_i \in \mathcal{H}(b_{ji}), \quad j \in \mathcal{N}_i}$$

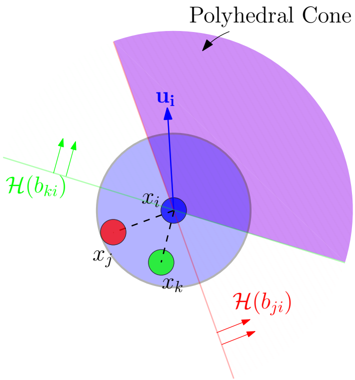
Proposed Approach

Reciprocal Safety Velocity Cone (RSVC)

- Velocity constraint:

$$u_i \in \bigcap_{j \in \mathcal{N}_i} \mathcal{H}(b_{ji})$$

- This defines a (convex) **polyhedral cone** in $\mathbb{R}^n$.



Definition (Polyhedral Cone)

For any matrix A, the set $\mathcal{C}(A) := \{x \in \mathbb{R}^n : Ax \leq 0\}$ defines a polyhedral cone.

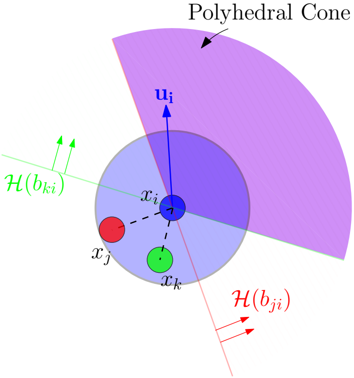
Proposed Approach

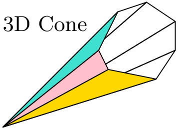
Reciprocal Safety Velocity Cone (RSVC)

- Velocity constraint:

$$\boxed{u_i \in \mathcal{C}(\mathbf{A}(x_{\mathcal{N}_i}))}$$

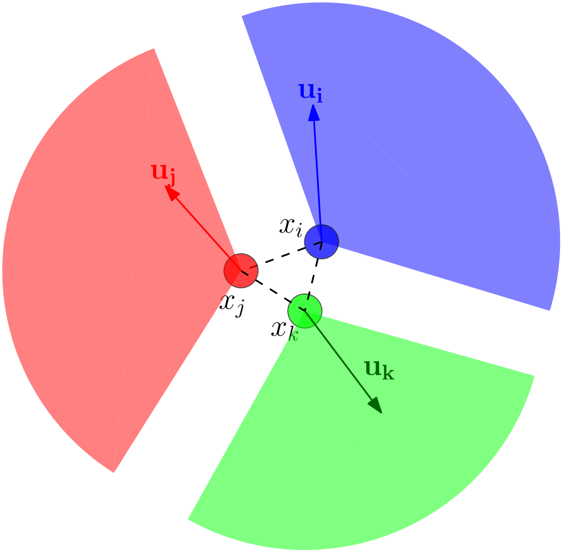
  where
  $\mathbf{A}(x_{\mathcal{N}_i}) := \begin{bmatrix} b_{j_1 i} & b_{j_2 i} & \cdots \end{bmatrix}^\top$
  with $j_k \in \mathcal{N}_i$.
- All agents are assumed to impose the same constraint (**reciprocity**)
- This cone is termed: **reciprocal safety velocity cone (RSVC)**





Proposed Approach

Reciprocal Safety Velocity Cone (RSVC)



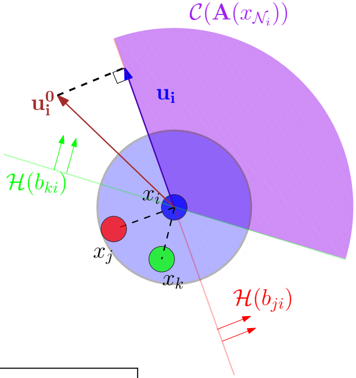
Proposed Approach

Projection onto the RSVC

- **Nominal velocity** for each agent:

  $$u_i^0 := -k_i(x_i - x_i^d), \quad k_i > 0.$$

- Project the nominal controller u_i^0 onto the closed convex cone $\mathcal{C}(\mathbf{A}(x_{\mathcal{N}_i}))$.
- Equivalent to solving the following **quadratic program (QP)**:



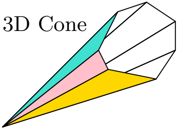
$$\min_{u_i} \frac{1}{2}\|u_i - u_i^0\|^2 \quad \text{subject to } u_i \in \mathcal{C}(\mathbf{A}(x_{\mathcal{N}_i}))$$

Lemma (Rutkowski 2017)

There exists a submatrix of A (formed of rows of A), such that

$$\Pi(\mathcal{C}(A), x) = (I_n - \bar{A}^\top (\bar{A}\bar{A}^\top)^{-1} \bar{A})x.$$

3D Cone



Algorithm 1 Projection onto a polyhedral cone

Require: a matrix $A \in \mathbb{R}^{m\times n}$ and a vector $x \in \mathbb{R}^n$
Ensure: $y = \Pi(\mathcal{C}(A), x)$
  if $Ax \leq 0$ **then** $y = x$
  else
    Let $\Delta = \{\mathcal{I}_1, \mathcal{I}_2, \cdots\}$ ▷ Collection of all non-empty subsets $\mathcal{I}_k$ of $\{1, \cdots, m\}$ with cardinality $\leq \operatorname{rank}(A)$.
    for $k \in \{1, 2, \cdots\}$ **do**
      $\bar{A} = A[\mathcal{I}_k]$
      if $\det(\bar{A}) \neq 0$ **then**
        Solve $\bar{A}\bar{A}^\top \nu = \bar{A}x$ for ν
        $y = x - \bar{A}^\top \nu$
        if $\nu > 0$ and $y \in \mathcal{C}(A)$ **then** return y
        end if
      end if
    end for
  end if

K. E. Rutkowski, "Closed-Form Expressions for Projectors Onto Polyhedral Sets in Hilbert Spaces," *SIAM Journal on Optimization*, vol. 27, no. 3, pp. 1758–1771, 2017.

- Only **relative bearing** is needed
- Other agents are assumed to **cooperate**
- **Explicit expression** of the projection operator

Algorithm 2 Collision Avoidance with Stabilization

Require: Current position x_i and reference position x_i^d
Ensure: Results of Theorem 1.
  Detect neighbouring agents set $\mathcal{N}_i$
  Compute inter-agents bearings b_{ij} with $j \in \mathcal{N}_i$
  Construct bearing matrix $\mathbf{A}(x_{\mathcal{N}_i})$ in (11).
  Calculate nominal controller $u_i^0(x_i, x_i^d)$ in (12).
  Calculate projection $\Pi(\mathcal{C}(\mathbf{A}(x_{\mathcal{N}_i})), u_i^0(x_i, x_i^d))$ using Algorithm 1.
  Assign agent's velocity $u_i = \Pi(\mathcal{C}(\mathbf{A}(x_{\mathcal{N}_i})), u_i^0(x_i, x_i^d))$.

Stability & Safety Analysis

Stability & Safety Analysis

Closed-Loop System

- Define the vector $\mathbf{x} := [x_1^\top, \cdots, x_N^\top]^\top \in \mathbb{R}^{nN}$.
- Multi-agent closed-loop system:

$$\dot{\mathbf{x}} = \begin{bmatrix} \Pi(\mathcal{C}(\mathbf{A}(x_{\mathcal{N}_1})), u_1^0(x_1, x_1^d)) \\ \vdots \\ \Pi(\mathcal{C}(\mathbf{A}(x_{\mathcal{N}_N})), u_N^0(x_N, x_N^d)) \end{bmatrix} =: f(\mathbf{x}). \qquad (1)$$

- **Note**: $f(\mathbf{x})$ is discontinuous whenever agents enter/leave the avoidance region of other agents.
- Collision avoidance is equivalent to forward invariance of the set

$$\mathcal{W} := \{\mathbf{x} \in \mathbb{R}^{nN} : \|x_i - x_j\| > r_i + r_j, \forall i, j \in \mathbb{I}, i \neq j\}. \qquad (2)$$

- **Global target:**

$$\mathbf{x}_d := [(x_1^d)^\top, \cdots, (x_N^d)^\top]^\top \in \mathbb{R}^{nN}.$$

- Define the vector $\mathbf{x} := [x_1^\top, \cdots, x_N^\top]^\top \in \mathbb{R}^{nN}$.
- Multi-agent closed-loop system:

$$\dot{\mathbf{x}} = \begin{bmatrix} \Pi(\mathcal{C}(\mathbf{A}(x_{\mathcal{N}_1})), u_1^0(x_1, x_1^d)) \\ \vdots \\ \Pi(\mathcal{C}(\mathbf{A}(x_{\mathcal{N}_N})), u_N^0(x_N, x_N^d)) \end{bmatrix} =: f(\mathbf{x}). \quad (1)$$

- **Note**: $f(\mathbf{x})$ is discontinuous whenever agents enter/leave the avoidance region of other agents.
- Collision avoidance is equivalent to forward invariance of the set

$$\mathcal{W} := \{\mathbf{x} \in \mathbb{R}^{nN} : \|x_i - x_j\| > r_i + r_j, \forall i, j \in \mathbb{I}, i \neq j\}. \quad (2)$$

- **Global target:**

$$\mathbf{x}_d := [(x_1^d)^\top, \cdots, (x_N^d)^\top]^\top \in \mathbb{R}^{nN}.$$

Stability & Safety Analysis

Closed-Loop System

- Define the vector $\mathbf{x} := [x_1^\top, \cdots, x_N^\top]^\top \in \mathbb{R}^{nN}$.
- Multi-agent closed-loop system:

$$\dot{\mathbf{x}} = \begin{bmatrix} \Pi(\mathcal{C}(\mathbf{A}(x_{\mathcal{N}_1})), u_1^0(x_1, x_1^d)) \\ \vdots \\ \Pi(\mathcal{C}(\mathbf{A}(x_{\mathcal{N}_N})), u_N^0(x_N, x_N^d)) \end{bmatrix} =: f(\mathbf{x}). \tag{1}$$

- **Note**: $f(\mathbf{x})$ is discontinuous whenever agents enter/leave the avoidance region of other agents.
- Collision avoidance is equivalent to forward invariance of the set

$$\mathcal{W} := \{\mathbf{x} \in \mathbb{R}^{nN} : \|x_i - x_j\| > r_i + r_j, \forall i, j \in \mathbb{I}, i \neq j\}. \tag{2}$$

- **Global target:**

$$\mathbf{x}_d := [(x_1^d)^\top, \cdots, (x_N^d)^\top]^\top \in \mathbb{R}^{nN}.$$

Stability & Safety Analysis

Closed-Loop System

- Define the vector $\mathbf{x} := [x_1^\top, \cdots, x_N^\top]^\top \in \mathbb{R}^{nN}$.
- Multi-agent closed-loop system:

$$\dot{\mathbf{x}} = \begin{bmatrix} \Pi(\mathcal{C}(\mathbf{A}(x_{\mathcal{N}_1})), u_1^0(x_1, x_1^d)) \\ \vdots \\ \Pi(\mathcal{C}(\mathbf{A}(x_{\mathcal{N}_N})), u_N^0(x_N, x_N^d)) \end{bmatrix} =: f(\mathbf{x}). \tag{1}$$

- **Note**: $f(\mathbf{x})$ is discontinuous whenever agents enter/leave the avoidance region of other agents.
- Collision avoidance is equivalent to forward invariance of the set

$$\mathcal{W} := \{\mathbf{x} \in \mathbb{R}^{nN} : \|x_i - x_j\| > r_i + r_j, \forall i, j \in \mathbb{I}, i \neq j\}. \tag{2}$$

- **Global target:**

$$\mathbf{x}_d := [(x_1^d)^\top, \cdots, (x_N^d)^\top]^\top \in \mathbb{R}^{nN}.$$

- Define the vector $\mathbf{x} := [x_1^\top, \cdots, x_N^\top]^\top \in \mathbb{R}^{nN}$.
- Multi-agent closed-loop system:

$$\dot{\mathbf{x}} = \begin{bmatrix} \Pi(\mathcal{C}(\mathbf{A}(x_{\mathcal{N}_1})), u_1^0(x_1, x_1^d)) \\ \vdots \\ \Pi(\mathcal{C}(\mathbf{A}(x_{\mathcal{N}_N})), u_N^0(x_N, x_N^d)) \end{bmatrix} =: f(\mathbf{x}). \tag{1}$$

- **Note**: $f(\mathbf{x})$ is discontinuous whenever agents enter/leave the avoidance region of other agents.
- Collision avoidance is equivalent to forward invariance of the set

$$\mathcal{W} := \{\mathbf{x} \in \mathbb{R}^{nN} : \|x_i - x_j\| > r_i + r_j, \forall i, j \in \mathbb{I}, i \neq j\}. \tag{2}$$

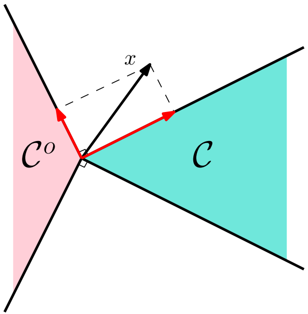
- **Global target:**

$$\mathbf{x}_d := [(x_1^d)^\top, \cdots, (x_N^d)^\top]^\top \in \mathbb{R}^{nN}.$$

Theorem (Moreau Decomposition)

Let $\mathcal{C}$ be a closed convex cone and $\mathcal{C}^o := \{x : x^\top y \leq 0, \ \forall y \in \mathcal{C}\}$ be its polar cone. Then

$$x = \Pi(\mathcal{C}, x) + \Pi(\mathring{\mathcal{C}}, x), \quad \forall x \in \mathbb{R}^n$$



Stability & Safety Analysis

Deadlocks

Corollary

$\Pi(\mathcal{C}, u) = 0$ *iff* $u = 0$ *or* $u \in \mathcal{C}^o$.

Stability & Safety Analysis

Deadlocks

Corollary

$\Pi(\mathcal{C}, u) = 0$ *iff* $u = 0$ *or* $u \in \mathcal{C}^o$.

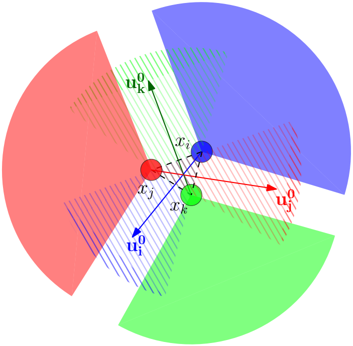
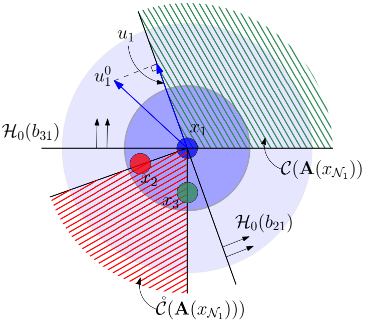


Figure 1: Example of a deadlock.

Theorem

1. *The set $\mathcal{W}$ is forward invariant (collision avoidance).*
2. *All distances $\|x_i - x_i^d\|$ are non-increasing.*
3. *The equilibrium $\mathbf{x}_d$ is locally exponentially stable.*
4. *All solutions must converge to the largest invariant set contained in*

$$\mathcal{E} := \{(x_1, \cdots, x_N) : x_i = x_i^d \text{ or } (x_i^d - x_i) \in \mathring{\mathcal{C}}(\mathbf{A}(x_{\mathcal{N}_i})), \forall i\}.$$



Numerical Simulation

Numerical Simulation

Example without deadlocks

- Two dimensional 1×1 squared region
- 36 agents with $r_i = 0.05$ and $R_i = 0.07$
- Clutteredness $\approx 28.3\%$

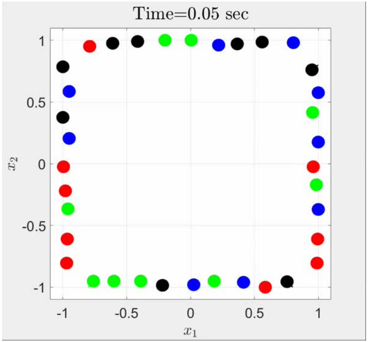


Figure 2: Deadlock-free scenario.

Numerical Simulation

Example with deadlocks

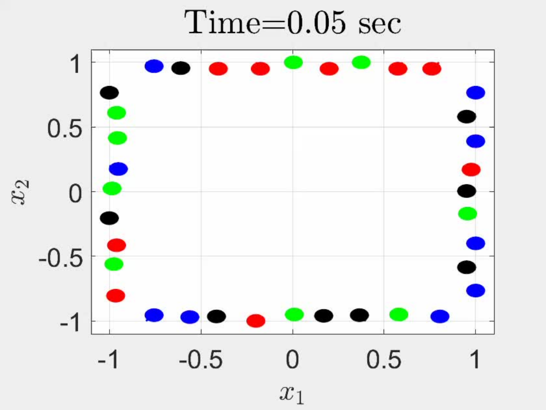


Figure 3: Scenario with deadlocks.

Conclusion & Future Work

Conclusion & Future Work

Conclusion:

- A multi-agent cooperative control strategy with **guaranteed collision avoidance**
- A **sensor-based** approach (based on local inter-agent bearings)

Future Work:

- An approach to **resolve deadlocks**
- smoothing mechanism to **remove discontinuities**
- **High-order dynamics** for the agents

Thank you

Questions?