Reciprocal Collision Avoidance for Quadrotor Helicopters using LQR-Obstacles

Daman F. Bareiss
Jur Van Den Berg

Algorithmic Robotics Laboratory (ARL)

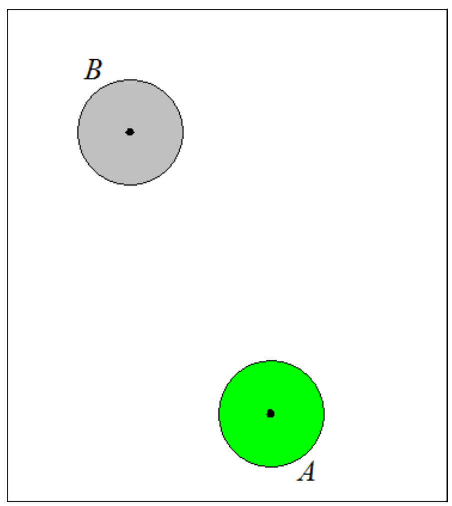
THE UNIVERSITY OF UTAH

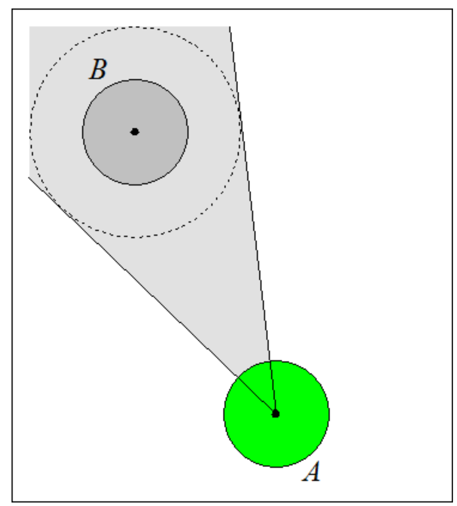
Problem Statement

- Multiple robots with linear dynamics in a common workspace
- Decentralized collision avoidance without communication between the robots
- Similar to humans walking, on a campus for example
- How can it be done?

Velocity Obstacles

- All velocities resulting in a collision between agent A and agent B [Fiorini, Shiller, '98]
- Used for reactive collision avoidance among agents





Control Obstacle

- Need obstacle for robots with dynamics
  - VO’s do not consider robots with dynamics
- Prefer higher-level control obstacle
  - Low-level control obstacle is difficult
- For quadrotors, selecting a position or velocity is much simpler than individual motor thrusts

LQR Feedback Control

- Optimally control robot towards a goal without applying extreme control inputs

- Dynamics: $\mathbf{x}_i[t+1] = A\mathbf{x}_i[t] + B\mathbf{u}_i[t]$
- Cost: $\sum_{t=0}^{\infty}((V\mathbf{x}_i[t] - \mathbf{v}_i^\star)^T Q_v (V\mathbf{x}_i[t] - \mathbf{v}_i^\star) + \mathbf{u}_i[t]^T R\mathbf{u}_i[t])$
- Control Input Minimizing Cost: $\mathbf{u}_i[t] = -L\mathbf{x}_i[t] + E\mathbf{v}_i^\star$
- Higher-Level Control Input:

$$\mathbf{x}_i[t+1] = \tilde{A}\mathbf{x}_i[t] + \tilde{B}\mathbf{v}_i^\star \qquad \tilde{A} = A - BL,\ \tilde{B} = BE$$

LQR Feedback Control

- Closed-Loop Dynamics:

$$\mathbf{x}_i[t] = F[t]\mathbf{x}_i + G[t]\mathbf{v}_i^\star \quad F[t] = \tilde{A}^t,\ G[t] = \sum_{k=0}^{t-1} \tilde{A}^k \tilde{B}$$

$$\mathbf{x}_j[t] = F[t]\mathbf{x}_j + G[t]\mathbf{v}_j^\star$$

- Relative Formulation:

$$\mathbf{x}_{ij}[t] = \mathbf{x}_i[t] - \mathbf{x}_j[t]$$

$$\mathbf{x}_{ij}[t] = F[t]\mathbf{x}_{ij} + G[t]\mathbf{v}_{ij}^\star$$

- Know how to control, but when do the robots collide?

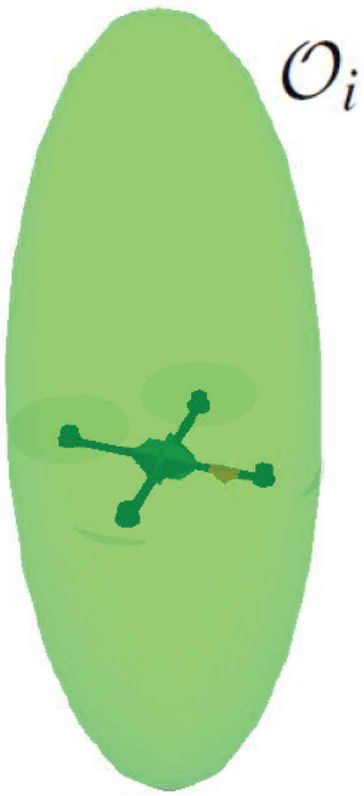
Collision

- Definition:

$$\mathcal{O}_{ij} = \mathcal{O}_j \oplus -\mathcal{O}_i$$

Robot i and robot j *collide* at time t iff

$$C\mathbf{x}_{ij}[t] \in \mathcal{O}_{ij}.$$



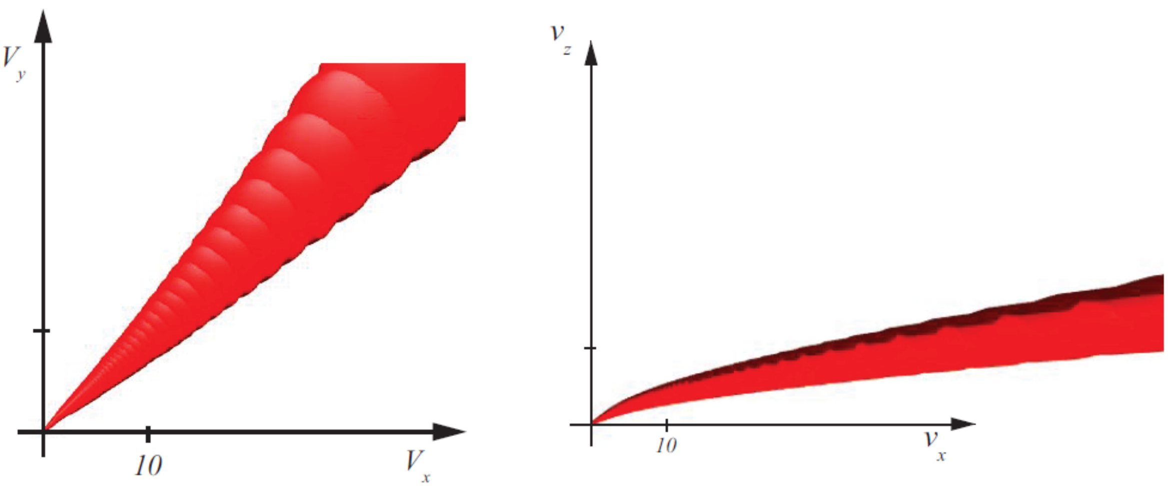
Relative LQR-Obstacles

- Given relative state:

$$C\mathbf{x}_{ij}[t] \in \mathcal{O}_{ij} \qquad \mathbf{x}_{ij}[t] = F[t]\mathbf{x}_{ij} + G[t]\mathbf{v}^{\star}_{ij}$$

$$CF[t]\mathbf{x}_{ij} + CG[t]\mathbf{v}^{\star}_{ij} \in \mathcal{O}_{ij} \iff \mathbf{v}^{\star}_{ij} \in (CG[t])^{-1}(\mathcal{O}_{ij} \oplus \{CF[t]\mathbf{x}_{ij}\})$$

$$\mathcal{LQR}^{\tau}_{ij}(\mathbf{x}_{ij}) = \bigcup_{t=1}^{\tau}(CG[t])^{-1}(\mathcal{O}_{ij} \oplus \{CF[t]\mathbf{x}_{ij}\})$$



Avoiding Collisions

- The LQR-Obstacle defines a set of relative target velocities that would result in collision
- To avoid collision target velocity must not be within that space:

$$\mathbf{v}_i^\star \notin \bigcup_{j \neq i} (\mathcal{LQR}_{ij}^{\tau}(\mathbf{x}_i - \mathbf{x}_j) \oplus \{\mathbf{v}_j^\star\})$$

- Equivalent of VO for robots with dynamics [van den Berg, 2012]

RCA – Pair of Robots

- Only accounts for passive robots, must expand for active robots
- Must consider action of other robot or oscillations in motion can occur
- Designed for each robot to take 50% of the responsibility

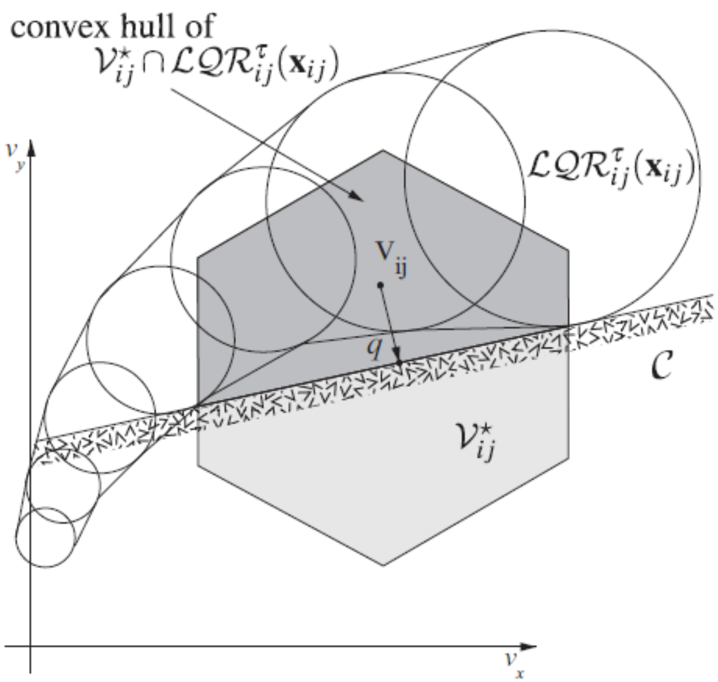
RCA – Pair of Robots

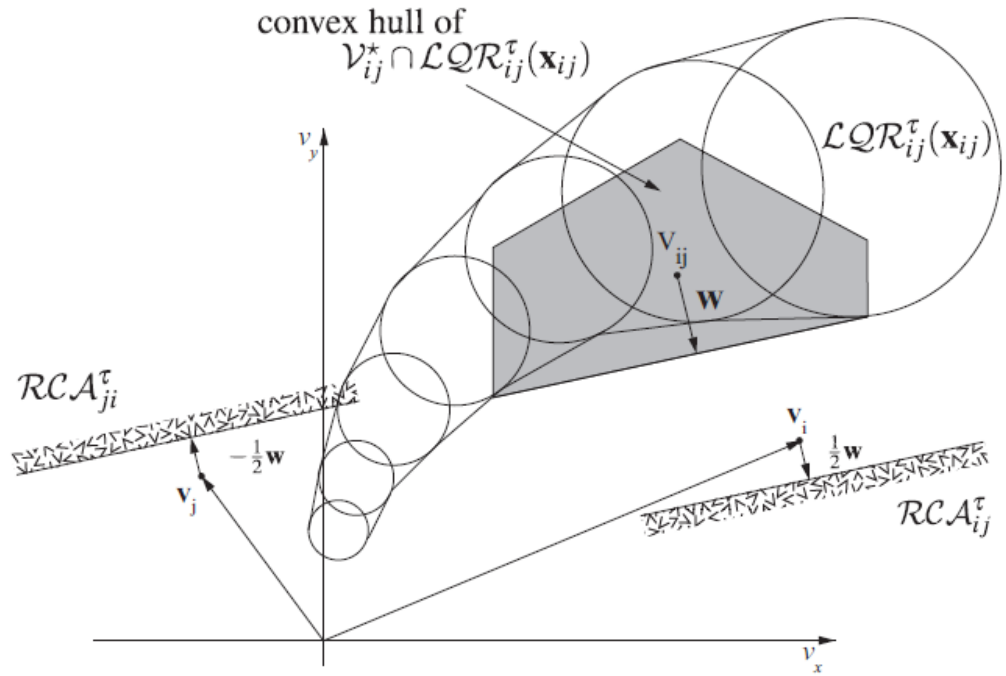
- Relative target velocity must be chosen:

$$\mathbf{v}^{\star}_{ij} \notin \mathcal{LQR}^{\tau}_{ij}(\mathbf{x}_{ij})$$

- Must define set of potential target velocities, or RCA set

RCA – Pair of Robots



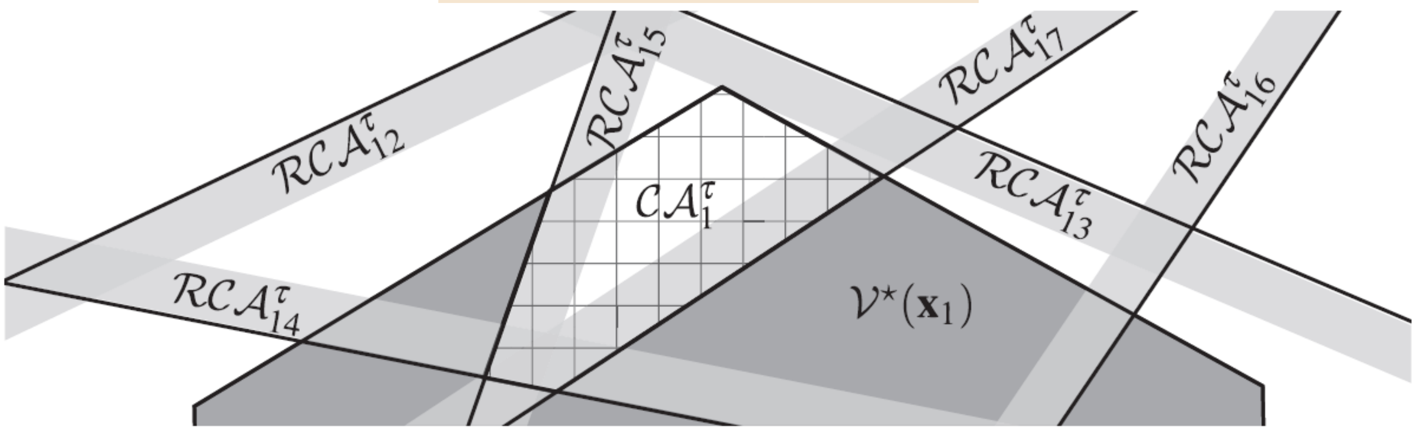


$\mathcal{V}^\star_{ij}$	Possible Relative Control Velocities
$\mathbf{v}_{ij}$	Current Relative Velocities
$\mathcal{LQR}^\star_{ij}(\mathbf{x}_{ij})$	Relative LQR-Obstacle
$\mathcal{C}$	Possible Collision-Free Relative Velocities
$\mathbf{w}$	Vector representing change in $\mathbf{v}_{ij}$ to escape collision
$\mathcal{RCA}^\tau_{ij}$	Collision-avoiding velocity space for robot i
$\mathcal{RCA}^\tau_{ji}$	Collision-avoiding velocity space for robot j

RCA – Multiple Robots

- Each robot creates an RCA with respect to every other robot
- The combination of these creates a target set avoiding collisions with every robot

$$\mathcal{CA}_i^\tau = \bigcap_{j \neq i} \mathcal{RCA}_{ij}^\tau$$



Determining Preferred Velocity

- Given a goal position, what velocity is required?
- Knowing that preferred velocity, find the closest such velocity that avoids collision
- Use of a second layer of LQR control:
  - Cost: $\sum_{t=0}^{\infty}((C\mathbf{x}_i[t]-\mathbf{p}_i^\star)^T Q_p (C\mathbf{x}_i[t]-\mathbf{p}_i^\star)+\mathbf{u}_i[t]^T R\mathbf{u}_i[t])$
  - Substitute initial control policy: $\mathbf{u}_i[t] = -L\mathbf{x}_i[t] + E\mathbf{v}_i^\star$
  - New control policy: $\mathbf{v}_i^\star[t] = \tilde{L}\mathbf{x}_i[t] + \tilde{E}\mathbf{p}_i^\star$

Implementation Details

- C++ Simulator
- Qhull Library for convex hull of ellipsoids
- GJK-Algorithm to find escape velocity
- RVO2 Library for linear programming

- Simulation Computer Specifications:
  - Windows 7 Professional 64-bit
  - Intel i7-2600 CPU, 8GB RAM

Results – 2 Quadrotors

- Videos can be found at:
  - http://arl.cs.utah.edu/research/rca/

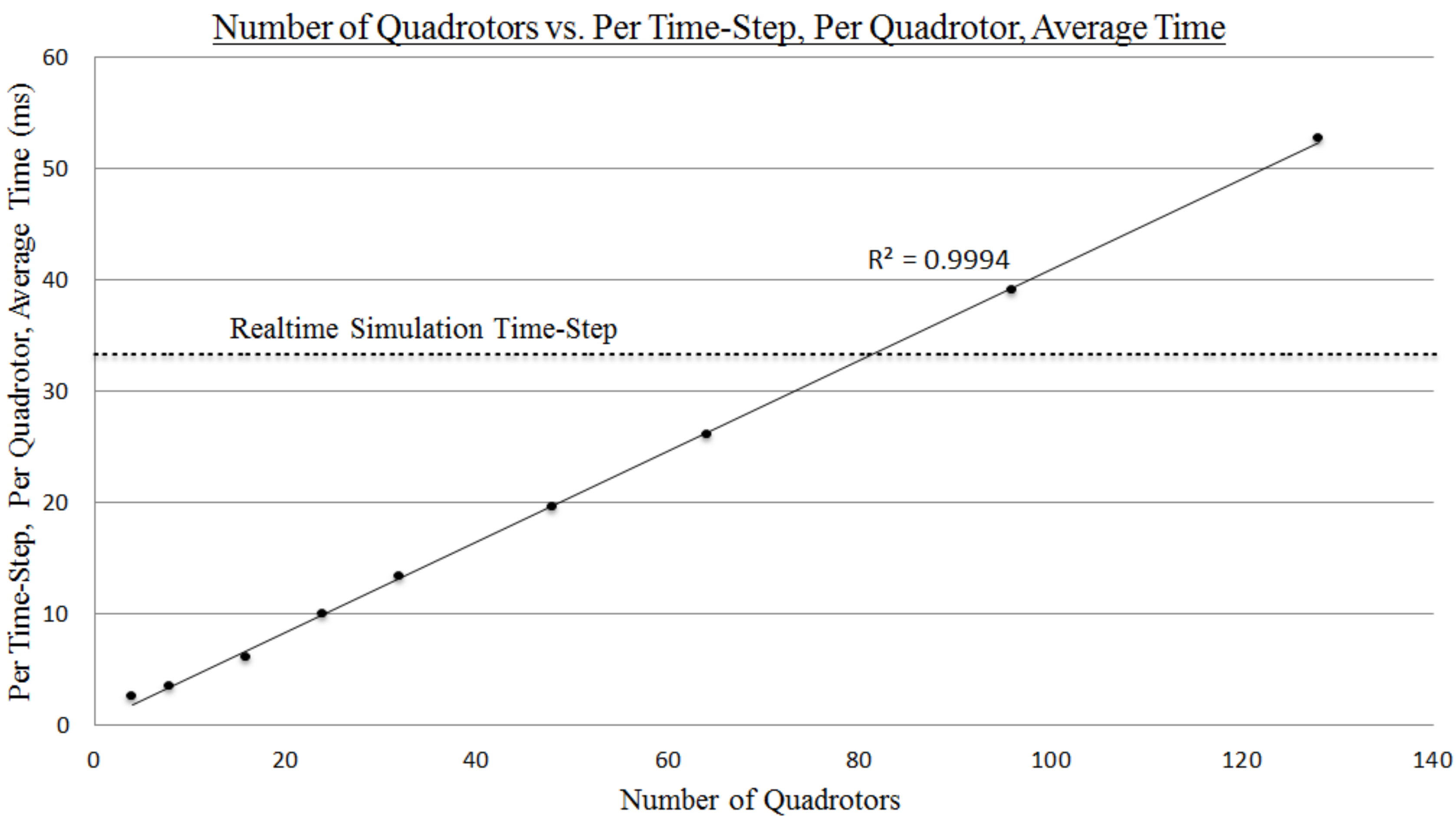
Results – 24 Quadrotors

- Videos can be found at:
  - http://arl.cs.utah.edu/research/rca/

Results – 100 Quadrotors

- Videos can be found at:
  - http://arl.cs.utah.edu/research/rca/

Results



Number of Quadrotors vs. Per Time-Step, Per Quadrotor, Average Time
Per Time-Step, Per Quadrotor, Average Time (ms)
0
10
20
30
40
50
60
Realtime Simulation Time-Step
R² = 0.9994
0
20
40
60
80
100
120
140
Number of Quadrotors

Conclusions

- Simulation results displayed validity of our approach.
- Robots with linear dynamics were able to independently navigate to a goal position with no communication between the robots in real time

Conclusions

- Limitations
  - Requires position and velocity to be contained in the state of the robot
  - Geometry of robot translates but does not rotate
  - Robots of same dynamics
  - Requires full state observation

Future Work

- Expanding algorithm for robots with different dynamics
- Incorporating a state estimator with possible uncertainties
- Implement algorithm on real quadrotors to obtain physical data

Questions