Reading report
Reciprocal n-body collision avoidance[1]

Dharmin Bakaraniya

June 2, 2018

Abstract

In this paper, we present a formal approach to reciprocal n-body collision avoidance, where multiple mobile robots need to avoid collisions with each other while moving in a common workspace. In our formulation, each robot acts fully in- dependently, and does not communicate with other robots. Based on the definition of velocity obstacles[3], we derive sufficient conditions for collision-free motion by reducing the problem to solving a low-dimensional linear program. We test our approach on several dense and complex simulation scenarios involving thousands of robots and compute collision-free actions for all of them in only a few milliseconds. To the best of our knowledge, this method is the first that can guarantee local collision-free motion for a large number of robots in a cluttered workspace.

1 Summary

- The authors state the problem same as they did in reciprocal velocity obstacle[2]. The robots are working in a multi agent environment where they cannot communicate with each other. They need to avoid collision with each other and with other static and moving obstacles. They have their target position and they would like to reach there with their preferred velocity. The information of this target and their preferred velocity is not known to any other robot.

- The robots are assumed to have the information of other robot's current velocity, position and their shape.

- The authors define avoiding velocities for robot A as $CA^{\tau}_{A|B}(V_B)$. Here A and B were on a collision course and they were going to collide within time τ. If A chooses it velocity from the set of avoidance velocity, then it is guaranteed to not collide with B within τ, given that B chooses a velocity from V_B.

- If A and B, both, choose from their avoiding velocities, then V_A and V_B are called *reciprocally collision avoiding* and if V_A and V_B are exactly equal to their avoiding velocity, then they are called *reciprocally maximal*.

- Out of many possibility of par os set V_A and V_B, the authors want the velocity to be as close as possible to their optimal/preferred velocity. These sets are called *Optimal reciprocal collision avoidance velocity*, $ORCA^{\tau}_{A|B}$ for A and $ORCA^{\tau}_{B|A}$ for B.

- The authors assumes **u** to be minimum velocity change that A and B has to perform collectively to avoid each other for at least τ time. Both A and B will change their velocity by $\frac{1}{2}u$ to take equal responsibility in avoiding each other. All the velocity left for A and B are their respective ORCA velocity. ORCA velocity here are half plane starting at point $v_A + \frac{1}{2}u$ in direction of n (where n in normal vector pointing out of $VO_{A|B}$). This is one way to arrive at the definition of ORCA apart from set approach mentioned above.

- For multiple objects B_n, ORCA velocities of A, can be calculated by intersection of half planes from each obstacles.

- Each robot follows a cycle repeatedly

  1. Sense position and velocity of other robots
  2. Calculate ORCA for each robot collision
  3. Select new velocity from the union of ORCAs
  4. Apply chosen velocity and update position
  5. Repeat steps 1–4

- To choose new velocity, the authors are using a linear programming algorithm with complexity O(n), where n is number of constraints. They propose to spped up the process even further by excluding robots that are far away and will not collide with A before time τ.

- For static obstacles, the robots should take full responsibility for avoiding the collision. A static obstacle can be defined as a set of line segments. Each line segment will induce constraints on the ORCA of the robot avoiding it. This way the algorithm remains same and approach covers static obstacles as well.

- The authors apply their approach on 3 experimental scenarios

  - *Exchange position*: Two robot have to exchange their position. This ran successfully in simulation.

  - *Antipodal position*: Numerous robots are placed on a circle, their goal position is the antipodal (directly opposite of them) position on the circle. This was applied to 5 robots and 1000 robots in a simulation.

  - *Evacuation*: 1000 virtual agents had to evacuate a office building. Their global path were planned to reach outside area of the office building. This ran successfully in simulation.

- As there is no communication between the robots, each computation can be performed in parallel. For 5000 agents, anitpodal position experiment takes 8ms per iteration and evacuation experiment takes 15.6ms per iteration on 8 Intel Xeon (2.66 Ghz) cores.

- The authors show that as the number of agents increase, the time taken per iteration increases linearly. They also show that as the number of cores increases, the speedup increases linearly.

2 Scientific contributions

- Local collision avoidance for multi-agent system without inter-communication is performed very fast.

- Experiment of office evacuation for multi agent done successfully.

3 Scientific deficits

- Almost no improvement on reciprocal velocity approach[2] except providing additional experiment and few reasons for choosing optimal velocity.

- No implementation on real robots. Not addressing errors induced due to sensor and actuator errors. The authors mentioned this in future work.

- Kinematic and dynamic constraints are not considered, hence this approach is not applicable on real robots. The authors mentioned this in future work.

References

[1] Van Den Berg, J.; Guy, S. J.; Lin, M. & Manocha, D."Reciprocal n-body collision avoidance" *Robotics research, Springer*, 2011.

[2] Van Den Berg, J.; Lin, M. & Manocha, D. "Reciprocal velocity obstacles for real-time multi-agent navigation" *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on* 2008.

[3] P. Fiorini, Z. Shiller. "Motion planning in dynamic environments using Velocity Obstacles". *Int.Journal of Robotics Research*, 1998.