

Avoiding Collision Logjams through Cooperation and Conflict Propagation

K. Madhava Krishna Henry Hexmoor P Subbarao
Dept. of Computer Science and Computer Engineering
University of Arkansas
Fayetteville AR-72701
mkrishna@uark.edu, hexmoor@uark.edu, spasupu@uark.edu

Abstract— A methodology for resolving collision conflicts of many moving robots involving distributed cooperative behavior is proposed in this work. Robots are unaware of the goals and plans of other robots. Each robot is however assumed to be in knowledge of the current states of other robots within a given range of vision. The current state of a robot is represented by its current and target velocities and its heading direction. A conflict between two robots is a detection of an upcoming collision. The resolution of the conflicts occurs at three levels. At the first level the robots attempt to resolve the conflict individually. If individual attempts to resolve fail the robots enter into a cooperative phase of resolving the conflict at the second level. At the third level conflicts are propagated to robots not directly involved in any conflicts at that instant and their assistance sought for possible resolution. At all the three levels conflict resolution is attempted by velocity control of the robots through an algorithm that is distributed across all the robots in the system. Simulation results are shown to validate the efficacy of this methodology. Conflict resolution through cooperation and conflict propagation is pertinent to all those applications that entail a number of robots to crisscross each other in quick succession or in situations where robots find themselves coming together to get across an intersection from various directions which would otherwise result in a logjam.

1. Introduction

Research in multi-robot systems have been propelled due to advantages arising from improved efficiency, faster responses due to spread of computational burden, augmented efficiencies and also due to research interests in study of emergent behaviors that arise from interaction between individual behaviors. Multiple mobile robot systems find applications in material handling operations in difficult or hazardous terrains [1], fault-tolerant systems [2], covering and guarding of unmanned terrains [3] and in cargo transportation [4] to name a few.

In many of those applications that involve coordination of many robots a scheme for cooperative collision avoidance between them becomes inevitable. Whether it is a case of cooperative navigation of robots in a rescue and relief operation after an earthquake or while searching the various parts of a building or in case of a fully automated shop floor or airports where there are only robots going about

performing various chores, cooperative collision avoidance becomes mandatory simply because robots cannot help crisscrossing each other and often in large numbers and in quick succession.

Most research in multi-robot navigation has focused on issues related with motion planning for robots through centralized [5] and decentralized approaches [6]. Recently a motion planning strategy through dynamic networks [7] that is centralized within a local network of robots but decentralized across networks has been reported. Centralized approach plans for the totality of the robots in a system through a centralized planner, whereas in decentralized approach the planning algorithm is distributed across the robots which results in each robot computing a plan for itself. The tradeoffs between the two approaches would not be discussed here. It could be worthwhile to mention however that an important component of decentralized planning is the plan merging paradigm, PMP. [8,9]. The basic PMP consists of a plan merging operation (PMO), which sees to that a robot's current plan is valid in the multi robot context by considering all potential conflicts with other plans. The PMP also serves as a kind of bridge between central and distributed planning.

This paper deals with multi robot coordination more akin to a real-time reactive setting where each robot is unaware of the complete plans of the other robots and the model of the environment. The work closest to the current reported work in literature we could place our hands upon was a scheme for cooperative collision avoidance (CCA) by Fujimora and others [10]. Their work is based on coming up with a CCA for two robots based on orientation and velocity control and extend this strategy for the multi robot case based on the usual technique of priority based averaging (PBA). However we have proved in an earlier effort of ours [11] that such PBA techniques fail when individual actions that get weighted and averaged in the PBA are conflicting in nature.

The present work is novel and different from others as the resolution of collision conflicts is attempted at three levels, namely the individual, cooperative and propagation levels. In the individual either of the two robots involved in a collision conflict is capable of resolving the conflict independently. The cooperative phase is resorted to when neither of the two robots involved in the conflict is capable of resolving it individually. The cooperative phase

constitutes actions taken by the robots involved in a conflict in a synchronous fashion that facilitates resolution. When joint actions in the cooperative phase are not sufficient for conflict resolution the help of other robots who are in a conflict free state at that instant is sought by the robots in conflict by propagating descriptions of the conflicts to them. When such free robots are also unable to resolve the conflict collision is deemed inevitable.

Conflict resolution is achieved by choosing appropriate velocities from the solution space of velocities. In the individual phase a search for a velocity in the individual solution space would suffice for avoiding conflicts, in the cooperative phase the individual solution spaces are exhausted and a search in the joint space of both the robots yields a possible means of averting conflicts. The propagation phase is resorted when the joint solution space of both the robots is also exhausted. Elsewhere we have shown through mathematical arguments the existence of a cooperative phase in the navigation of a two bodied system and studied the effects of parametric variations on the requirement of cooperative phase in two-bodied as well as multi-bodied systems [12]. Here the discussion would be focused more on the methodology of multi robot avoidance of collision logjams.

It is to be noted that the concept of propagating conflict to robots not a part of the conflict as an element of the overall strategy to multi-robot collision avoidance does not find mention in robotic literature. Such kind of transmission of requests to robots though not invoked frequently is however helpful in resolving a class of conflicts which otherwise would not be possible as simulation results reveal.

2. Formulation

Given a system of N mobile robots, each charting a linear trace towards its goal, the problem is to resolve all possible collision conflicts that occur during the course of navigation between the robots. The robots are allowed to be heterogeneous in terms of their geometric, kinematic and dynamic attributes.

2.1 Detecting direct collision conflicts

Robots are considered as discs of finite radii. The following geometric procedure is used for detecting conflicts. It is assumed that the robots' motion direction is constant and the current state of the robot is represented by (vc, vn, θ) where vc, vn indicate the robot's current and aspiring velocities while θ is indicative of the robot's motion direction. Figure 1 shows two robots R1 and R2 of radii r_1 and r_2 and whose states are (vc_1, vn_1, θ_1) and (vc_2, vn_2, θ_2) respectively. Point C in the figure represents the intersection of the future paths traced by their centers. For purpose of collision detection one of the robots is shrunk to a point while the other is grown by the radius of

the shrunken robot. This scenario is depicted in figure 2 where R1 is depicted as a point and R2 is grown by r_1 and its radius is now r_1+r_2 .

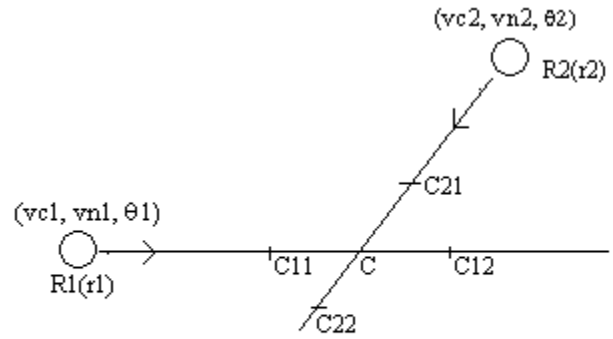


Figure 1: Two robots R1 and R2 with radii r_1 and r_2 along with their current states are shown.

The points of interest in figure 2 are the centers C21 and C22 of R2 where the path traced by the point robot R1 becomes tangential to R2. At all points between C21 and C22 R2 can have a potential collision with R1. C21 and C22 are at distances $(r_1 + r_2) \cos ec(\theta_1 - \theta_2)$ on either side of C. The time taken by R2 to reach C21 and C22 given its current state (vc_2, vn_2, θ_2) is denoted by t_{21} and t_{22} . Similar computations are made for R1 with respect to R2 by making R2 a point and growing R1 by r_2 . Locations C11 and C12 and the time taken by R1 to reach them t_{11} and t_{12} are thus computed. A collision is said to be averted between R1 and R2 if and only if $[t_{11}, t_{12}] \cap [t_{21}, t_{22}] \in \emptyset$

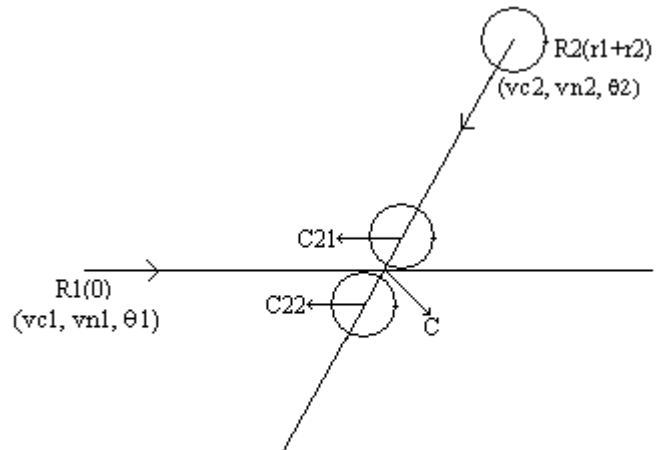


Figure 2: R1 is shrunk to a point while R2 is grown by radius of R1. C21 and C22 are the centers of R2 where the path traced by R1 becomes tangential to R2.

A direct conflict between robots R1 and R2 is said to occur if R1 occupies a space between C11 and C12 when the center of R2 lies between C21 and C22 at some time t .

2.2 Characterizing the Individual Phase

A pair of robots R1 and R2, which have a direct conflict between them are said to be in individual phase of navigation if and only if:

- (i) R1 is able to control its velocity such that it is able to get past C12 before R2 reaches C21 or R1 controls its velocity such that it does not reach C11 before R2 reaches C22.
- (ii) R2 is able to control its velocity such that it is able to get past C22 before R2 reaches C11 or R2 controls its velocity such that it does not reach C21 before R1 reaches C12.

In both the above cases it would suffice that only one of the two robots controls its velocity. This indeed is the crux of the individual phase where at-least one of the two robots is able to individually avoid the conflict without requirement of the other to take part.

2.3 The Cooperative Phase

A pair of robots R1 and R2 are said to be in cooperative phase of navigation if and only if they are able to resolve the collision conflict between the two through either of the following ways:

- (i) R1 is able to get past C12 under maximum acceleration before R2 can get to C21 under maximum deceleration.
- (ii) R2 is able to get past C22 under maximum acceleration before R1 can get to C11 under maximum deceleration.

It is to be noted here that **both** R1 and R2 engage in velocity control in appropriate fashion to circumvent the collision since individual efforts as mentioned in subsection 2.2 had failed.

2.4 Conflict Propagation Phase:

At times robots R1 and R2 are unable to resolve conflicts between them either individually or cooperatively because velocities that resolve the conflict between the two result in conflicts with other robots. For example in figure 3 velocities for robots A and D that resolve the conflict that exists between the two results however in new conflicts with B and C. In such a scenario B and C are requested to aid in solving the conflict between A and D and are brought into a larger scheme of cooperation that involves all the four robots at that instant.

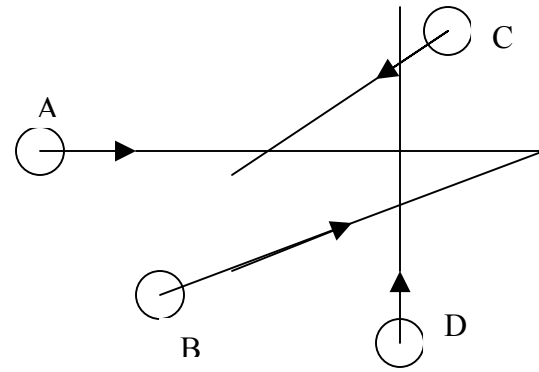


Figure 3: The direct conflict between A and D needs the assistance of B and C for resolution. Hence we say that the conflict between A and D propagates to B and C

3. The Algorithm

Before entering into the outline or the sequential characterization of the algorithm it is useful to define the following terminologies.

Mutual Direct Conflict (MDC): A pair of robots R1, R2 are said to be in MDC with one another if R1's first DC in reaction time t_r is with R2 and R2's first DC in reaction time t_r is with R1.

Non-mutual Direct Conflict (NMDC): A robot R1 is said to be in NMDC with R2 when R1's first DC in t_r is with R2 while R2's first DC in t_r is however not with R1.

Indirect Conflict (IDC): A robot R1 is said to be in IDC with R2 if R1 has no MDC or NMDC at that instant and has received a request for resolving R2's conflict with some other robot R3.

Reaction time (t_r): A robot looks for and tries to resolve the first conflict that occurs within t_r and postpones consideration of other conflicts for future samples. The value of t_r is fixed by the user, however t_r generally follows the equation $t_r \geq (\eta v_m^2 / 2a_M) + t_{samp}$, where v_m and a_M are the robot's maximum velocity and acceleration and t_{samp} is the maximum sampling time between any two samples of the environment by a robot.

Two robots R1 and R2 are very rarely in NMDC for the probability that R1 detects a DC with R2 and R2 detects a DC with some other robot R3 within the same t_r is almost negligible.

The broad steps of the overall algorithm are delineated below. Each step of the algorithm may itself require further

decomposition into several modules and subtasks that are not elaborated here for brevity.

Overall algorithm for conflict resolution in a multi-robot system:

1. Repeat steps 2 down to 6 after every sampling instant for all the robots that have not reached their goal yet.
2. Add to the list, *ListDC*, all robots, a_i , which have a DC with another robot a_j , within reaction time, t_r , which a_i is unable to resolve individually..
3. Find all pairs of robots (a_i, a_j) from *ListDC* that have a *MDC* between them and place them in the list, *ListMDC*
4. For all pairs in *ListMDC*, resolve collision conflicts through a module denoted as CCAM (Cooperative Collision Avoidance Module). Add to list *ListIDC* all those robots that are not in *ListDC* and to whom request for participation in conflict resolution had been transmitted by the robots in *ListMDC* while trying to resolve conflicts through CCAM.
5. For all robots a_i in *ListDC* but not in *ListMDC* resolve conflicts by a module called *ResNMDC* (Resolve Non-Mutual Direct Conflicts)
6. Finally for all robots belonging to *ListIDC* resolve conflicts through module *ResIDC* (Resolve Indirect Conflicts)

The last step of the algorithm dealing with the resolution of IDC is further treated through figure 4. This step deals with robots that have received requests to resolve conflicts of other robots, which is the third step in the overall resolution strategy, also called as conflict resolution phase. In figure 4

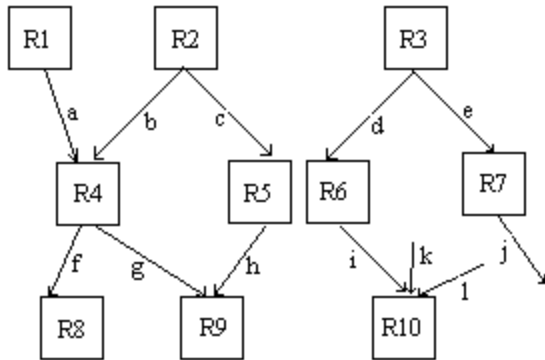


Figure 4: The propagation of conflicts

robots R1, R2 and R3 at the first level of a tree like structure are the ones that are unable to resolve their MDC and propagate it to R4, R5, R6 and R7 at the second level via the links shown. The robots at the second level that receive requests can solve it or propagate to a level below if there are robots willing to accept or return a message to the transmitting robot. A robot can receive requests from more than one robot to participate in its conflict such as R9 receives requests from R4 and R5, in which case R9

prioritizes the requests in order of time to collision of R4 and R5 with the robots with which R4 and R5 are in conflict. If any one of the requests could be resolved no further requests are entertained and pending requests attempted to be resolved. If a request does not get resolved it could be further propagated to robots that are willing to accept. For example in figure 4, R4 propagates R1's request to R9 via link g. When a robot resolves a request all the links connecting to the robot that sourced the request is removed and all other pending requests with their links to the source are also removed. If R9 manages to solve the request from R4, all links till source R1, namely 'g' and 'a' are removed as well as the links of the other pending request from R5 sourced from R2, namely 'h' and 'c' are also removed.

4. Simulation Results

In simulations depicted robots are modeled homogenously mainly for simplicity with maximum acceleration and deceleration of $2m/s^2$, maximum velocity of $5m/s$ and reaction time of $12s$.

Figure 5 shows an instance during the navigation of a three robot system when robots 0 and 1 as well as 1 and 2 need to enter the cooperative phase for conflict resolution.

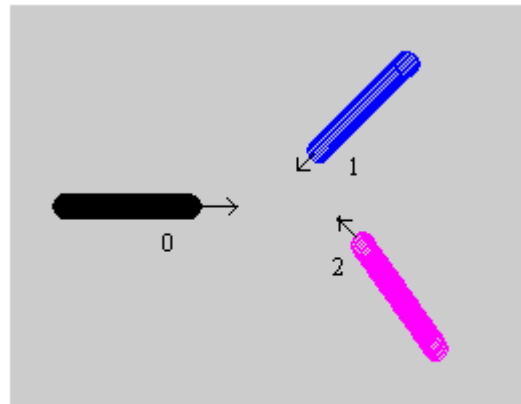


Figure 5 An instance of a system of 3 robots where robots 0 and 1 as well as 1 and 2 enter into a cooperative phase for conflict resolution

Figure 6 shows the space-time evolution of trajectories of the system of figure 5. The x and y axes represent the circular region of the x-y plane occupied by the robots during every sample of the environment shown along the z axis.

Figure 7a shows an instant during the navigation of a system of five robots where robots 1 and 3 are unable to resolve their conflicts between them individually as well as cooperatively as cooperative solutions lead to indirect conflict with robot 4. Hence 1 and 3 propagate a request to resolve their conflict to 4 thereby embarking on the conflict propagation phase as the last attempt to resolve their

conflicts. Robot 4 accepts requests from 1 and 3 and is able to solve the request of 1 by modifying its current velocity such that 1 manages to avoid its indirect conflict with 4. This scenario is depicted in figure 7b where 4 moves faster in such a way 1 and 3 are able to avoid their mutual direct conflict (MDC).

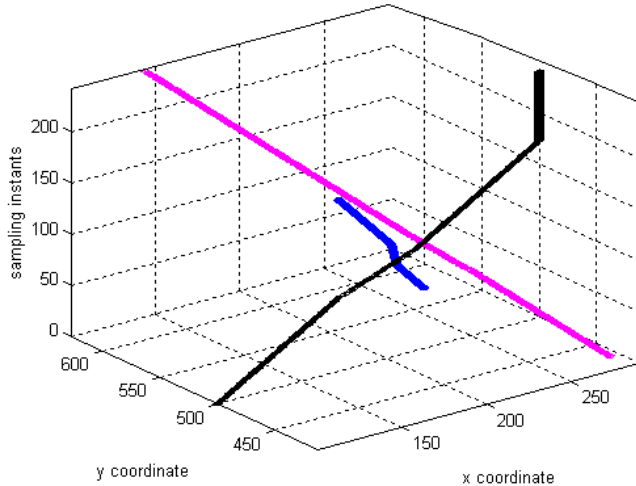


Figure 6: Spacetime evolution of the trajectories of the three robot system

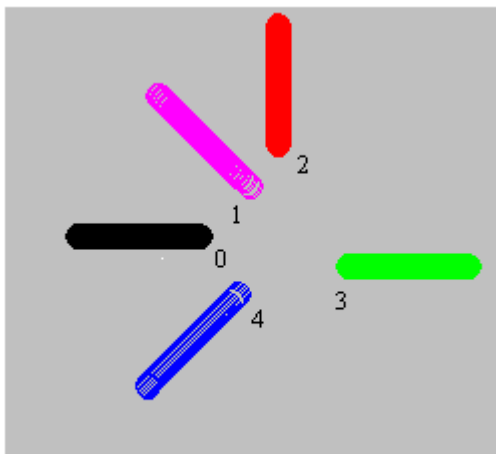


Figure 7a: Snapshot of a system of five robots

Assumptions: Certain assumptions have been made in the formulation that have been carried out in the simulation and need to be spelt out.

- (i) Every robot is capable of assessing its own state (position, velocity) and reports to other robots within a particular range fixed at 100 pixels in simulation
- (ii) All information that needs to be passed back and forth between robots such as in the cooperative and conflict propagation phase is fast enough to occur within sampling time

$$t_{\text{samp}}$$

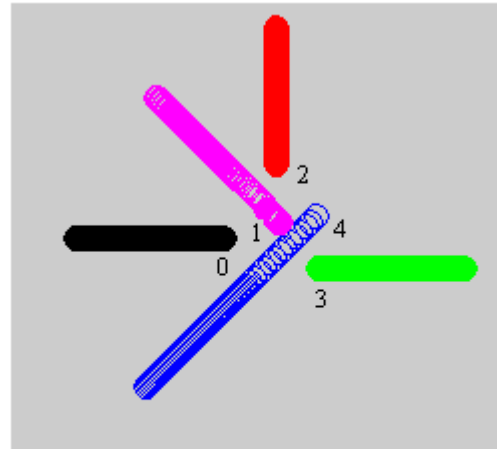


Figure 7b: Robot 1 and 3 propagate a request to resolve their conflict to robot 4, which accepts the request and moves faster such that 1 and 3 are able to avoid their mutual direct conflict

5 Conclusions

A novel distributed three-tiered approach for coordinated cooperative collision avoidance for a multi robot system from a reactive navigation standpoint has been presented and the simulation results confirm the efficacy of the proposed model. Robots resolve conflicts at three levels namely, individual, cooperative and conflict propagation phases. The approach is particularly suitable for a large number of robots moving about in shop floors, factories, airports and the like where a-priori knowledge of the plans of all other robots in the system is not made available for every robot in lieu of computational complexity. Future areas of work include incorporating a cooperative orientation control scheme and the investigation of various social cues such as benevolence, deception in conflict resolution in a multi-robot system.

References:

- [1]. V. Genevose, R. Magni and L. Odetti, "Self-organizing Behavior and Swarm Intelligence in a Pack of Mobile Miniature Robots in Search of Pollutants", *Proc. 1992, IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems*, Raleigh, NC, 1992, pp. 1575-1582
- [2] L E. Parker, ALLIANCE: An Architecture for Fault Tolerant Multi-Robot Cooperation, *IEEE Transactions on Robotics and Automation*, 14 (2), 1998.
- [3] H. Choset, "Coverage for robotics - a survey of recent results". *Annals of Mathematics and Artificial Intelligence*, 31:113-126, 2001.
- [4] R. Alami, S. Fleury, M. Herbb, F. Ingrand and F. Robert, "Multi Robot Cooperation in the Martha Project", *IEEE Robotics and Automation Magazine*, 5(1), 1998

- [5] S.J. Buckley. Fast Motion Planning for Multiple Moving Robots. *Proc. IEEE Int. Conf. on Robotics and Autom.*, p. 1419-1424, 1989.
- [6] K. Azarm & G. Schmidt. Conflict-Free Motion of Multiple Mobile Robots Based on Decentralized Motion Planning and Negotiation, *Proc. IEEE Int. Conf. on Robotics and Automation*, p. 3526-3533, 1997.
- [7] C M Clark, S M Rock and J C Latombe, "Motion Planning for Mobile Robots using Dynamic Networks", to appear in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2003
- [8] F. Gravit and R. Alami "An extension of the plan-merging paradigm for multi-robot coordination" *Proc. IEEE Int. Conf. on Robotics and Automation*, pp.2929-2934, May 2001
- [9] R Alami, F Ingrand and S Qutub, "A Scheme for Coordinating Multi-robot Planning Activities and Plans Execution", *Proc 13th European Conference on AI*, 1998
- [10] A Fujimori, M Teramoto, P N Nikiforuk and M M Gupta, "Cooperative Collision Avoidance between Multiple Mobile Robots", *Journal of Robotic Systems* 17(7), 347-363, 2000
- [11] K Madhava Krishna and Prem K Kalra, "Detection Tracking and Avoidance for Multiple Dynamic Objects", *Journal of Intelligent and Robotic Systems*, 33(3), 371-408 Apr 2002, Kluwer Academic
- [12] K Madhava Krishna and H Hexmoor, "Towards Quantification of the need to Cooperate between Robots", submitted to *PerMIS'03, Performance Metrics in Intelligent Systems*