Multi Robotic Conflict Resolution by Cooperative Velocity and Direction Control

Satish Pedduri+
K Madhava Krishna+
+ *International Institute of Information Technology, Hyderabad, India*

1. Introduction

Collision avoidance is one of the essential pillars of a wheeled robotic system. A wheeled mobile robot (called mobile robot for conciseness henceforth) requires for effective functioning an integrated system of modules for (i) map building, (ii) localization, (iii) exploration, (iv) planning and (v) collision avoidance. Often (i) and (ii) are entailed to be done simultaneously by robots resulting in a vast array of literature under the category SLAM, simultaneous localization and mapping. In this chapter we focus on the aspect of collision avoidance specifically between multiple robots, the remaining themes being too vast to even get a brief mention here.

We present a cooperative conflict resolution strategy between multiple robots through a purely velocity control mechanism (where robots do not change their directions) or by a direction control method. The conflict here is in the sense of multiple robots competing for the same space over an overlapping time window. Conflicts occur as robots navigate from one location to another while performing a certain task. Both the control mechanisms attack the conflict resolution problem at three levels, namely (i) individual, (ii) mutual and (iii) tertiary levels. At the individual level a single robot strives to avoid its current conflict without anticipating the conflicting robot to cooperate. At the mutual level a pair of robots experiencing a conflict mutually cooperates to resolve it. We also denote this as mutually cooperative phase or simply cooperative phase succinctly. At tertiary level a set of robots cooperate to avoid one or more conflicts amidst them. At the tertiary level a robot may not be experiencing a conflict but is still called upon to resolve a conflict experienced by other robots by modifying its velocity and (or) direction. This is also called as propagation phase in the chapter since conflicts are propagated to robots not involved in those. Conflicts are resolved by searching the velocity space in case of velocity control or orientation space in case of direction control and choosing those velocities or orientations that resolve those conflicts. At the individual level the search is restricted to the individual robot's velocity or direction space; at the mutual level the search happens in the velocity or direction space of the robot pair experiencing the conflict and at tertiary levels the search occurs in the joint space of multiple robots. The term cooperative is not a misnomer for it helps in achieving the following capabilities:

Source: Mobile Robots: Perception & Navigation, Book edited by: Sascha Kolski, ISBN 3-86611-283-1, pp. 704, February 2007, Plv/ARS, Germany

1 Avoid collision conflicts in a manner that conflicting agents do not come too near while avoiding one and another whenever possible. Thus agents take action in a fashion that benefits one another apart from avoiding collisions.
2 Provides a means of avoiding conflicts in situations where a single agent is unable to resolve the conflict individually.
3 Serves as a pointer to areas in the possible space of solutions where a search for solution is likely to be most fruitful.

The resolution scheme proposed here is particularly suitable where it is not feasible to have a-priori the plans and locations of all other robots, robots can broadcast information between one another only within a specified communication distance and a robot is restricted in its ability to react to collision conflicts that occur outside of a specified time interval called the reaction time interval. Simulation results involving several mobile robots are presented to indicate the efficacy of the proposed strategy.

The rest of the chapter is organized as follows. Section 2 places the work in the context of related works found in the literature and presents a brief literature review. Section 3 formulates the problem and the premises based on which the problem is formulated. Section 4 mathematically characterizes the three phases or tiers of resolution briefly mentioned above. Section 5 validates the efficacy of the algorithm through simulation results. Section 6 discusses the limitations of the current approach and its future scope and ramifications and section 7 winds up with summary remarks.

2 Literature Review

Robotic navigation for single robot systems has been traditionally classified into planning and reactive based approaches. A scholarly exposition of various planning methodologies can be found in (Latombe 1991). A similar exposition for dynamic environments is presented by Fujimora (Fujimora 1991). Multi-robot systems have become an active area of research since they facilitate improved efficiency, faster responses due to spread of computational burden, augmented capabilities and discovery of emergent behaviors that arise from interaction between individual behaviors. Multiple mobile robot systems find applications in many areas such as material handling operations in difficult or hazardous terrains (Genevose at. al, 1992)[3], fault-tolerant systems (Parker 1998), covering and exploration of unmanned terrains (Choset 2001), and in cargo transportation (Alami et. al, 1998). Collaborative collision avoidance (CCA) between robots arises in many such multi-robot applications where robots need to crisscross each other's path in rapid succession or come together to a common location in large numbers. Whether it is a case of 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, CCA becomes unavoidable.

Multi-robotic navigation algorithms are traditionally classified as centralized or decentralized approaches. In the centralized planners [Barraquand and Latombe 1990, Svetska and Overmars 1995] the configuration spaces of the individual robots are combined into one composite configuration space which is then searched for a path for the whole composite system. In case of centralized approach that computes all possible conflicts over entire trajectories the number of collision checks to be performed and the planning time tends to increase exponentially as the number of robots in the system increases. Complete recalculation of paths is required even if one of the robot's plans is altered or environment changes. However centralized planners can guarantee completeness and optimality of the method at-least theoretically.

Decentralized approaches, on the other hand, are less computationally intensive as the computational burden is distributed across the agents and, in principle, the computational complexity of the system can be made independent of the number of agents in it at-least to the point of computing the first individual plans. It is more tolerant to changes in the environment or alterations in objectives of the agents. Conflicts are identified when the plans or commands are exchanged and some kind of coordination mechanism is resorted to avoid the conflicts. However, they are intrinsically incapable of satisfying optimality and completeness criterion. Prominent among the decentralized approaches are the decoupled planners [Bennewitz et. al, 2002], [Gravot and Alami 2001], [Leroy et. al 1999]. The *decoupled* planners first compute separate paths for the individual robots and then resolve possible conflicts of the generated paths by a hill climbing search [Bennewitz et. al, 2004] or by plan merging [Gravot and Alami 2001] or through dividing the overall coordination into smaller sub problems [Leroy et. al 1999].

The method presented here is different in that complete plans of the robots are not exchanged. The locations of the robots for a certain T time samples in future are exchanged for robots moving along arcs and for those moving with linear velocities along straight lines it suffices to broadcast its current state. The collisions are avoided by searching in the velocity or the orientation space (the set of reachable orientations) of the robot. In that aspect it resembles the extension of the Dynamic Window approach [Fox et. al, 1997] to a multi robotic setting however with a difference. The difference being that in the dynamic window the acceleration command is applied only for the next time interval whereas in the present method the restriction is only in the direction of change in acceleration over a time interval $t < T$ for all the robots.

The present work is also different from others as the resolution of collision conflicts is attempted at three levels, namely the individual, cooperative, and propagation levels. Functionally cooperation is a methodology for pinning down velocities or orientations in the joint solution space of velocities or orientations of the robots involved in conflict when there exists no further solution in the individual solution space of those robots. When joint actions in the cooperative phase are not sufficient for conflict resolution assistance of other robots that 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. The concept of propagating conflict resolution requests to robots not directly involved in a conflict is not found mentioned in robotic literature. Such kind of transmission of requests to robots though not invoked frequently is however helpful in resolving a class of conflicts that otherwise would not be possible as our simulation results reveal.

The method presented here is 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 present is a scheme for cooperative collision avoidance by Fujimora's group (Fujimora et. al, 2000) and a distributed fuzzy logic approach as reported in (Srivastava et. al, 1998). Their work is based on devising collision avoidance 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 (Krishna and Kalra, 2002) that such PBA techniques fail when individual actions that get weighted and averaged in the PBA are conflicting in nature. The work of Lumelsky (Lumelsky and Harinarayanan 1998) is of relation here in that it does not entail broadcast of plans to all other robots. It describes an extension of one of the *Bug* algorithms to a multi robotic setting. There is not much mention of cooperation or collaborative efforts between the robots except in the limited sense of "reasonable behavior" that enables shrinking the size of collision front of a robot that is sensed by another one.

3 Objective, Assumptions and Formulations:

Given a set of robots $R = \{R_1, R_2, \ldots, R_n\}$, each assigned a start and goal configuration the objective is to navigate the robot such that they reach the goal configuration avoiding all collisions.

While collisions could be with stationary and moving objects in this chapter we focus specifically how the robots could avoid collisions that occur amongst them in a cooperative fashion. For this purpose the following premises have been made:

a. Each robot *Ri* is assigned a start and goal location and it has access to its current state and its current and aspiring velocities. The current state of *Ri* is represented as $\psi_i = \{vc_i, vn_i, \theta c_i, \theta n_i\}$ where vc, vn represent its current and aspiring velocities and $\theta c, \theta n$ its current and aspiring directions. The aspiring direction to be reached at a given time *t* is the angle made by the line joining the current position to the position reached at *t* with the current heading. This is shown in figure 1 where a robot currently at P reaches a point N moving along an arc, the aspiring orientation is the angle made by the dashed line connecting P to N with the current heading direction.
b. All robots are circular and described by their radius
c. Robots are capable of broadcasting their current states to each other. They do so only to those robots that are within a particular range of communication.
d. Robots accelerate and decelerate at constant rates that are same for all. Hence a robot *Ri* can predict, when another robot *Rj* would attain its aspiring velocity vn from its current velocity vc if it does not change its direction.

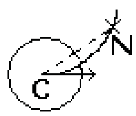

Fig. 1. A robot currently at location C moves along a clothoidal arc to reach position N. The aspiring orientation is computed as mentioned in premise a in text. The heading at C is indicated by the arrow

3.1 Robot Model

We consider a differential drive (DD) mobile robot in consonance with the robots available in our lab. Figure 2a shows an abstraction of a DD robot consisting of two wheels mounted on a common axis driven by two separate motors. Consider the wheels rotating about the current center C, at the rate ω as shown in figure 2. The coordinates of the axis center is (x, y) and the robot's bearing is at θ with respect to the coordinate frame. The distance between the two wheels is L and the radius of curvature of the robot's motion is R (distance from C to robot's center). Given that the left and right wheel velocities are v_l, v_r the following describe the kinematics of the robot:

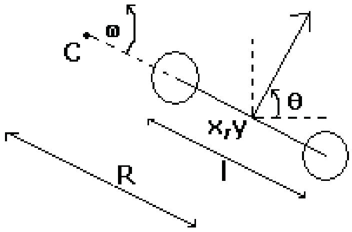

Fig. 2a. A differential drive robot with left and right wheels driven by two motors that are independently controlled.

$$\omega = \frac{v_r - v_l}{l} \quad \cdots\cdots \quad (3.1.1);\ v = \frac{v_r + v_l}{2} \quad \cdots\cdots \quad (3.1.2);\ R = \frac{l(v_r + v_l)}{2(v_r - v_l)} \quad \cdots\cdots \quad (3.1.3)$$

Here v, ω represent the instantaneous linear and angular velocity of the robot. Given the current configuration of the robot's center as x_0, y_0, θ_0 at t_0 the coordinates reached by the robot at time *t* under constant linear and angular acceleration (a, α) is given by

$$x = x_0 + \int_{t_0}^{t} (v(t_0) + at)\cos\left(\theta_0 + \omega t + \alpha t^2\right) dt \quad \cdots\cdots \quad (3.1.4)$$ and

$$y = y_0 + \int_{t_0}^{t} (v(t_0) + at)\sin\left(\theta_0 + \omega t + \alpha t^2\right) dt \quad \cdots\cdots \quad (3.1.5).$$

Integrals 3.1.4 and 3.1.5 require numerical techniques to compute. Hence in a manner similar to (Fox et.al, 1997) we assume finite sufficiently small intervals for which the velocity is assumed to be a constant and the coordinates of the robot are then computed as

$$x(t_n) = x_0 + \sum_{i=0}^{n-1} \int_{t_i}^{t_{i+1}} v_i \cos(\theta(t_i) + \omega(t - t_i)) dt \quad \ldots\ldots \quad (3.1.6)$$

$$y(t_n) = y_0 + \sum_{i=0}^{n-1} \int_{t_i}^{t_{i+1}} v_i \sin(\theta(t_i) + \omega(t - t_i)) dt \quad \ldots\ldots \quad (3.1.7)$$

In the collision avoidance maneuver it is often required to check if the robot can reach to a location that lies on one of the half-planes formed by a line and along a orientation that is parallel to that line. In figure 2b a robot with current configuration x_s, y_s, θ_s with velocity v_s wants to reach a position on the left half plane (LHP) of line *l* along a direction parallel to *l*. For this purpose we initially compute where the robot would reach when it attains either the maximum angular velocity shown in angular velocity profiles of figures 2c and 2d under maximum angular acceleration. The positions reached at such an instant are computed through (3.1.6) and (3.1.7). Let the maximum angular velocity in a given velocity profile as determined by figures 2d and 2e be ω_{aM} and the location reached by the robot corresponding to ω_{aM} be x_{aM}, y_{aM}. ω_{aM} is not necessarily the maximum possible angular velocity ω_M and is determined by the time for which the angular acceleration is applied.

Consider a circle tangent to the heading at x_{aM}, y_{aM} with radius $\frac{v_s}{\omega_{aM}}$, this circle is shown dashed in figure 2c. Consider also the initial circle which is drawn with the same radius but which is tangent to θ_s at x_s, y_s, which is shown solid in 2c. Evidently the initial circle assumes that the robot can reach ω_{aM} instantaneously. Let the displacements in the centers of the two circles be $d_{s,aM}$. Then if the initial circle can be tangent to a line parallel to *l* that is at-least $kd_{s,aM}$ from *l* into its LHP then the robot that moves with an angular velocity profile shown in figures 2d or 2e can reach a point that lies in the LHP of *l* along a direction parallel to *l*. We found *k*=*2* to be a safe value. It is to be noted checking on the initial circle is faster since it avoids computing the entire profile of 2d or 2e before concluding if an avoidance maneuver is possible or not.

3.2 The Collision Conflict

With robots not being point objects a collision between two is modeled as an event happening over a period of time spread over an area. The collision conflict (CC) is formalized here for the simple case of two robots moving at constant velocities. The formalism is different if velocity alone is controlled or direction control is also involved. Figure 3 shows the CC formalism when velocity control alone is involved.

Shown in figure 3, two robots R1 and R2 of radii r1 and r2 and whose states are $\psi_1 = (vc_1, vn_1)$ and $\psi_2 = (vc_2, vn_2)$ respectively, where vc_1, vc_2 are the current velocities while vn_1, vn_2 are the aspiring velocities for R1 and R2 respectively. The orientations are omitted while representing the state since they are not of concern here. 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 *R1* is shrunk to a point and the other *R2* is grown by the radius of the shrunken robot. The points of interest 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 $(r1 + r2)\cos ec(|\theta_1 - \theta_2|)$ on either side of C. The time taken by *R2* to reach C21 and C22 given its current state (vc_2, vn_2) 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 r2. Locations C11 and C12 and the time taken by R1 to reach them t_{11} and t_{12} are thus computed. A collision conflict or CC is said to be averted between R1 and R2 if and only if $[t_{11}, t_{12}] \cap [t_{21}, t_{22}] \in \Phi$. The locations C11, C12, C21 and C22 are marked in figure1.

A direct collision conflict (*DC*) 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.

For direction control the CC is formalized as follows. Consider two robots R1 and R2 approaching each other head on as in figure 4a and at an angle in figure 4b. The points at which the robots are first tangent to one another (touch each other exactly at one point) correspond to locations C11 and C21 of R1 and R2's center. The points at which they touch firstly and lastly are marked as P in 4a and P1, P2 in 4b. Let t_{c1}, t_{c2} denote the times at which they were first and lastly tangent to each other. We expand the trajectory of R2 from all points between and including C21 and C22 by a circle of radius r1 while R1 is shrunk to a point. The resulting envelope due to this expansion of the path from C21 to C22 is marked E. All points outside of E are at a distance r2+r1 from R2's center when it belongs to anywhere on the segment connecting C21 to C22. The envelope E consists of two line segment portions $\overline{E1E2}$, $\overline{E3E4}$ and two arc segment portions $E1A1E4$, $E2A2E3$ shown in figures 4a and 4b. We say a CC is averted if R1 manages to reach a location that is outside of E with a heading θ_a for the time R2 occupies the region from C21 to C22 and upon continuing to maintain its heading guarantees resolution for all future time.

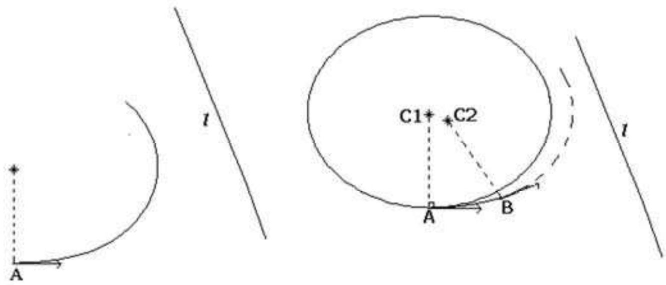

Fig. 2b. A robot at A heading along the direction denoted by the arrow wants to reach a position that lies on the LHP of line *l* along a orientation parallel to *l*. Its angular velocity should reach zero when it reaches a orientation parallel to *l*.

Fig. 2c. In sequel to figure 2b, the robot at A takes off along a clothoidal arc approximated by equations 3.1.6 and 3.1.7 and reaches B with maximum angular velocity. It then moves along a circle centered at C2 shown dotted and then decelerates its angular velocity to zero when it becomes parallel to *l*. The initial circle is drawn centered at C1 tangent to the robot's heading at A. The distance between C1 and C2 decides the tangent line parallel to *l* to which the robot aspires to reach.

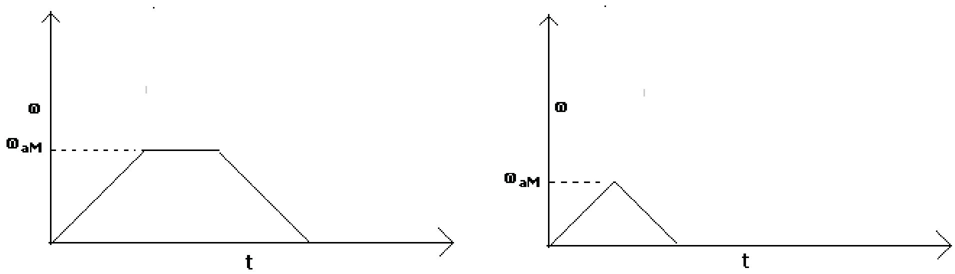

Fig. 2d and 2e. Two possible angular velocity profiles under constant acceleration. Figure 3d corresponds to a path that is a circle sandwiched between two clothoids, while Figure 3e corresponds to the path of two clothoids.

For example in 4a R1 reaches the upper half plane of the segment $\overline{E1E2}$ or the lower half plane of $\overline{E3E4}$ before R2 reaches P then it guarantees resolution for all future times provided R2 does not change its state. Similarly in figure 4b by reaching a point on the lower half plane of $\overline{E3E4}$ with a heading parallel to $\overline{E3E4}$ collision resolution is guaranteed. It is obvious R2 would not want to maintain its heading forever, for it will try to reach its actual destination once the conflict is resolved.

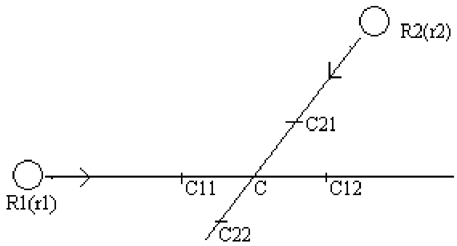

Fig. 3. Two robots *R1* and *R2* with radii r1 and r2 along with their current states are shown. When *R1* is shrunk to a point and *R2* grown by radius of R1, C21 and C22 are centers of *R2* where the path traced by *R1* becomes tangential to *R2*.

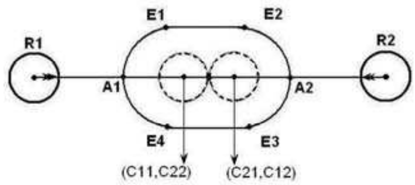

Fig. 4a. Situation where two Robots approaching head on.

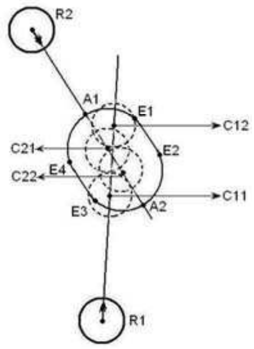

Fig. 4b. Situation where two Robots approaching at an angle.

4 Phases of Resolution

Let S_T be the set of all possible solutions that resolve conflicts among the robots involved. Depending on the kind of control strategy used each member $s_i \in S_T$ can be represented as follows:

i. An ordered tuple of velocities in case of pure velocity control i.e. $s_i = \{v_{1i}, v_{2i}, \ldots, v_{Ni}\}$, for each of the N robots involved in the conflict. Obviously the set s_i is infinite, the subscript *i* in s_i is used only for notational convenience.

ii. An ordered tuple of directions in case of pure direction control i.e. $s_i = \{\theta_{1i}, \theta_{2i}, \ldots, \theta_{Ni}\}$.

iii. An ordered tuple of velocity direction pairs in case of velocity and direction control, $s_i = \{\{v_{1i}, \theta_{1i}\}, \{v_{2i}, \theta_{2i}\}, \ldots, \{v_{Ni}, \theta_{Ni}\}\}$ in case of both velocity and direction control.

Conflicts are avoided by reaching each component of s_i, i.e. the velocities or directions or both within a stipulated time tuplet $\{t_{1i}, t_{2i}, \ldots, t_{Ni},\}$. For purely velocity control the velocities to be attained involved not more than one change in direction of acceleration, i.e., they are attained by an increase or decrease from current acceleration levels but not a combination of both. For purely direction control the final orientation aspired for involves not more than one change in turning direction. However the final direction attained could be through a sequence of angular velocity profiles such as in figures in 2d & 2e that involve only one change in turning direction.

The *cooperative space* is represented by the set $S_C \subseteq S_T$, i.e., the cooperative space is a subset of the total solution space and where every robot involved in the conflict is required to modify its current aspiring velocity or direction to avoid the conflict. In other words robots modify the states in such a manner that each of the robot involved has a part to play in resolving the conflict. Or if any of the robots had not modified its velocity it would have resulted in one or more collisions among the set of robots involved in the conflict.

The *cooperative phase* in navigation is defined by the condition $S_C = S_T$, where each robot has no other choice but to cooperate in order to resolve conflicts. In individual resolution robots choose velocities in the space of $S_I = S_T - S_C$, where the entailment for every robot to cooperate does not exist. When $S_I = \Phi$, the null set, we say the navigation has entered the cooperative phase.

Figures 5a-5d characterize the changes in solution space due to velocity control alone for evolving trajectories of two orthogonal robots while those of 6a-6e do the same for orientation control of robots that approach each other head on. Figure 5a shows evolution of trajectories of two robots, marked R1 and R2, moving orthogonal to one another. The arrows show the location of the two robots at time $t = 0$ sample. The robots move with identical speed of $v_{R1} = v_{R2} = 2.5$ units. The states of the two robots are represented as $\psi_1 = (v_{R1}, v_{R1}, 0, 0)$ and $\psi_2 = (v_{R2}, v_{R2}, -90, -90)$. The equality in the current and aspiring velocities merely indicates that the robot moves with uniform velocity and is not a loss of generality from the case when the aspiring velocity differs from the current. The subsequent discussion holds equally for the case when the current and aspiring velocities differ. Corresponding to this location of the robots at the beginning of their trajectories, figure 5b depicts the total space of velocities bounded within the outer rectangle (shown thick) whose length and breadth are 5 units respectively. In other words each robot can have velocities in the interval $[0,5]$ units. The abscissa represents the range for one of the robots (R1) and the ordinate the range for the other (R2). The center of the figure marked as O indicates the location corresponding to their respective velocities of 2.5units each. The strips of shaded region represent those velocities not reachable from O due to the limits of acceleration and deceleration. The inner rectangle, marked ABCD, represents the region of velocities where a possible solution can be found ***if and only if*** both robots alter their velocities. For $v_{R1} = 2.5$ corresponding to R1's velocity on the abscissa, R2 must possess a velocity, which lies either above or below the segments AB and CD of the rectangle when projected onto the ordinate. Similarly for $v_{R2} = 2.5$ on the ordinate, robot R1 must possess a velocity either to the right or left of the segments BC and AD when projected onto the abscissa to avert collision. We denote the velocities that make R1 reach the velocities at D and C from O as v_{11} and v_{12} respectively, while the velocities that make R2 reach A and D from O by v_{21} and v_{22} respectively. With reference to figure 3 v_{11} and v_{12} correspond to velocities that enable R1 to reach C11 and C12 in the time R2 reaches C22 and C21 respectively without R2 changing its current aspiring velocity from v_{R2}.

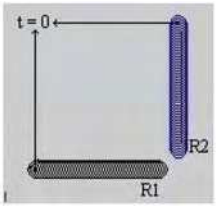

Fig. 5a. Two robots approach each other along orthogonal directions.

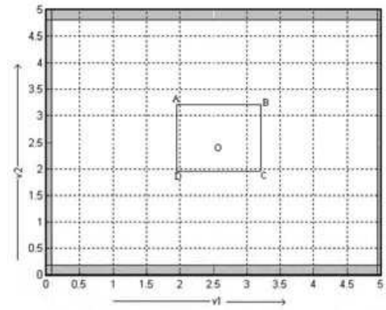

Fig. 5b. The possible range of velocities for robots R1 and R2 shown along the x and y axis. The inner rectangular area being cooperative region.

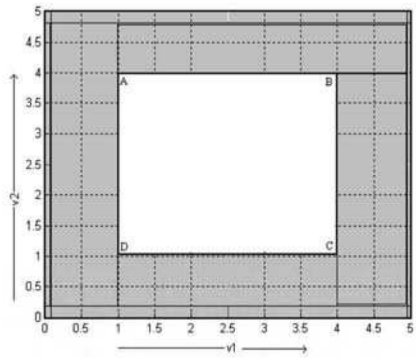

Fig. 5c. At t=25 the conflict area occupies the entire possible space of velocities.

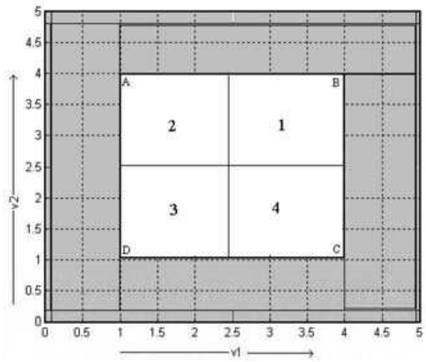

Fig. 5d. Search is limited to quadrants 2 and 4 where robot actions are complementary.

Figure 6a shows the snapshot at time $t = 0$ or $t_{c1} = 19$ (the time left for the robots to become tangent to one another for the first time) when robots approach each other head on. Figure 6b shows the collision region marked on θ axis for R1. All θ values in the interval [b,d] on the right and [a,c] on left are reachable and collision free. Values in the interval [d,M] and [m,c] are not accessible or unattainable due to the limits on angular acceleration of the robot, while those in [a,b] conflict with the impinging robot. Figure 6c shows the conflicting and inaccessible orientations overlap in intervals [a,c] and [d,b] for time $t_{c1} = 14$. Figure 6c shows the need for cooperation since the entire θ axis of R1 is either conflicting or inaccessible or both. The values of θ to the left of O (corresponding to current heading of R1) on the θ axis of R1 are those obtained by turning R1 right in figure 6a & while those on the right of O on the θ axis are obtained by turning R1 to its left in figure 6a. While depicting the solution space in terms of θ for a robot the current heading is always 0 degrees for convenience.

Fig. 6a. Robots R1 and R2 approaching Head on.

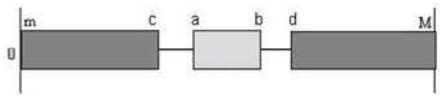

Fig. 6b. Collision and accessible regions on θ axis for robot R1 where [a, b] being the collision range.

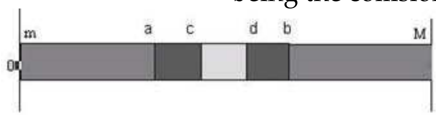

Fig. 6c. Collision and accessible regions on θ axis. Dark area showing the overlapped collision and inaccessible regions.

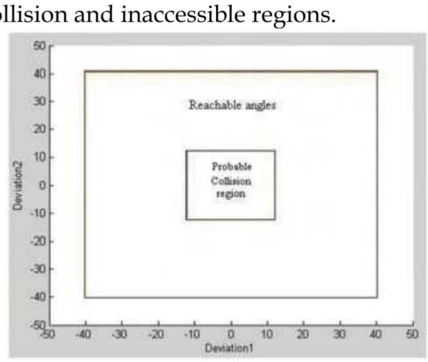

Fig. 6d. Joint orientation space for robots R1 and R2 in terms of θ_1 and θ_2. Outer rectangle representing accessible combination.

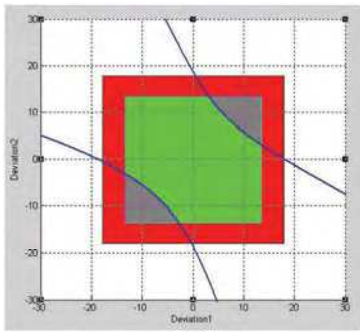

Fig. 6e. Joint orientation space for robots R1 and R2, where accessible region is inside the collision region where gray region representing cooperation zone.

Figures 6d and 6e depict the joint orientation solution space for robots R1 and R2 in terms of θ_1 (abscissa) and θ_2 (ordinate). Figure 6d corresponds to the situation for time $t = 0$ or $t_{c1} = 17$; the shaded parts of the rectangle comprises of regions inaccessible to both R1 and R2. R2 must reach a orientation on the ordinate that is either above or below segments AB and CD while R1 should reach a orientation that is either to the right of BC or left of AD. These orientations are denoted as θ_{11}, θ_{12} for R1 and θ_{21}, θ_{22} for R2 in a manner similar to velocity control discussed before. With reference to figure 4a θ_{11}, θ_{12} correspond to directions that enable R1 to reach the upper half plane of the segment $\overline{E1E2}$ or the lower half plane of $\overline{E3E4}$ before R2 reaches C21 without R2 changing its current aspiring orientation that is 0 degrees with respect to itself and 180 degrees with respect to a global reference frame, F shown in 6a.

4.1 Individual Phase for Velocity Control

A pair of robots *R1* and *R2*, which have a DC between them are said to be in individual phase of navigation if the conflict is resolved by either of the following two means:

(i) *R1* controls its velocity to v_{12} such that it is able to get past C12 before *R2* reaches C21 with its aspiring velocity as v_{R2} or *R1* controls its velocity to v_{11} such that it does not reach C11 before *R2* reaches C22 without changing its aspiring velocity from v_{R2}.

(ii) *R2* controls its velocity to v_{22} such that it is able to get past C22 before *R1* reaches C11 with its current aspiring velocity as v_{R1} or *R2* controls its velocity to v_{21} such that it does not reach C21 before *R1* reaches C12 without changing its aspiring velocity from v_{R1}.

In both cases it would suffice that only one of the two robots controls or modifies its aspiring 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 requiring the other to take action. Thus the range of velocities that permit individual resolution of conflict by *R1* is given by: $v \in [0, v_{11}] \cup [v_{12}, v_{1M}]$, where v_{1M} represents the maximum permissible velocity for *R1,* which is 5 units in figure 2b. They are given by: $v_{11} = vc_1 + a_{-m}t_{22} \pm \sqrt{(vc_1 + a_{-m}t_{22})^2 + (vc_1^2 + 2a_{-m}s)}$ Here s denotes the distance from *R1's* current location to C11, a_{-m} is the maximum possible deceleration and t_{22} is the time taken by R2 to reach C22 given its current state ψ_2. In the same vein the velocity that causes *R1* to be ahead of C12 when *R2* reaches C21 under maximum acceleration, a_m, is given by:

$v_{12} = vc_1 + a_m t_{21} \pm \sqrt{(vc_1 + a_m t_{21})^2 + (vc_1^2 + 2a_m s')}$, where, s' the distance from R1's current location to C12 can also be written as $s' = s + (r1 + r2)\cos ec(|\theta_1 - \theta_2|)$ and t_{21} is the time taken by R2 to reach C21 given its current state ψ_2. In a similar fashion velocities v_{21} and v_{22} are computed. Thus some of the possible sets of solutions from the set S_T are enumerated as: $s_1 = \{v_{11}, v_{R2}\}$, $s_2 = \{v_{12}, v_{R2}\}$, $s_3 = \{v_{R1}, v_{21}\}$, $s_4 = \{v_{R1}, v_{22}\}$, $s_5 = \{v_{11}, v_{22}\}$, $s_6 = \{v_{21}, v_{12}\}$.

From the above list the first four solutions involve change in velocities of only one of the robots while the last two solutions involve change in velocities of both the robots. The last two solutions are examples of collaboration even in the individual phase as robots involve in a combined effort to avoid conflict even though they are not entailed to do so. The collaboration in the individual phase achieves the first capability mentioned in section 1 of avoiding conflicts in a manner that conflicting agents do not come too near while avoiding one and another. Amongst the last two solutions (s_5, s_6) that one is selected which involves minimal change from the current state of the respective robots. The last two solutions indicate that collaboration involves complementary decision making since one of the robots accelerates from its current velocity the other decelerates.

Henceforth for any robot the lower velocity is denoted as v_1 and the higher velocity by v_2 with the robot index dropped for notational simplicity. In other words the lower and upper velocities are denoted as v_1 and v_2 instead of v_{21} and v_{22} for R2 or instead of v_{I1}, v_{I2} for RI.

It is to be noted that the phrase that a robot change or modify its velocity is more precisely stated as the robot control or modify its *aspiring* velocity.

4.1.2 Individual Phase for Direction Control

Unlike velocity control a unique way of characterizing θ_{11}, θ_{12} is difficult depending on the angular separation between the robots and their directions of approach. However certain commonalities can be observed, namely (i) the robot to be avoided can be encapsulated within a planar envelope E (section 3.2), (ii) the robot that avoids has essentially two turning options either to turn left or right, (iii) the robot can reach a point in the plane that has no overlaps with E by reaching a heading, can in principle continue with the heading and avoid conflict forever with the same robot. Based on the above observations we formulate a conservative resolution criteria based on the angular separation between the two robots.

In purely velocity control a closed form solution to the values v_{11} & v_{12} was possible to ascertain, whereas in direction control a closed form expression for θ_{11}, θ_{12} is very difficult to obtain due to following reasons. Firstly in velocity control the robot had to reach a particular point for the limiting case. Whereas in direction control the robot is can reach any point on a line as long as its orientation is the same as that line in the limiting case. This leads to several velocity profile choices for the same target criteria. Secondly in the velocity control scheme it is possible to reach a particular linear velocity and maintain that as the aspiring velocity, however in direction control the eventual angular aspiring velocity needs to be zero for any avoidance maneuver. Hence it is easier to work in the space of directions than in space of angular velocities. For computing the solution space an exhaustive search mechanism is resorted by changing the time for which an acceleration command is applied for the same linear velocity. These are the solution spaces shown in the chapter under the assumption current linear velocity remains unchanged since those depicted are those for purely direction control. In case of the actual algorithm running real-time few sample points in the $\{v, \alpha\}$ space are computed before a conclusion regarding which phase of resolution is to be resorted to. The basis or the motivation for selecting the candidate points will be discussed elsewhere.

Figures 7a and 7b are similar to those of 4a and 4b. Figure 7a depicts the head on case while 7b portrays the case when angular separation between the robots lies in the interval [90,180). Both the cases have been discussed in detail in section 3.2 and early parts of this section when figures 6a – 6d were discussed. For the sake of completion we briefly mention them here. For 7a θ_{11}, θ_{12} are easily computed and correspond to directions that enable R1 to reach the lower half plane of the segment $\overline{E3E4}$ or the upper half plane of $\overline{E1E2}$ before R2 reaches P. For a given linear velocity of R1 θ_{11}, θ_{12} are symmetric on either sides of the current heading of R1 and this is expected as there are equal opportunities to avoid a conflict on both sides of the current heading. For figure 7b the conflict is best resolved if R1 reaches a point with a heading parallel to $\overline{E3E4}$ in the lower half plane of $\overline{E3E4}$ that does not contain R2. This can be achieved by either turning to its left or right. R1 can also aspire to reach a location in the upper half plane formed by $\overline{E1E2}$ that does not contain R2 before R2 reaches C21. This would once again involve R1 turning right. Hence θ_{12} corresponds to the value that is collision free by turning left whereas θ_{11} corresponds to the value that is collision free by turning right and reaching a point either on the upper half plane of $\overline{E1E2}$

before R2 reaches C21 or the lower half plane of the same $\overline{E1E2}$ without entering the envelope E during the time R2's center occupies the space from C21 to C22.

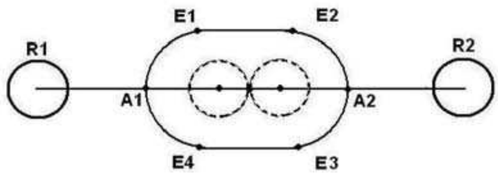

Fig. 7a. Robots R1 and R2 approaching head on.

Fig. 7b. Robots R1 and R2 approaching at an angle in range [90,180).

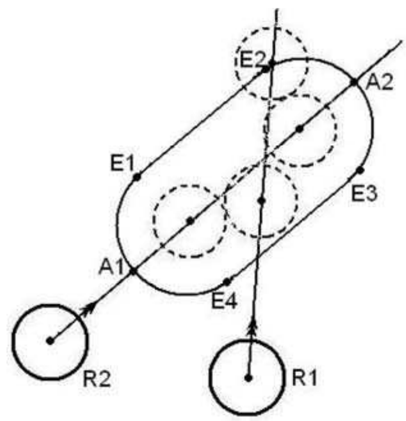

Fig. 7c. Robots R1 and R2 approaching at an angle less than 90 degrees.

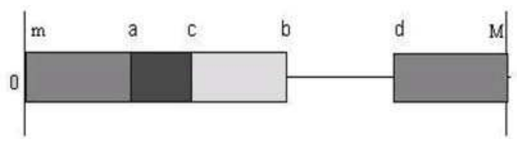

Fig. 7d. Collision and accessible regions on θ axis for robot R1 where [a,b] being the collision range.

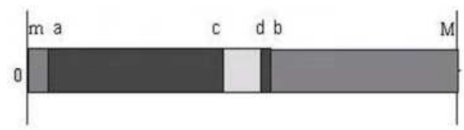

Fig. 7e. Collision and accessible regions on θ axis. Dark area showing the overlapped collision and inaccessible regions

Figure 7c depicts the case when the angular separation between the robots at the first instance of collision lies in (0,90]. Once again conflicts are resolved if the robot reaches a point in the half plane formed by $\overline{E3E4}$ along a orientation parallel to $\overline{E3E4}$ without entering the half-plane that contains R2. Figures 7d and 7e have exactly the same connotations as figures 6b and 6c except that they are the plots for robots approaching each other not head on but as in figure 7c. The θ axis is depicted as shown before. Figure 7d corresponds to $t_{c1} = 31$. Note the entire reachable space lies on the right of current heading of R1. This indicates only turns to the left avoid conflicts or a value for θ_{11} does not exist even very early in resolution. This is only expected since a cursory glance of figure 7c indicates most of the turns of R2 to its right could only collide with R1. Figure 7e indicates the onset of cooperation with $t_{c1} = 11$ where all the reach orientations all are in conflict with R2.

4.3 Mutual (Cooperative) Phase for Velocity Control

The area enclosed within the rectangle ABCD of figure 5b is termed as conflict area for the pair of velocities $\{v_{R1}, v_{R2}\}$ for time $t = 0$ and denoted as $CA(v_{R1}, v_{R2}, t = 0)$. Let $V_{r1} = [v_{l1}, v_{h1}]$ represent the range of velocities for which there is a collision for robot R1 when R2 possesses a velocity v_{R2}. Similarly let $V_{r2} = [v_{l2}, v_{h2}]$ represent the range of velocities for which there is a collision for robot R2 when robot R1 possesses a velocity v_{R1}. We define the conflict area for the velocity pair $\{v_{R1}, v_{R2}\}$ for a given time t as $CA(v_{R1}, v_{R2}, t) = \{v_{R1}, v_{R2} \mid v_{R1} \in V_{r1}, v_{R2} \in V_{r2}\}$. The velocities v_{l1}, v_{h1} for R1 and v_{l2}, v_{h2} for R2 are arbitrarily close to their respective upper and lower control velocities v_1, v_2 that are used for resolving conflicts. In other words $|v_{l1} - v_1| < \varepsilon \text{ for } R1, |v_{l2} - v_1| < \varepsilon \text{ for } R2$ and similarly $|v_{h1} - v_2| < \varepsilon, |v_{h2} - v_2| < \varepsilon$ where ε is any arbitrarily low value. With progress in time if control actions to avoid conflicts were not resorted to the conflict area expands to occupy the entire space of possible velocities. This is shown in figure 5c where the conflict area fills up the entire velocity space. Any combination of velocities outside the rectangle ABCD now falls inside the shaded border strips, which are not accessible from O due to the limits imposed by acceleration and deceleration. Hence individual resolution of conflicts by any one of the robots is ruled out since the upper and lower velocities v_1 and v_2 for both R1 and R2 now lie inside the shaded area.

Since the upper and lower velocities are situated well inside the shaded area the velocity pairs corresponding to the vertices ABCD of the conflict area are unknown. Hence a cooperative search ensues for finding the pair of velocities that would resolve the conflict. Cooperation between robots averts an exhaustive search and restricts it two quadrants 2 and 4 (figure 5d) of the conflict area where robot actions are complementary and yield best results for conflict resolution. Since a search is nonetheless time intensive the rules (i) and (ii) mentioned below where robots resort to maximum acceleration and deceleration in a complementary fashion offer the boundary value solutions. A failure of the solutions at the bounds implies a failure anywhere inside and a pointer to resort to conflict propagation as the last resort.

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

(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.

The difference between the above rules and those mentioned in section 4.1 is that in section 4.1 R1 finds a control velocity that avoids conflict with R2 under the premise that R2 would not alter its aspiring velocity. Similarly R2 finds a control velocity under the impression R1 is dumb. However in the cooperative phase R1 anticipates a modification in the aspiring velocity of R2 such as in rule 1 where R2 modifies its state (and hence its aspiring velocity) such that it reaches C12 under maximum deceleration. Under this anticipation of change in R2's control action R1 tries to attain the corresponding control velocity that would avoid conflict.

4.4 Mutual phase for direction control

As in velocity control figure 6e shows the situation when cooperation is inevitable since the entire accessible area (inner green rectangle) lies completely within the conflict area. The outer rectangle is the conflict area. The areas between the inner and outer rectangle are shown in red. The areas shown in gray within the rectangle are the solution pairs for which resolution is possible. Like in velocity control the solutions exist in opposing quadrants. The gray areas in first quadrant correspond to R1 and R2 turning left in figure 6a and 7a, while those in third quadrant correspond to R1 and R2 turning right. Once again if a solution does not exist at the top right and bottom left corners of the inner rectangle implies lack of solutions anywhere inside the inner rectangle.

Individual resolution through direction control fails because R1(R2) is unable to get out of E onto the half planes discussed earlier before R2(R1) reaches C21(C11). In such a situation the perpendicular distance from R1's (R2's) location to R2's (R1) trajectory is still less than r1+r2. Hence R2(R1) also changes its orientation to reach a location that would be r1+r2 away from each others trajectory by the instant it would have reached C21(C11) on the original trajectory had it not changed its direction. Hence a pair of robots can avoid conflicts mutually only if turning with maximum angular accelerations they can orient their trajectories by t_{c1} such that the perpendicular distance between a robot's position and the other robot's trajectory is at-least r1+r2. If the robots cannot reach such a location by t_{c1} under maximum angular acceleration applied till maximum angular velocities are attained then cooperative resolution would fail for all other values of α. Failure at ω_M obtained under maximum acceleration implies failure at the corners of the inner rectangle and hence a failure of the mutual phase to resolve conflicts.

4.5 Tertiary (Propagation) Phase for Velocity Control

Figure 8a shows the velocity axis for a robot RN. RN's current velocity is shown as O in the figure. The portions of the velocity axis shown shaded are those portions of the velocity forbidden from the current state of RN either because they are not reachable or they conflict with other robots. For example portions AB and FG on the axis are not reachable while portions BC, CD and EF conflict with robots R1, R2 and R3 respectively. At O, RN enters into a new conflict with a robot RM. Individual resolution of RN's conflict with RM results in conflict with R1 on the lower side and enters forbidden region on the upper side. Similarly RM's individual resolution leads to conflict with other robots or results in access of forbidden regions. When RN cooperates with RM to resolve the conflict it again results in conflict with R2 on the lower side and R3 on the upper side. In such a scenario RN propagates cooperation request to R1, R2 and R3. The tree structure of figure 8b depicts this propagation. All nodes on the left of RN are requests arising due to lower aspiring velocities while nodes on the right of RN are requests that arise due to higher aspiring velocities. This convention would be followed for all robots involved in the propagation phase. Thus robot RN's resolution of its DC (Direct Conflict) with RM results in indirect conflict (IDC) with robots R1, R2 and R3 and hence RN is considered to be in IDC with R1, R2 and R3. When R1 or R2 try to collaborate in conflict resolution of RN by changing their aspiring velocities it can lead to further conflict with other robots to whom requests are transmitted by R1 or R2 for collaboration. Thus propagation can be recursive and results in a multiple tree like or forest data structure shown in figure 8c. A graph like propagation is avoided since a robot-node that has already propagated a request to another node below does not entertain any new requests.

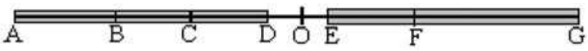

Fig. 8a. The velocity axis of the robot whose current velocity is at O. Shaded represents the inaccessible velocities due to conflicts.

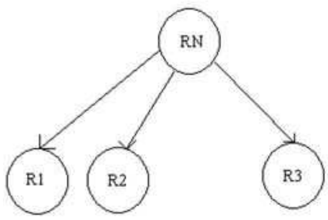

Fig. 8b. RN propagates requests to R1 and R2 on the left due to conflicts with lower velocities and on the right to R3 due to higher velocity.

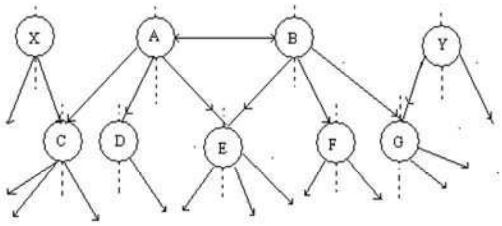

Fig. 8c. Propagation can result in a generalized multiple tree or forest structure whose links represent the flow of conflicts between robots.

Thus any robot has the following functionalities with regard to propagating requests which are taken up for discussion below

- Transmit requests
- Receive requests
- Reply to requests
- Receive replies