

Reactive Collision Avoidance for Multiple Robots by Non Linear Time Scaling

Arun Kumar Singh¹ and K.Madhava Krishna¹

Abstract—Reactive Collision avoidance for non-holonomic robots is a challenging task because of the restrictions in the space of achievable velocities. The complexity increases further when multiple non-holonomic robots are operating in tight/cluttered spaces. The present paper presents a framework specially carved out for such situations. But at the same time can be easily appended with any existing collision avoidance framework. At the crux of the methodology is the concept of non-linear time scaling which allows robots to reactively accelerate/de-accelerate without altering the geometric path. The framework introduced is completely independent of the robot kinematics and dynamics. As such it can be applied to any ground or aerial robot. Through this concept the collision avoidance is framed as a problem of choosing appropriate scaling transformations. We present a "scaled" variant of the collision cone concept which automatically induces distributiveness among robots. The efficacy of the proposed work is demonstrated through simulations of both ground as well as UAVs.

I. INTRODUCTION

In multi-robotic applications, such situations often arise when multiple robots are operating in tight/cluttered spaces. One such situation is shown in figure 1(a) and 1(b) which shows robots moving in a corridor and a cluttered space respectively. In these kind of situations ensuring collision avoidance with constraints on velocities and accelerations becomes a challenging task. Moreover the space itself is constrained since the robots should not collide with the lane boundary or static obstacles. The major motivation behind the current work lies in solving collision avoidance problems with velocity/acceleration control with the additional constraint that the path of the robot never changes. Since the initial path planned for the robot would generally be the most preferable or the most optimal one, ensuring this constraint guarantees that the robots stay in the lane or do not collide with the static obstacles in a cluttered workspace during the collision avoidance maneuvers.

In general it is very difficult to come up with collision avoiding velocities which will not result in a change of path of the robot. Moreover it is even more difficult to execute them because while changing velocities but continuing to remain on path is trivial for straight line trajectories and doable for circular arcs, it becomes formidable for highly non linear curves such as clothoids and splines. Most non holonomic robots move along such curves when subject to non uniform angular velocities and this paper presents an effective formulation for the same. To the best of authors knowledge, a generic framework which allows robots to reactively accelerate/de-accelerate on a given path does not

exist. These are the two fronts where the paper makes the major contribution.

To achieve both the above objectives, a novel concept of non-linear time scaling proposed in our earlier works [14] is used. This concept modifies the collision cone concept [3] to result in such velocities which will not change the path and at the same time provides a methodology to achieve them.

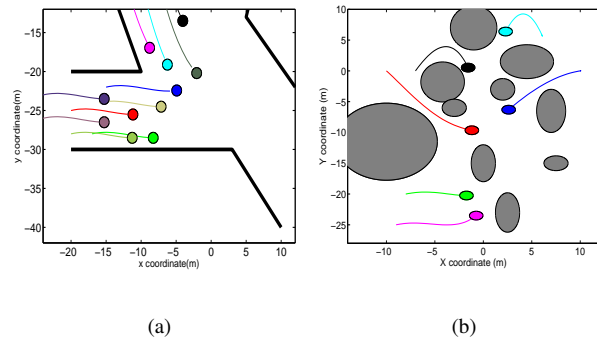


Fig. 1. (a),(b): Multiple robots facing imminent collisions while travelling through lanes/corridors/cluttered spaces. The robots should not collide with the lane boundaries or collide with the static obstacles while performing collision avoidance manoeuvres.

The proposed work goes beyond the standard path velocity decomposition (PVD) and similar methods [6] [5] [9] used to plan collision avoiding velocity profiles along specified paths. However the key difference is that PVD is a planning framework which requires complete information about other robot's trajectories. Moreover it uses arc length parametrization which has requires computing/inverting numerical integrations [19] to plan velocity profiles. If PVD would have to be done reactively, it would require repeated such computations which could become computationally cumbersome, specially for high speed trajectories.

The proposed work on the other hand need not know other robot's trajectory information. Moreover it uses non-linear time scaling which do away with the numerical computations of arc length parametrizations. In-fact the re-activeness of the proposed framework is a direct consequence of the ability of the non-linear time scaling to smoothly accelerate/de-accelerate a robot along a path with very simplified computations. Moreover it can be easily integrated or dovetailed with the kinematics of heterogeneous set of robots. For example in this paper we show the efficacy of this framework through collision avoidance between multiple ground robots and multiple aerial robots. As is well know the kinematic relations of such robots widely vary.

¹ Arun Kumar Singh and K.Madhava Krishna are with Robotics Research Centre, IIT-Hyderabad, India

II. RELATED WORK

The concept of Velocity Obstacle for collision avoidance in a dynamic environment was proposed in [4] and was later extended to the case of reactive collision avoidance among multiple robots by Manocha et al. in [17],[18]. The described methodology was distributive and reactive but required robots to move in straight line trajectories. They used the effective centre approach [7] in their work [15] to extend their framework to differential drive robots. In [18], Manocha et al. proposed to solve the problem of collision avoidance for car-like robots by considering only those accelerations which are attainable from any robot configuration. This technique however strictly restricts the space of feasible accelerations for collision avoidance. A work which considers the full general kinematics of car-like robots for collision avoidance problem has been recently presented in [1]. Although this framework still works with holonomic trajectories, it incorporates the tracking error that a car-like robot will exhibit while tracking these trajectories.

A work which comes closest to the proposed work can be found in [8] where time scaling was used for collision avoidance at the planning level. The algorithm only considered straight line trajectories and required complete knowledge of colliding robot's trajectories. The current work differs greatly from [8] in the sense it performs reactive collision avoidance on non-linearly varying trajectories.

A critical advantage of the current proposed work is that the mathematical equations framed does not depend on the specific robot kinematics and as such it can be applied to any car-like robot, fixed wing UAVs and other such vehicles. Since the framework described here does not require the path of the robots to be changed for collision avoidance, it automatically provides a solution for the problem of path-coordination [11],[13] which has real world applications like [16]. It also provides a solution for problems where collision avoidance has to be embedded in the path following control [12]. Since robots avoids collisions without changing the paths, quality of path following is not compromised.

Other key novelties of the current work includes: Firstly the concept of non-linear time scaling has been modified from our earlier work [14] to work at the acceleration level. This is imperative since at the level of reactive navigation, the control inputs should conveniently be in the form of accelerations. Thirdly, the scaled version of collision cone constraints presented in this paper automatically induces a distributive behaviour in the collision avoidance framework. Distributiveness removes the need for communication between the robots. Each robot can act independently and collisions can still be avoided. Conditions under which a valid scaling transformation exists for collision avoidance is presented for a simple two robot case. For a more general case of multiple robots, we present analytical formulae to check whether collisions can be avoided by scaling transformations.

The rest of the paper is organised as follows: Section III describes the problem formulation and the notations used in the paper. Section IV introduces the concept of non-

linear time scaling and describes the construction of scaling functions. Section V describes the collision cone concept and solution of collision avoidance constraints for two robots. The process of combing the scaling transformations and collision avoidance constraints is explained in section VI. Conditions under which a valid scaling transformations exists for a two robot case is presented in section VII. Section VIII provides closed form expressions for the solution space of collision avoidance constraints in the case of multiple robots. Simulation results are presented in section IX.

III. PROBLEM FORMULATION

Given a system of n robots at initial position $X_0 = (x_{0i}, y_{0i})$, the objective is to guide them without collision to their goal location $X_f = (x_{fi}, y_{fi}) \forall i \in \{1, 2, \dots, n\}$. A initial trajectory for each robot is assumed to be present which connects the robots initial and goal state. This initial trajectory could be obtained independently without considering other robots. A one such planner which produces spline based paths in cluttered space can be found in [10] The initial trajectory of the i_{th} robot would be represented as $\chi_i(t) = (x_i(t), y_i(t))^T$. The instantaneous position and velocity of i_{th} robot would be represented as $x_i, y_i, \dot{x}_i, \dot{y}_i$.

With the above notations in place, the next section describes the process of modifying velocity and acceleration through non-linear time scaling.

IV. VELOCITY AND ACCELERATION MANIPULATION THROUGH SCALING TRANSFORMATION

A change in the independent variable from t to t_n in the trajectory definition $\chi(t)$ does not change the path taken by the robot, but brings the following changes in the velocity and acceleration profile of the trajectory.

$$\dot{\chi}(t_n) = \dot{\chi}(t) \frac{dt}{dt_n} \quad (1)$$

$$\ddot{\chi}(t_n) = \ddot{\chi}(t) \left(\frac{dt}{dt_n}\right)^2 + \dot{\chi}(t) \frac{d^2t}{dt_n^2} \quad (2)$$

The above transformation equations are key to solving the collision avoidance problem. As it can be seen that the transformation just depends on the initial trajectory information. The inherent nature of trajectory itself does not have any influence on the transformation. In other words the initial trajectory could be of an omni-directional robot, car-like vehicle or a fixed wing UAV. Hence the entire framework in this paper which depends on the above two transformations could be applied to any vehicle.

The variable t is the old time while t_n is the new time taken to traverse the trajectory. $\frac{dt}{dt_n}$ is the scaling function, which scales up the velocity and acceleration for $\frac{dt}{dt_n} > 1$ and scales it down for $\frac{dt}{dt_n} < 1$. Since time cannot reverse itself, $\frac{dt}{dt_n}$ has to be a monotonic function. The following two general forms for the scaling function is used in the current work

$$\frac{dt}{dt_n} = k_1 e^{-k_2 t} \quad (3)$$

$$\frac{dt}{dt_n} = p \quad (4)$$

k_1, k_2 and $p > 0$ are constants. In earlier works [14] it has been highlighted that the scaling function as given by (3) is necessary for reactively modifying the velocity and acceleration profile of the trajectory.

As stated earlier, the path of the robot does not change during collision avoidance. Hence the collision avoidance manoeuvre reduces to either accelerating or de-accelerating the robot. The objective then is to create a scaling transformation which results in acceleration/de-acceleration of the robot. A generic framework is next presented which calculates the coefficients k_1 and k_2 which results in continuous and smooth increase/decrease of the velocity until some desired(or safe) velocity is reached from where the scaling function becomes constant and assumes the form of (4). The scaling function can also saturate and become constant when the max/min limit of the velocity is reached. This is illustrated in fig. 2(a) which shows a generic scaling function constructed from an exponential and a constant scaling function.

A. Determining parameters of scaling function

Let $t \in (t_a, t_b)$ i.e the exponential scaling function is defined for the time interval (t_a, t_b) . To determine the coefficients k_1 and k_2 we first identify the following relation

$$\frac{dt}{dt_n} \Big|_{t=t_a} = s_o \rightarrow k_1 = s_o e^{k_2 t_a} \quad (5)$$

In (5), s_o refers to the scale at the start of the interval in which the scaling function is defined. $s_o = 1$ if the scaling transformation is applied to the initial trajectory for the first time. The above relation ensures that there is no velocity discontinuity between the scaled and unscaled velocity profile at the beginning of the defined interval. To determine the coefficient k_2 , it is first noted that the acceleration/de-acceleration produced by the scaling transformation should respect the acceleration bounds and hence the following inequalities are obtained.

$$\ddot{x}_{max} \geq \ddot{x}(t)(k_1 e^{-k_2 t})^2 + \dot{x}(t)(-k_2 k_1^2 e^{-2k_2 t}) \quad (6)$$

if, sign($\ddot{x}(t)$) > 0

$$\ddot{x}_{min} \leq \ddot{x}(t)(k_1 e^{-k_2 t})^2 + \dot{x}(t)(-k_2 k_1^2 e^{-2k_2 t}) \quad (7)$$

if, sign($\ddot{x}(t)$) < 0

Similar inequalities exist for the y component of the trajectory. $\ddot{x}_{max} > 0$ and $\ddot{x}_{min} < 0$ are the maximum acceleration and de-acceleration ability of the robot. (6) and (7) are complicated inequalities which are difficult to solve. Hence a relatively simpler approach is adopted to obtain an approximate solution which is as follows. The interval (t_a, t_b) are divided into n smaller intervals as $(t_a, t_a + \delta, t_a + 2\delta, \dots, t_b)$. At these n points (6) and (7) are symbolically evaluated which gives rise to $2n$ inequalities (n for x and

n for y) in terms of only k_2 (since k_1 depends on k_2 through (5)). These inequalities could be solved as a non-linear optimization problem to obtain the most optimal value of k_2 . Optimal here means minimising k_2 for increasing exponential and maximising k_2 for decreasing exponential as these leads to scaling function with steepest slope and hence maximum acceleration/de-acceleration. This in turn leads to better time-optimality. However since non-linear programming often suffers from the problem of local extrema and non-convergence of constraints, a sampling based method is adopted where a set of values of k_2 is generated and tested for satisfaction of (6) and (7) and then the most optimal value of k_2 is chosen. Beyond time t_b a constant scaling function given by (4) is used with $p = k_1 e^{-k_2 t_b}$.

With the framework for constructing scaling function in place, the collision avoidance problem becomes of choosing what scaling function to use for which robot at any given instant. This is explained in the next section.

V. COLLISION AVOIDANCE

A. Collision Avoidance Between a Robot and a Moving Obstacle

The case of collision avoidance between two robots where one of the robots is just a passive moving obstacle is first considered. This is illustrated in figure 2(b) which shows robot 1 in collision course with passively moving robot 2. Robot 1 is reduced to a point and robot 2 is enlarged by the dimension of robot 1 i.e $R = R_1 + R_2$. Collision is detected through the concept of collision cone [3], which states that the robot 1 and robot 2 is heading for a collision if the relative velocity vector $\vec{V}_{1/2}$ lies inside the collision cone of robot 1 with respect to robot 2, $C_{1/2}$. Although initially stated for straight line trajectories, the collision cone concept can be applied to non-linearly varying trajectories by performing the collision check iteratively.

Collision is avoided if the relative velocity vector $\vec{V}_{1/2}$ is out of the collision cone $C_{1/2}$ which is given by the following condition [2]

$$d^2 = |\vec{r}|^2 - \frac{\vec{r} \cdot \vec{V}_{1/2}}{|\vec{V}_{1/2}|} \geq R^2 \quad (8)$$

where

$$\vec{r} = (x_1 - x_2)\hat{i} + (y_1 - y_2)\hat{j} \quad (9)$$

Noting that the scaling transformation described in the previous section will always result in a velocity that is a scalar multiple of the current velocity, $\vec{V}_{1/2}$ is written as

$$\vec{V}_{1/2} = (s\dot{x}_1 - \dot{x}_2)\hat{i} + (s\dot{y}_1 - \dot{y}_2)\hat{j} \quad (10)$$

The collision avoidance requires solving for the scale factor s from the following quadratic inequality.

$$(x_1 - x_2)^2 + (y_1 - y_2)^2 - R^2 - \frac{((s\dot{x}_1 - \dot{x}_2)(x_1 - x_2) + (s\dot{y}_1 - \dot{y}_2)(y_1 - y_2))^2}{(s\dot{x}_1 - \dot{x}_2)^2 + (s\dot{y}_1 - \dot{y}_2)^2} \geq 0 \quad (11)$$

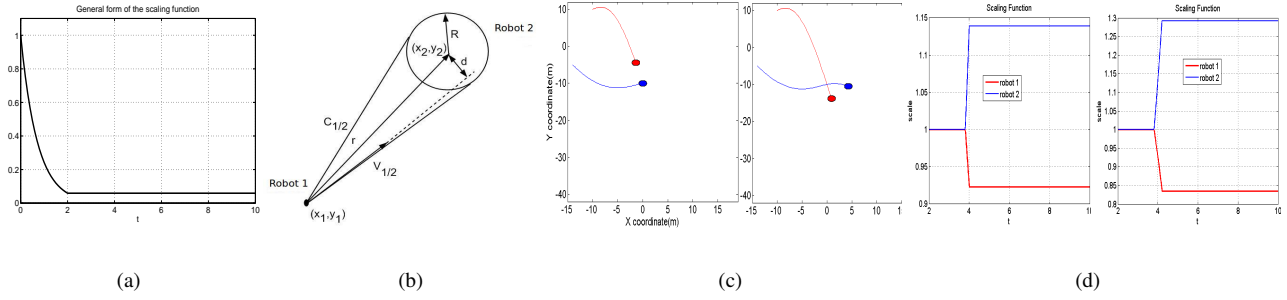


Fig. 2. (a) A generic Scaling function. (b) Collision Cone definition. (c) Shows an example of two robot collision avoidance with scaling transformation. (d) evolution of the scaling function. The final scaling function results from the use of multiple scaling transformations at various instance

(11) can be written in the form $as^2 + bs + c \geq 0$. Noting that it is required that the scale factor $s > 0$, the solution to the quadratic inequality will always be either of the following form

$$s \in [0, -b - \frac{\sqrt{b^2 - 4ac}}{2a}] \cup [-b + \frac{\sqrt{b^2 - 4ac}}{2a}, \infty) \quad (12)$$

, or $[-b + \frac{\sqrt{b^2 - 4ac}}{2a}, \infty)$, if $a > 0$

$$s \in [-b - \frac{\sqrt{b^2 - 4ac}}{2a}, -b + \frac{\sqrt{b^2 - 4ac}}{2a}] \quad (13)$$

or $[0, -b + \frac{\sqrt{b^2 - 4ac}}{2a}]$, if $a < 0$

Let s_{min} denote the scale factor solution which gives the minimum deviation from the current velocity profile. How exactly this scale factor solution is used in collision avoidance is explained in the subsequent sections. However it can be noted at the moment that for $s_{min} > 1$ robot needs to accelerate while for $s_{min} < 1$ de-acceleration is required.

B. Collision Avoidance Between two active robots

The complexity of the collision avoidance increases when robot 2 is also a decision making entity and not a passive obstacle [17]. The manner in which (11) has been framed leads to the following simple lemma which is utilised to solve the collision avoidance problem between two active robots.

Lemma 5.1: If s is the scale factor solution which makes $\vec{V}_{1/2}$ come out of $C_{1/2}$, then $\frac{1}{s}$ is the scale factor solution which makes $\vec{V}_{2/1}$ come out of $C_{2/1}$

Proof: For getting the collision avoidance condition for robot 2, $\vec{V}_{2/1}$ is expressed in the following manner

$$\vec{V}_{2/1} = (s'\dot{x}_2 - \dot{x}_1)\hat{i} + (s'\dot{y}_2 - \dot{y}_1)\hat{j} \quad (14)$$

Substituting (14) in (8) results in the following expression

$$\frac{(x_2 - x_1)^2 + (y_2 - y_1)^2 - R^2}{((s'\dot{x}_2 - \dot{x}_1)(x_2 - x_1) + (s'\dot{y}_2 - \dot{y}_1)(y_2 - y_1))^2} \geq 0 \quad (15)$$

Taking $\frac{1}{s'}$ common from the second term in (15) from the numerator and denominator which eventually gets cancelled out, the following expression is obtained

$$\frac{(x_1 - x_2)^2 + (y_1 - y_2)^2 - R^2}{((\frac{1}{s'}\dot{x}_1 - \dot{x}_2)(x_1 - x_2) + (\frac{1}{s'}\dot{y}_1 - \dot{y}_2)(y_1 - y_2))^2} \geq 0 \quad (16)$$

Comparing (11) and (16), it can be inferred that $s = \frac{1}{s'}$. ■

The significance of this result is that if s_{min} denotes the minimum deviation scale factor for robot 1, then $\frac{1}{s_{min}}$ denotes the minimum deviation scale factor for robot 2. Hence if robot 1 accelerates i.e $s_{min} > 1$, robot 2 automatically chooses de-acceleration $s_{min} < 1$. Thus collision can be avoided even though both the robots act independently.

VI. COMBINING SCALING TRANSFORMATION AND COLLISION AVOIDANCE

Figure 2(c) shows an example of two robot collision avoidance. Since the trajectories are non-linear, a distance threshold is also used along with collision cone constraint for inferring collision. Collision is detected at $t = 3.81$ s. (11)/(15) are used by the robots for calculating s_{min} . In this particular case $s_{min} < 1$ for robot 1 and consequently in line with the theory described above, $s_{min} > 1$ for robot 2. s_{min} denotes the minimum amount the current velocity profile should be scaled in order to avoid collision. Since robots cannot achieve the scaled velocity instantaneously, they accelerate/de-accelerate by constructing a increasing/decreasing scaling function iteratively, while satisfying the acceleration constraints (6) and (7). However instead of always choosing the maximum/minimum acceleration bound as reference, a value which is proportional to magnitude of s_{min} is used.

In this particular case, after collision is detected robot 1 and robot 2 construct a decreasing/increasing exponential scaling function respectively for the time interval [3.81 4.02], followed by constant scaling function as shown in figure 2(d). While the robots are evolving according to the new scaled trajectory, they recompute s_{min} and apply a new scaling transformation on the scaled trajectory and the process is

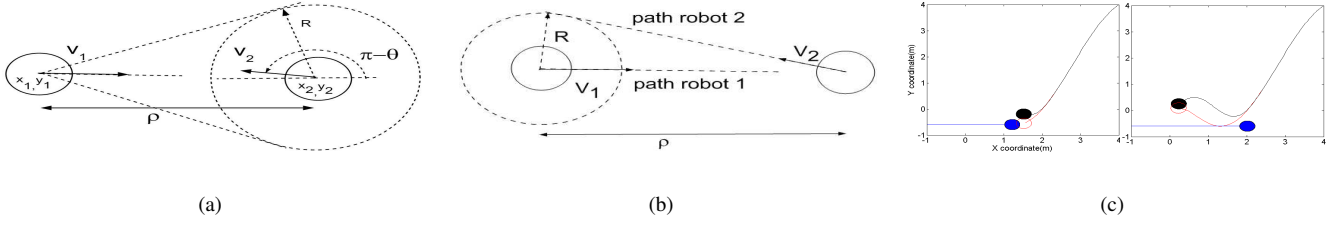


Fig. 3. (a) Condition for two robot case to avoid collision through scaling transformations. (b): Minimum angular deviation that the paths should have at the point of collision detection for collision to be avoided by scaling transformations. (b) Path Perturbation to avoid head on collisions

repeated till a safe velocity is reached. From figure 2(d) it can be seen that a new scaling function is created in the interval [4.02 4.23], followed by a constant scaling function. In figure 2(d) scaling function is plotted against the original time variable t .

VII. EXISTENCE CONDITION FOR COLLISION AVOIDING SCALING TRANSFORM

A. For Two Robot Case

In this section, we analyse conditions under which collisions can be avoided by scaling transformations. In other words we are interested in finding conditions under which, there exists atleast one positive root ($s > 0$) for (11). As mentioned in the previous section, (11) can be put in the form $as^2 + bs + c \geq 0$. So $\frac{c}{a} < 0$ ensures that atleast one positive root exists. To visualise this condition geometrically, consider a simple two robot case shown in figure 3(a). Without loss of generality, we assume that the centre of both robots lie on x axis. Let $\rho \geq R$ be the sensing range of the robot 1. The coordinates of both the robots can then be written as x_1, y_1 and $(x_1 + \rho), y_1$ respectively. Let robot 1 velocity be along the x axis. The coefficients for this situation is given by

$$a = \rho^2 \sin^2(\theta) - R^2 \quad (17)$$

$$c = -R^2 \quad (18)$$

From (17)-(18) it can be seen that $\frac{c}{a} < 0$ as long as

$$\sin^2 \theta > \frac{R^2}{\rho^2} \quad (19)$$

In other words scaling transformations can lead to collision avoidance as long as the velocity vectors of the robot are deviated from each other by a minimum amount given by (19). Another way to look at this result is that since scaling transformation does not result in the change of path, collision can be avoided only if at the point of collision detection, the angular deviation of the paths are greater than that given by (19). This is illustrated in figure 3(b).

It can be easily seen that if $\theta = 0$, i.e both robots are on a head-on collision course, $\frac{c}{a} > 0$ and no positive solution exists. In such cases it will be necessary to alter the path. Usually it is convenient to locally perturb the path. For a spline based paths this can be achieved by perturbing the knot points [10]. One such implementation is shown in 3(c) which shows two intersecting trajectories having a head on-collision

segment. The new trajectory shown in black is computed by locally perturbing the initial trajectory.

For a generic multi-robot case, we present formulae which allows us to check whether any positive roots exists. This is explained in the next section.

B. For a Generic Multi-Robot Case

Let a robot be in collision with n robots. Extending the collision avoidance involves finding the solution space of s resulting in a scaled velocity profile which is outside all the n collision cones. This requires finding the intersection of the solution region of the n quadratic inequalities of the form of (11). The structure of the constraints which are in the form $a_i s^2 + b_i s + c_i \geq 0 \forall i = 1, 2, \dots, n$ allows us to have a closed form expression for the solution space of s . For this consider the following cases

- Case 1. $a_i > 0 \forall i = 1, 2, 3, \dots, n$

The solution space of s in this case will be of the form

$$s \in [0, \min\{-b_i - \frac{\sqrt{b_i^2 - 4a_i c_i}}{2a_i}\}] \quad (20)$$

$$\cup [\max\{-b_i + \frac{\sqrt{b_i^2 - 4a_i c_i}}{2a_i}, \infty)$$

If $\min\{-b_i - \frac{\sqrt{b_i^2 - 4a_i c_i}}{2a_i}\} < 0$, the solution space is of the form

$$[\max\{-b_i + \frac{\sqrt{b_i^2 - 4a_i c_i}}{2a_i}, \infty) \quad (21)$$

No feasible solution exists if $\max\{-b_i + \frac{\sqrt{b_i^2 - 4a_i c_i}}{2a_i}\} < 0$. In this case solution space will only comprise of negative values of s .

- Case 2. $a_i < 0 \forall i = 1, 2, 3, \dots, n$

The solution space is of the form

$$[\max\{-b_i - \frac{\sqrt{b_i^2 - 4a_i c_i}}{2a_i}, \min\{-b_i + \frac{\sqrt{b_i^2 - 4a_i c_i}}{2a_i}\}] \quad (22)$$

If $\max\{-b_i - \frac{\sqrt{b_i^2 - 4a_i c_i}}{2a_i}\} < 0$, the solution space is of the form

$$[0, \min\{-b_i + \frac{\sqrt{b_i^2 - 4a_i c_i}}{2a_i}\}] \quad (23)$$

Even in this case no feasible solution exists if $\max\{-b_i + \frac{\sqrt{b_i^2 - 4a_i c_i}}{2a_i}\} < 0$

• Case 3. $a_i > 0$ for m constraints and $a_i < 0$ for $m - n$ constraints.

For clarity let coefficients arising out of m constraints be indexed with i while that arising from $m - n$ constraints be indexed with j . Then the solution space of s will in the following form

$$s \in [s_{jmax}, s_{imin}] \cup [s_{imax}, s_{jmin}], \quad (24)$$

where

$$s_{jmax} = \max\{-b_j - \frac{\sqrt{b_j^2 - 4a_j c_j}}{2a_j}\} \quad (25)$$

$$s_{imin} = \min\{-b_i - \frac{\sqrt{b_i^2 - 4a_i c_i}}{2a_i}\} \quad (26)$$

$$s_{jmin} = \min\{-b_j + \frac{\sqrt{b_j^2 - 4a_j c_j}}{2a_j}\} \quad (27)$$

$$s_{imax} = \max\{-b_i + \frac{\sqrt{b_i^2 - 4a_i c_i}}{2a_i}\} \quad (28)$$

A sufficient condition for solution space to be null is

$$s_{jmax} > s_{imin}, \text{ and, } s_{imax} > s_{jmin} \quad (29)$$

No feasible solution exists if $s_{jmin} < 0$.

VIII. EXTENDING COLLISION AVOIDANCE TO MULTIPLE ROBOTS

In the generic multi-robot case each robot computes $s_{min}^i, \forall i = 1, 2, \dots, n$ corresponding to every other n robots. Avoiding collision with all the robots requires computing the intersection region of all the s_{min}^i planes. A linear programming based approach is constructed to compute the intersection region of these half planes. In case the intersection region is empty, we search for a solution that is closest to satisfying all the half plane constraints.

IX. SIMULATION RESULTS

The initial trajectories from start to the goal for the robots were obtained by modelling the robots as unicycle. The trajectories were obtained by [14] by modifying it to account for obstacles. With the help of obtained initial trajectories the framework has been tested in following challenging scenarios. A video of the simulation can be found in https://www.youtube.com/watch?v=uS1CG-x01_k

A. Robots Moving through lanes/corridors

Figure 4(a) shows a system of robots successfully avoiding collisions at the intersection of a lane. In line with the theory described, the robots de-accelerate/accelerate by constructing increasing/decreasing scaling functions for avoiding collisions. Any change in the scaling function from the default scaling function $s = 1$ suggests that collision avoidance maneuver is initiated. The maximum and minimum acceleration bounds were kept at $\pm 1m/s^2$ respectively. If during the collision avoidance, a robot cannot compute a

scaling transformation satisfying the acceleration bounds or the max/min limit of the velocity is reached, the robot continues to move with its current velocity and acceleration profile. In such cases collision can be avoided if the other robot takes full responsibility. The scaling functions used by the robots are shown in figure 4(b) with respect to the old time variable t . It can be seen from figure 4(c) and 4(d) that a smooth and continuous velocity profile is maintained while avoiding collisions. For the sake of clarity only the final scaled velocities are shown. This is one of the key advantages of the proposed work since a smooth velocity profile increases the tracking fidelity of the controller.

B. Robots Moving in Cluttered Space

Figure 5(a) shows robots performing collision avoidance in a space cluttered with static obstacles. The initial trajectories obtained for the robots avoided all static obstacles. To show the necessity of time scaling based collision avoidance in these situations, consider figure 5(a) which shows robots shown in cyan and black on collision course. As stated earlier, modifying velocities while preserving the path is extremely difficult without the scaling transformations. Hence the usual implementation of velocity obstacle [4] or collision cone concept would require considering the static obstacles as well. The fact that initial trajectories avoided all static obstacles is of little significance in this case. Figure 5(b) shows the solution space for the robot shown in cyan, obtained from collision cone considering static obstacles. The solution space is shown as an intersection of two resultant inequalities. The first solution space refers to the solution when the robot comes out of the collision cone by turning the relative velocity vector towards left of the current heading direction, while the second refers to the solution which will turn the relative velocity vector turn towards right of the current heading direction. For 5(b), it can be seen the first case solution requires very high velocities. In fact no solution would be obtained in this case if the constraint of max/min velocity of $\pm 4m/s$ is enforced. For the second case no solution is obtained since the resultant inequality lines do not have a common intersection region. It can also be conferred by the noting that in figure 5(a), space is highly restricted for the robot shown in cyan on the right side of its current heading.

The solution obtained from the scaled version of collision cone constraints(inequality (11)) proposed in this paper does not require to include static obstacles. As a result a significant improvement in solution space is obtained as shown in figure 5(a). All the velocities on the shown straight line are a solution of the collision avoidance. In this particular case, solutions which are a scaled down version of current velocity are shown. The solution space corresponding to scaled up velocities violated the max/min velocity constraint and hence are not shown. This particular example clearly shows that in these situations the scaling transformation based collision avoidance are not only useful but also imperative for safe navigation of multiple robots.

The scaling function used by the robots in this particular

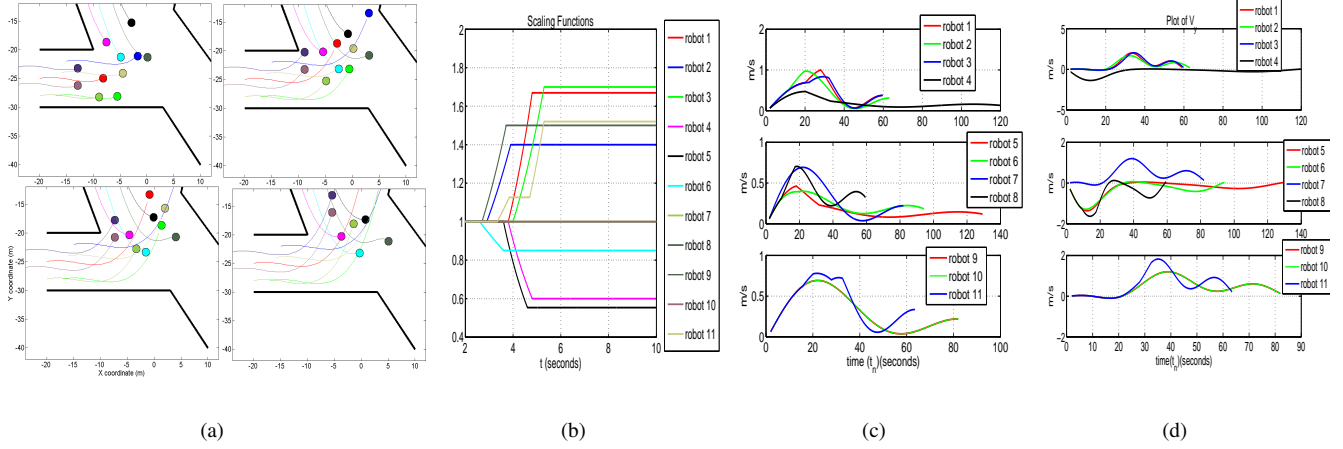


Fig. 4. (a) Simulation snapshots of robots performing collision avoidance in a lane. (b) Final scaling functions for each robot. (c-d) velocity profiles for each robot

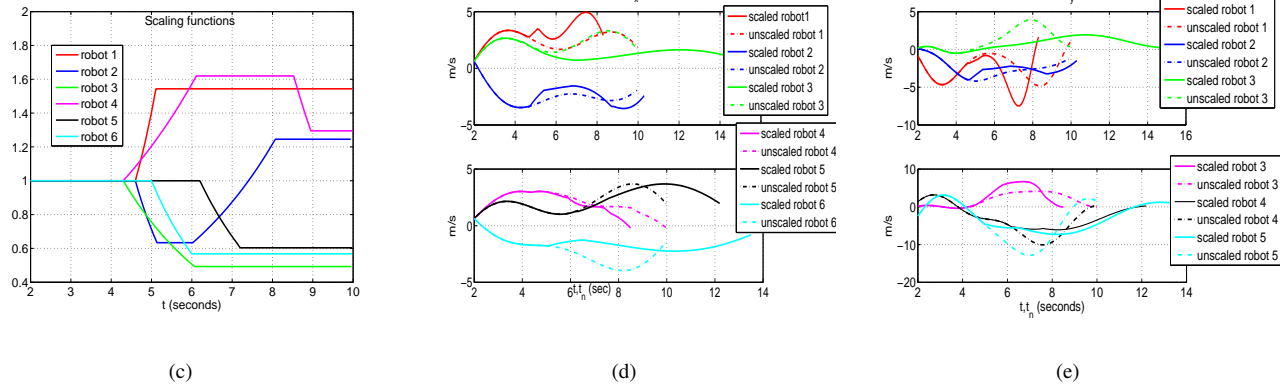
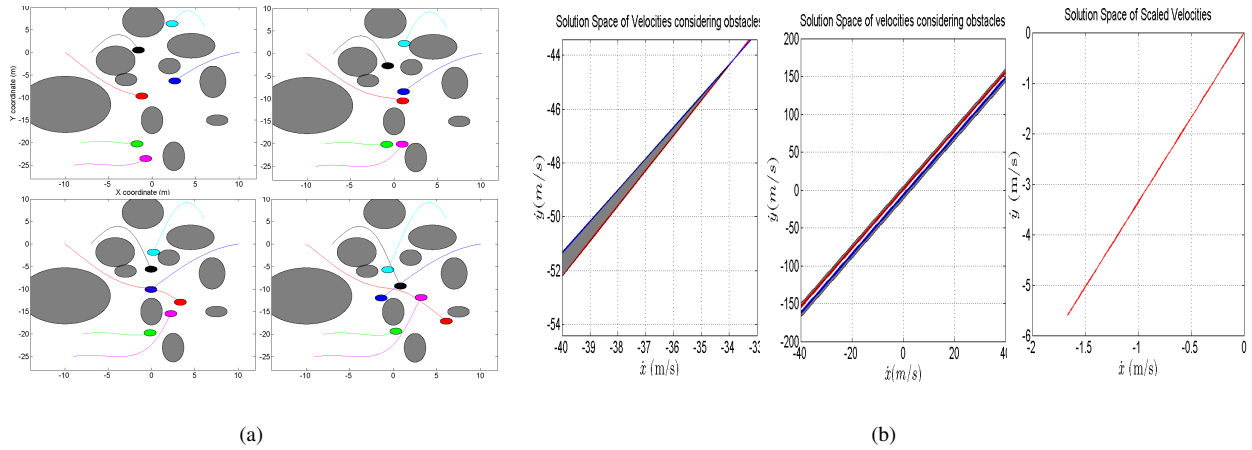


Fig. 5. (a) Simulation snapshots of robots performing collision avoidance in a cluttered workspace. (b) shows the solution space comparison between the usual velocity obstacle/collision cone implementation which has to consider the static obstacles and the scaled collision cone constraints (11) proposed in the paper. The solution space is shown for the cyan robot which is on collision course with the robot shown in black. Figure shows a significant improvement of solution space of scaled collision cone constraints. (d) shows the final scaling function for each robot. (e-f) scaled and unscaled velocity profiles for each robot.

case are shown in figure 5(c) while the scaled and unscaled velocity profiles are shown in figure 5(d)-5(e).

C. Collision Avoidance for Multiple UAVs

Figure 6(a) shows multiple UAV's avoiding collision with each other. The initial trajectories for the UAV's are calculated by considering the following simplified point mass model

$$\begin{cases} \dot{x} = v \cos \psi \cos \alpha, \dot{y} = v \sin \psi \cos \alpha, \dot{z} = v \sin \alpha \\ u_1 = \dot{v}, u_2 = \dot{\psi}, u_3 = \dot{\alpha} \end{cases} \quad (30)$$

Here ψ and α are heading and pitch angles. u_1, u_2, u_3 are the control inputs. The collision cone approach of [2] was used for 3D obstacle avoidance.

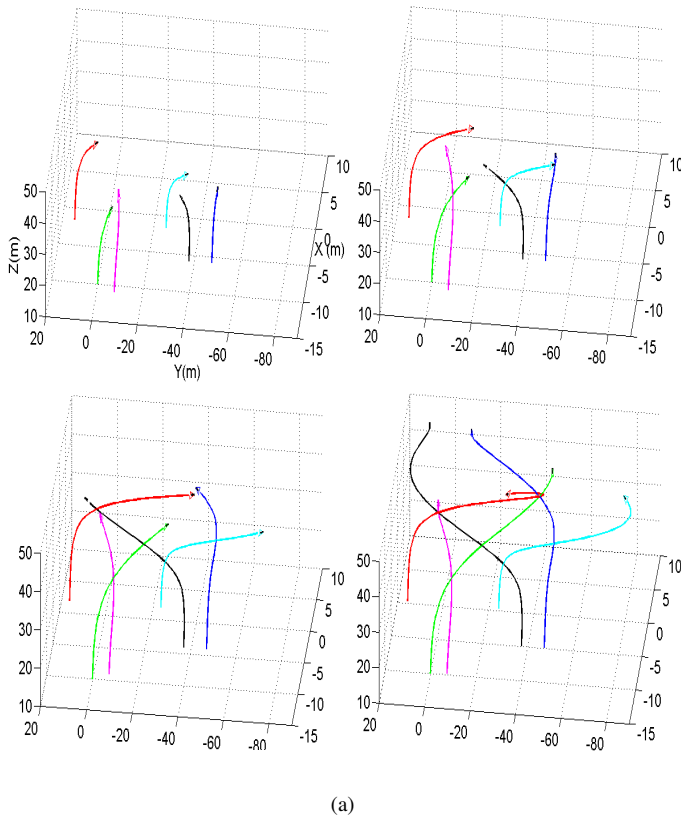


Fig. 6. (a) 3D collision avoidance for multiple UAVs

X. CONCLUSION AND FUTURE WORK

This paper proposes a novel concept of non-linear time scaling based collision avoidance. The robots only modify their velocities during the collision avoidance while preserving their original path. The fact that this methodology can be applied for any arbitrary non-linear trajectory makes it applicable to wide range of ground/aerial vehicles. A scaled version of collision cone constraints were presented which automatically and elegantly resulted in distributive behaviour in collision avoidance. Several scenarios were highlighted like high speed collision avoidance on curved trajectories where the presented methodology can be extremely useful.

The future work pertains to using the versatile nature of non-linear time scaling to multi-robot formation control embedded with collision avoidance. Since collisions can be avoided without changing the path, it is believed that it will help in preserving the geometry of formation.

REFERENCES

- [1] J. Alonso-Mora, A. Breitenmoser, P. Beardsley, and R. Siegwart. Reciprocal collision avoidance for multiple car-like robots. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 360–366. IEEE, 2012.
- [2] C. Carbone, U. Ciniglio, F. Corraro, and S. Luongo. A novel 3d geometric algorithm for aircraft autonomous collision avoidance. In *Decision and Control, 2006 45th IEEE Conference on*, pages 1580–1585. IEEE, 2006.
- [3] A. Chakravarthy and D. Ghose. Obstacle avoidance in a dynamic environment: A collision cone approach. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 28(5):562–574, 1998.
- [4] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772, 1998.
- [5] Y. Guo and L. E. Parker. A distributed and optimal motion planning approach for multiple mobile robots. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 3, pages 2612–2619. IEEE, 2002.
- [6] K. Kant and S. W. Zucker. Toward efficient trajectory planning: The path-velocity decomposition. *The International Journal of Robotics Research*, 5(3):72–89, 1986.
- [7] B. Kluge, D. Bank, E. Prassler, and M. Strobel. Coordinating the motion of a human and a robot in a crowded, natural environment. *Advances in Human-Robot Interaction*, pages 231–234, 2005.
- [8] I. Komlósi and B. Kiss. Motion planning for multiple mobile robots using time-scaling. *Mobile Robots Control Architectures, Bio-Interfacing, Navigation, Multi Robot Motion Planning and Operator Training, J. Bedkowski, Ed. InTech, 2011*, 1998.
- [9] K. Madhava Krishna, R. Alami, and T. Simeon. Safe proactive plans and their execution. *Robotics and Autonomous Systems*, 54(3):244–255, 2006.
- [10] J. Pan, L. Zhang, and D. Manocha. Collision-free and smooth trajectory computation in cluttered environments. *The International Journal of Robotics Research*, 31(10):1155–1175, 2012.
- [11] J. Peng and S. Akella. Coordinating multiple robots with kinodynamic constraints along specified paths. *The International Journal of Robotics Research*, 24(4):295–310, 2005.
- [12] K. Sakurama and K. Nakano. Deadlock-free path-following control for collision avoidance of multiple robots. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 5673–5678. IEEE, 2009.
- [13] T. Simeon, S. Leroy, and J. Laumond. Path coordination for multiple mobile robots: A resolution-complete algorithm. *Robotics and Automation, IEEE Transactions on*, 18(1):42–49, 2002.
- [14] A. Singh, K. Krishna, and S. Saripalli. Planning trajectories on uneven terrain using optimization and non-linear time scaling techniques. In *To appear in Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012.
- [15] J. Snape, J. van den Berg, S. Guy, and D. Manocha. Smooth and collision-free navigation for multiple robots under differential-drive constraints. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4584–4589. IEEE, 2010.
- [16] D. Soltero, S. Smith, and D. Rus. Collision avoidance for persistent monitoring in multi-robot systems with intersecting trajectories. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3645–3652. IEEE, 2011.
- [17] J. Van Den Berg, S. Guy, M. Lin, and D. Manocha. Reciprocal n-body collision avoidance. *Robotics Research*, pages 3–19, 2011.
- [18] J. van den Berg, J. Snape, S. Guy, and D. Manocha. Reciprocal collision avoidance with acceleration-velocity obstacles. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3475–3482. IEEE, 2011.
- [19] H. Wang, J. Kearney, and K. Atkinson. Arc-length parameterized spline curves for real-time simulation. In *5th international conference on Curves and Surfaces*, 2002.