MULTI-AGENT COLLISION AVOIDANCE USING INTERVAL ANALYSIS TECHNIQUE FOR 3D ROBOTS

Mayanka Medhe[†]

[†]Department of Computer Science and Engineering, Indian Institute of Technology-Bombay

Undergraduate thesis under the guidance of Prof. Leena Vachhani

ABSTRACT

Path planning of multi-agent systems is a fast emerging field in robotics. The main motivation behind this problem is that a collection of many mobile robots can collaborate to perform complex tasks which are used in many industrial applications like Military Logistic Planning [6], coverage [7][8], motion planning [9] [10]. These agents can be static or dynamic. In this technique, intervals are defined according to the robot's maximum change in velocity. They are 3D sectors representing the pose of a robot in a fixed time-interval. An inclusion test is performed which gives the interval of overlap between them which is used to get the free interval.

Index Terms— Collision Detection, Collision Avoidance, Dynamic Obstacle, Interval Analysis, Multi-agent

1. INTRODUCTION

Complex robotic tasks are best performed when a group of robots collaborate together. While doing so, it is important that the robots are able to detect and avoid collisions. This was my motivation to solve this problem for 3D robots. The expected motion of robot can be defined as an interval based on its current velocity and position. I used these intervals to calculate the overlap with different robots to get the collision free interval.

The research objective of my thesis is to develop a collision avoidance technique using interval arithmetic for 3-dimensional robots, which can be further categorized into:

(1) Developing a single technique that is applicable to different types of agents

(2) Providing collision avoidance solution that gives the flexibility to select a strategy based on the application

(3) Developing a methodology for parallel and embedded implementation

In this technique, there is an interval associated with every robot which is a spherical sector defined according to the robot's maximum variance in velocity. This interval is further divided into sub-intervals which facilitate parallelism in performing multiple interval inclusion tests and in handling multiple agents. The solution of the interval inclusion tests is used to find a set of sub-intervals where a collision might occur. The interval of guaranteed no collision is calculated by taking its complement. There can be different types of agents in the test environment such as static agent, dynamic agent with a predictable trajectory, randomly moving agent and agents performing a collaborative task.

In this implementation, I am considering one primary agent and all the other robots as secondary agent. All the interval tests are performed with respect to the primary agent. Section 3 describes in detail the formulation of the interval, interval inclusion test to get sub-intervals of possible collision and the calculation of the collision free interval.

2. LITERATURE SURVEY

This section provides the implementation details of multi-agent collision avoidance for 2D robots using interval analysis. It also describes other techniques used for collision detection and avoidance.

In the case of 2D robots, the intervals are 2D sectors as described in Figure 1. The interval is represented as $[Radius_{min}, Radius_{max}] * [\Theta_{min}, \Theta_{max}]$. The intervals are further subdivided into sub-intervals to bring parallelism. The sub-intervals are basically the boundaries of the interval. The interval inclusion test considers the intersection of all pairs of sub-interval and takes the XNOR of them. Using this test, the free interval is calculated by recursively dividing the sub-interval into smaller parts to get the interval of collision and subsequently the collision free interval.

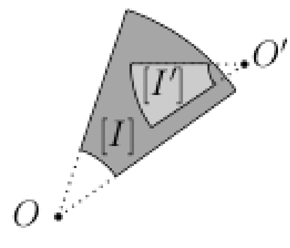

Fig. 1. Intersecting 2D intervals of 2 robots

Another way is to consider offline or reactive (sensor-based) methods. A popular reactive navigation method [11] is based on using repulsive potential fields (PF) from obstacles and attractive PF from the goal. The PF based techniques encounter the problem of local minima and oscillatory behavior in narrow passages. Moreover, determining a single potential function relating to the infinitely many possibilities of moving obstacle's trajectory is a highly demanding task. Some methods to address the oscillation and local minima problems have been introduced but they generally require substantial computation.

My algorithm focuses on extending the 2-dimensional interval analysis technique to 3-dimensions. It is described in details in the following sections.

3. METHODOLOGY

The pipeline of my technique is- calculating the intervals of the primary and the secondary agent, finding the intervals which overlap between them and finally calculating the free interval. Each step is elaborated below-

3.1. Interval definition

The interval of an agent is defined as a part of a 3D spherical sector because the range of the sensor used for distance calculation is of the similar shape. Let the instantaneous coordinates of the primary agent be $O(x, y, z, \theta, \alpha)$. Considering the maximum change in the linear and angular velocity of the primary agent to be v_m and ω_m in the time interval ΔT, the agent can cover the distance between $(v - v_m)\Delta T$ to $(v + v_m)\Delta T$ and can rotate atmost by $\omega_m \Delta T$. The angle of elevation is assumed to be constant in my implementation. Therefore, the interval I can be defined as-

$$[I] = [(v - v_m)\Delta T, (v + v_m)\Delta T]* \\ [(\theta - \omega_m \Delta T), (\theta + \omega_m \Delta T)] * [\alpha, \alpha]$$

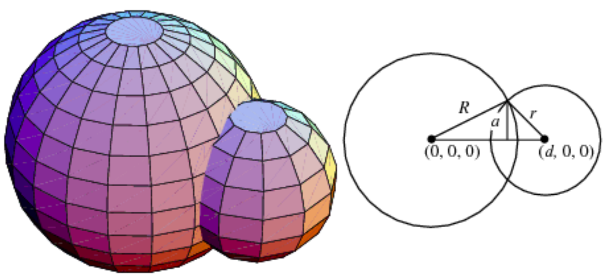

Fig. 2. Intersection of 2 spherical surfaces from [12]

3.2. Inclusion Test

The interval is divided into 4 sub-intervals- inner arc, outer arc, lowermost line of the conical surface and the uppermost line of the conical surface. Therefore, the following interactions are possible- line segment with line segment, spherical arc with spherical arc and line with spherical arc.

The line segment to line segment interaction is simply calculated by checking if they are intersecting. The line segment to spherical arc is also calculated by solving their equations in 3D. For checking the intersection of the 2 spherical arc, first the 2 spheres are solved to find the equation of the circle at the points of their intersection. The equation of the circle is later checked if it is within the boundaries of the spherical sector. It is illustrated in Figure 2. The equation of the circle is given by-

$$x = \frac{d^2 - r^2 + R^2}{2d}$$

$$y^2 + z^2 = \frac{(4d^2R^2 - (d^2 - r^2 + R^2))}{4d^2}$$

The inclusion test considers all possible combinations of sub-interval interactions and is given by-

$$[e_{I'}](I) = \bigcap_{i=1}^{4} \bigcap_{j=1}^{4} s_{ij}$$

where s_{ij} is the interaction of i^{th} sub-interval of I with j^{th} sub-interval of I' and $[e_{I'}](I)$ is the inclusion test function for testing inclusion of I' in I.

3.3. Collision free interval calculation

The free interval is calculated using algorithm 1- (adapted from the 2D implementation described in Literature Survey).

Algorithm 1 Interval Inclusion

function INTERVAL_INCLUSION($[I], [I'], N_r, N_\theta$)
  if $[e_{I'}]([I]) = 1$ **then**
    [J] = [0, 0, 0][0, 0, 0];
  else
    $N \leftarrow N_r; n \leftarrow 0$
    for $i = 1 to 2$ **do**
      $bisect [I] in [L] and [R]$
      while $n < N$ **do**
        bisect [L] in [LL] and [LR] and [R] in [RL] and [RR]
        if $[e_{I'}]([R]) = 1$ **then**
          $[L] \leftarrow [LL]; [R] \leftarrow [LR]$
        else if $[e_{I'}]([L]) = 1$ **then**
          $[L] \leftarrow [RL]; [R] \leftarrow [RR]$
        else
          if $[e_{I'}]([LL]) \neq 1$ **then**
            $[L] \leftarrow [LL]$
          else
            $[L] \leftarrow [LR]$
          end if
          if $[e_{I'}]([R]) \neq 1$ **then**
            $[R] \leftarrow [RR]$
          else
            $[R] \leftarrow [RL]$
          end if
        end if
        $n \leftarrow n + 1$
      end while
      $[J_i] = [\underline{L_i}, \overline{R_i}]; N \leftarrow N_\theta; n \leftarrow 0$
    end for
  end if
end function

4. RESULTS

I have tested the algorithm for different cases. I have illustrated a few cases in Table 1. The first case is when there is no overlap between the intervals. The second case is when there is a partial overlap between the two intervals and the third case is when there is a complete overlap.

Table 1. Collision free interval calculation for primary agent at origin

$O(x')$	I	I'	$[e_{I'}](I)$
[2,0,0]	[0.5,1]*[0,90]	[0.5,1]*[0,90]	1
[2,0,0]	[0.5,1]*[0,90]	[0.5,2]*[0,180]	[0,1]
[0.75,0,0]	[0.5,1]*[0,90]	[0.1,0.2]*[0,90]	0

5. CHALLENGES AND FUTURE WORK

One major challenge I faced was considering intervals which are oblique with respect to one another. Therefore, for now I am considering only such orientations with are head on. I plan to make a more generic method in my BTP-2 project. Another challenge I faced was to calculate the intersection of the conical surfaces. I tried to circumvent the problem by considering the intervals as thin spherical shells but I was not able to extend the 2D algorithm for calculating the interval inclusion for the same. Hence, I choose to consider just the topmost and bottom-most line segment of the conical surface when viewed from the plane perpendicular to plane containing the centres of the 2 spheres.

I will later simulate my technique on different subsets of the agents. The simulations will be performed on Gazebo ROS. Later on, I will test the technique on FPGA-driven differential drive mobile robots.

6. CONCLUSION

In a general autonomous robotic application there are various robotic agents which makes the collision detection and avoidance algorithms of utmost importance. In order to reduce the effect of the error in the measurements of the velocity and position, I am considering the maximum possible velocity. It will help in calculating the interval where there is guaranteed no collision.

7. REFERENCES

[1] John Snyder, Adam Woodbury, Kurt Fleischer, Bena Currin, and Alan Barr, "Interval methods for multi-point collisions between time-dependent curved surfaces," *Computer Graphics (SIGGRAPH '93 Proceedings)*, vol. 27, 03 1997.

[2] Kerianne Hobbs Stanley Bak, "Efficient n-to-n collision detection for space debris using 4d aabb trees (extended report)," *arXiv preprint arXiv:1901.10475*, 2019.

[3] Vachhani L. Sinha A. Nagavarapu, S., "Multi-robot graph exploration and map building with collision avoidance: A decentralized approach," *J. In- tell. Robotic Syst. 83(3-4), 503–523*, 2015.

[4] Tom Duff, "Interval arithmetic recursive subdivision for implicit functions and constructive solid geometry," *SIGGRAPH Comput. Graph.*, vol. 26, no. 2, pp. 131–138, July 1992.

[5] Tom Duff, "Interval arithmetic recursive subdivision for implicit functions and constructive solid geometry," in *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1992, SIGGRAPH '92, pp. 131–138, ACM.

[6] Terri Lenox, S. Hahn, Michael Lewis, Terry Payne, and Katia Sycara, "Task characteristics and intelligent aiding," 01 2000.

[7] T. M. Cheng and A. V. Savkin, "Decentralized control of mobile sensor networks for asymptotically optimal blanket coverage between two boundaries," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 365–376, Feb 2013.

[8] Nuwan Ganganath, Chi-Tsun Cheng, and Chi Tse, "Distributed anti-flocking algorithms for dynamic coverage of mobile sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 12, pp. 1795–1805, 01 2016.

[9] J. Kim and J. Lee, "Trajectory optimization with particle swarm optimization for manipulator motion planning," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 620–631, June 2015.

[10] N. Ganganath, C. Cheng, and C. K. Tse, "A constraint-aware heuristic path planner for finding energy-efficient paths on uneven terrains," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 601–611, June 2015.

[11] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, March 1985, vol. 2, pp. 500–505.

[12] Wolfram MathWorld, "Sphere-sphere intersection," [Online; accessed November 28, 2019].