Collision Avoidance While Reaching a Target (Project Final Report)

Ahmet Burak KOÇ
Computer Engineering Department
Hacettepe University
Ankara, Turkey
burak.koc@hacettepe.edu.tr

***Abstract*—This work proposes an algorithm for autonomous vehicles to avoid collision with moving obstacles. One of the most critical features of autonomous vehicles is to avoid collision to stay passengers and systems safe. Avoiding standing obstacles is important but collision risk with moving obstacles is higher in a dynamic environment such as traffic, air traffic or space.**

***Keywords*— obstacle avoidance, collision avoidance, autonomous vehicles, moving obstacles**

I. Introduction

In this study, collision avoidance will be performed by controlling acceleration of the vehicle while reaching the target. Simulation will be performed to visualize results and observe how vehicle update its motion plan.

II. Problem Definition

A. *Define Vehicle Properties*

Vehicles' properties such as mass and volume should be defined due to effect on dynamics and collision constraints of vehicles.

In this study, we will not consider vehicle dynamics such that we will not compute required forces and torques to obtain desired acceleration. Calculation and illustration will be done in 2D from top view so volume will not be considered as well. However, vehicles will be defined as circles and collision constraint will be defined using distance between centers of two circles. If distance between two circle centers gets lower than 2r where r is radius of each circle.

B. *Reaching a Target*

Main vehicle will try to reach a target with constant speed. In the meantime, moving obstacle will be moving with constant speed as well.

C. *Collision Avoidance*

Main vehicle shall predict collision and choose acceleration or decelaration to avoid collision.

III. Solution

In this work, two method is studied to avoid collision. In the first method, Vehicle A tried to avoid collision with Vehicle B. In the second method, Vehicle 1 tried to avoid collision with Vehicle 2.

In the first method, distance function between vehicles is calculated and it is tried to avoid collision without changing trajectory but only accelerating or decelerating and total change tried to be minimized to save energy. However, resulting quadratic equations were complex to solve and the results were not very satisfying. In the second method, relative velocity region of collision is determined so velocity kept outside this region to avoid collision. In this method, both magnitude and direction of velocity changed with time so trajectory also changed. Finally with second method, satisfying results are obtained.

A. *Method 1*

Positions are being changed as given in Figure 1. Vehicles A and B do not change their direction. Vehicle B moves with constant velocity in +x direction. Vehicle A travels in +y direction but it changes its accelaration to avoid collision. Note that, acceleration is either negative or positive value of a constant value or zero.

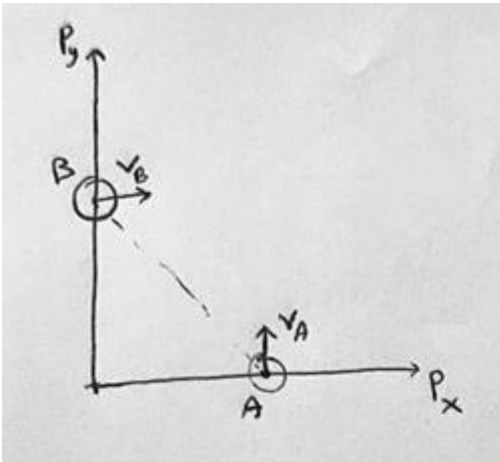

Figure 1: Positions of A and B

A starts with initial velocity $\vec{v}_A$ and its velocity relative to B is $\vec{v}_{A|B} = \vec{v}_A - \vec{v}_B$ (Figure 2).

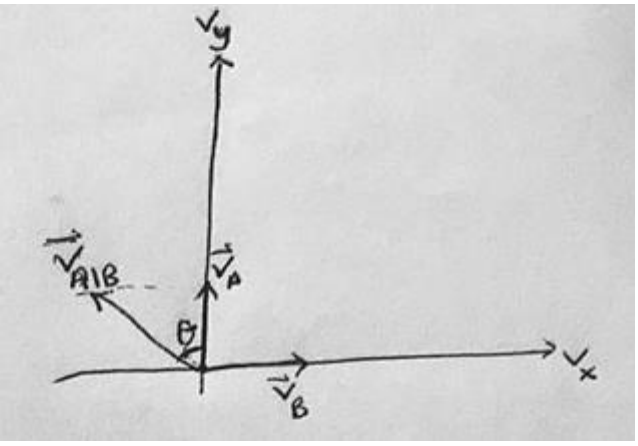

Figure 2: A's velocity relative to B

$\vec{a}_B$=0 and $\vec{a}_A$ has two components one parallel to $\vec{v}_{A|B}$ and one orthogonal (Figure 3).

There is an angle θ between vectors $\vec{a}$ and $\vec{v}_{A|B}$. Distance between A and B can be calculated as given in (1) and (2). Both formula includes $\vec{a}_A$ because it has components along both line (Figure 4).

$d_1 = d_{init} - (\lvert\vec{v}_A - \vec{v}_B\rvert)t - \frac{1}{2}a_A \cos\theta\, t^2$	(1)
$d_2 = \frac{1}{2}a_A \sin\theta\ t^2$	(2)

Where d_1 is distance between A and B along $\vec{v}_{A|B}$ line and d_2 is distance between A and B orthogonal to $\vec{v}_{A|B}$ line.

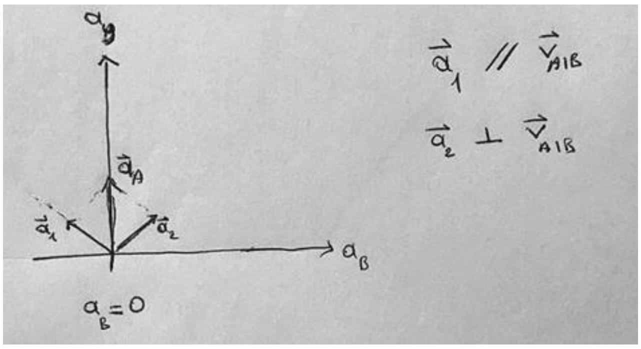

Figure 3: $\vec{a}_A$ components parallel and orthogonal to $\vec{v}_{A|B}$

Hence, distance between A and B is calculated as in **Eq. 3.**

$d = \sqrt{d_1^2 + d_2^2}$	(3)

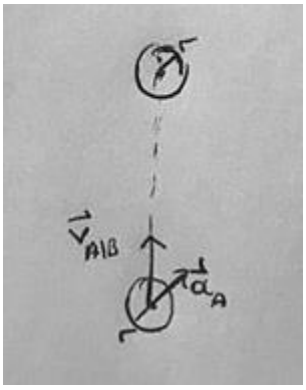

Figure 4: Relative velocity and acceleration

To avoid collision, distance between A and B should never drop below sum of radiuses of circles. Collision avoidance constraint is as given (**Eq. 4**)

$d > r_1 + r_2$	(4)

Where r_1 *and* r_2 are radius of vehicles that defines collision constraint.

It is assumed that theta is 45° and angle between two vectors $\overrightarrow{v_A}$ *and* $\overrightarrow{v_B}$ is also 45°. By using (2),(3) and (4) we derived collision inequality (5).

$\sqrt{t^4(a^2) + t^3\left(Va\sqrt{2}\right) + t^2\left(V^2 - d_{init}a\sqrt{2}\right) + t(-2d_{init}V) + d_{init}^2} > r_1 + r_2$	(5)

Where velocity magnitude is $\lvert\vec{v}_A - \vec{v}_B\rvert = V$. By taking square of two sides of equation (5), quartic equation form (6) can be obtained and 4th degree polynomial's roots can be solved to observe if there is any real root that results in collision.

$at^4 + bt^3 + ct^2 + dt + e = 0$	(6)

Where $a = (a^2)$, $b = \left(Va\sqrt{2}\right)$, $c = \left(V^2 - d_{init}a\sqrt{2}\right)$, $d = (-2d_{init}V)$, $e = d_{init}^2 - (r_1 + r_2)^2$.

It is desired that (5) has no real roots so it cannot satsify collision constraint for positive real values of t. To analyze solution of (5), discriminant is calculated as $\Delta_4 = 256a^3e^3 - 192a^2bde^2 - 128a^2c^2e^2 + 144a^2cd^2e - 27a^2d^4 + 144ab^2ce^2 - 6ab^2d^2e - 80abc^2de + 18abcd^3 + 16ac^4e - 4ac^3d^2 - 27b^4e^2 + 18b^3cde - 4b^3d^3 - 4b^2c^3e + b^2c^2d^2$. If $\Delta_4 > 0$, there is either all four roots are real or none is. Discriminant variation vs accelaration values is calculated on Matlab so accelaration that prevents collision conditions could be found (Figure 5).

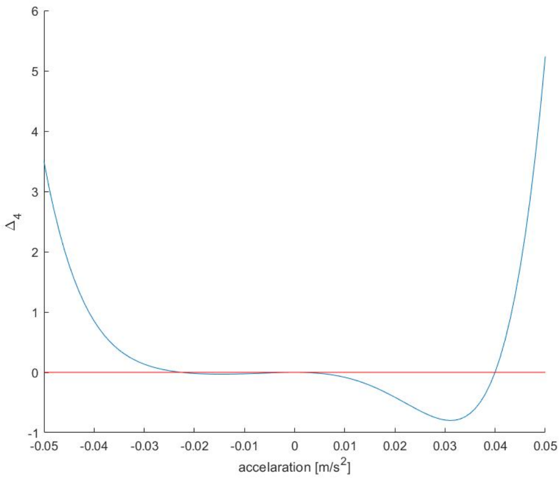

Figure 5: Discriminant vs Accelaration

And results are analyzed also in Matlab (Figure 6). If blue line decrease below red line, the collision occurs.

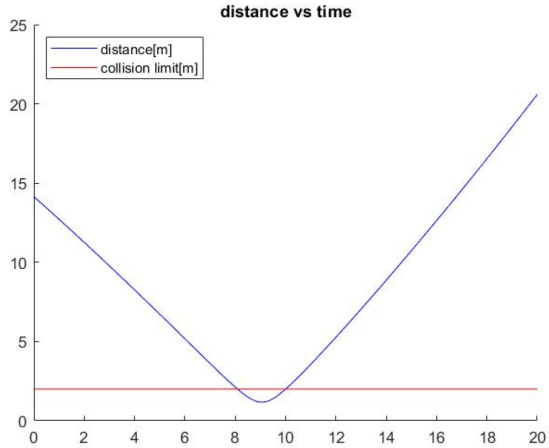

Figure 6: Distance vs Time

The acceleration values that satsifies $\Delta_4 > 0$ condition, is tried to avoid collision but it is observed that there is failed to avoid collision.

This approach can be improved by going into more detailed analysis on roots of quartic equation but it is found to be too complex for our purpose.

B. Method 2

Second method is implemented to avoid collision. In this method, vehicle 1 determine its preferred velocity based on reference speed value and target position. Reference speed value is given by user.

Similarly to method 1, vehicle 1 knows positions and velocities of vehicle 1 and vehicle 2. Then, vehicle 1 computes velocity interval that wil result in collision so that it modify its velocity to closest point that is closest to preferred velocity for target and not in collision interval.

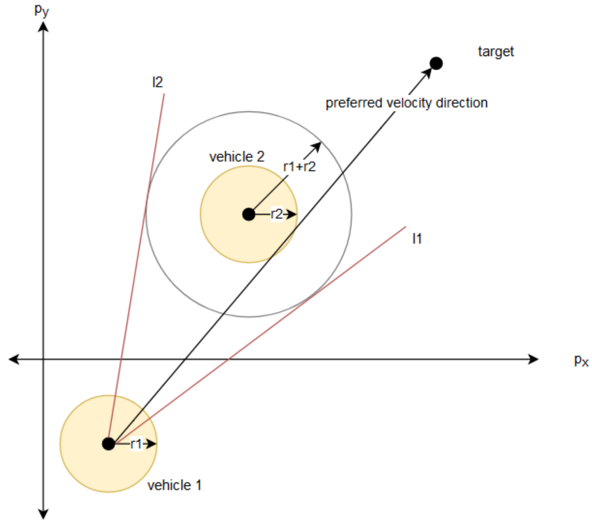

Figure 7: Positions of vehicles

It is obvious that velocity of vehicle 1 cannot be in any direction between line 1 and line 2 in order to avoid collision (Figure 7). The collision velocity interval of vehicle 1 is shifted from origin by velocity of vehicle 2 because collision according to relative velocity (Figure 8).

First preferred velocity is determined with reference speed value in target direction using equations 6,7 and 8.

$v_{1x,pref} = v_{ref} * \sin\Theta$	(7)
$v_{1y,pref} = v_{ref} * \cos\Theta$	(8)
$\Theta = \arctan\dfrac{p_{1y,target} - p_{1y}}{p_{1x,target} - p_{1x}}$	(9)

Then, preferred velocity is shifted by v_2 to find relative velocity and checked if its in collision interval. If it's not in collision region, velocity is not updated. Otherwise, closest points outside the collision region are calculated by equations 9-14. Then, the closer one is chosen as new velocity. For example in Figure 9, $\mathrm{v}_{1,\mathrm{new1}} = v\_n1$ is chosen because it is closer to preferred velocity.

$v_{1x,rel} = v_{1x,pref} - v_{2x}$	(10)
$v_{1y,rel} = v_{1y,pref} - v_{2y}$	(11)
$\mathrm{v}_{1\mathrm{x,new1}} = \mathrm{v}_{1\mathrm{x,rel}} + \dfrac{(v_{1y,rel} - m_1 * \mathrm{v}_{1\mathrm{x,rel}}) * m_1}{m_1^2 + 1}$	(12)
$\mathrm{v}_{1\mathrm{y,new1}} = \mathrm{v}_{1\mathrm{y,rel}} - \dfrac{(v_{1y,rel} - m_1 * \mathrm{v}_{1\mathrm{x,rel}})}{m_1^2 + 1}$	(13)
$\mathrm{v}_{1\mathrm{x,new2}} = \mathrm{v}_{1\mathrm{x,rel}} + \dfrac{(v_{1y,rel} - m_2 * \mathrm{v}_{1\mathrm{x,rel}}) * m_2}{m_2^2 + 1}$	(14)
$\mathrm{v}_{1\mathrm{y,new2}} = \mathrm{v}_{1\mathrm{y,rel}} - \dfrac{(v_{1y,rel} - m_2 * \mathrm{v}_{1\mathrm{x,rel}})}{m_2^2 + 1}$	(15)

After choosing nearest velocity that will avoid collision in raletive velocity graph, it is transformed back to velocity of vehicle 1 by using equations 9 and 10.

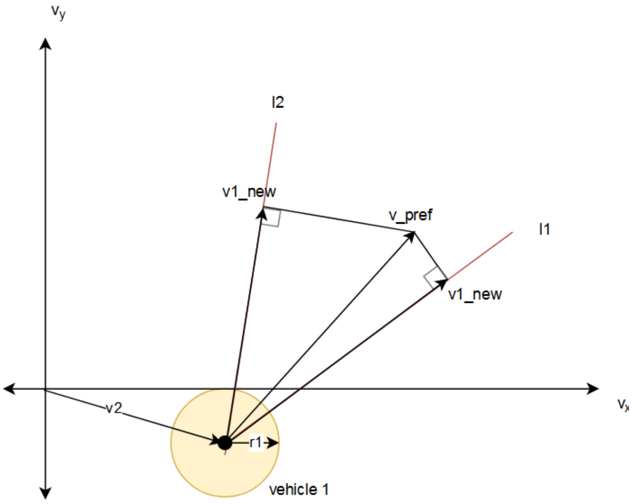

Figure 8: Velocity 1 and Velocity 2

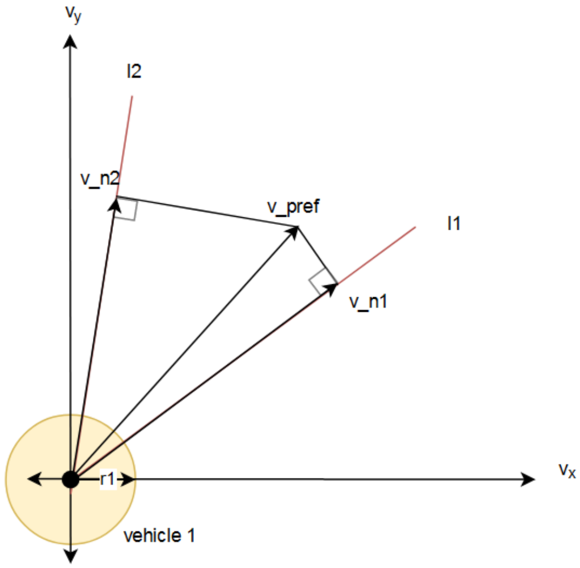

Figure 9: Relative velocity of vehicle 1

The methodology of the collision avoidance is as explained. Implementation of method 2 is explained under Implementation title.

IV. Implementation

Method 2 is implemented on MATLAB/Simulink. Firstly initialization script “simin.m” should be executed. Then, Simulink file will be ready to run. Codes are given as Appendices.

For our experiment, initial velocities are given to satisfy collision condition so that we can observe vehicle 1 modifies its velocity to avoid collision and reach the target without collide to vehicle 2.

Vehicle kinematics are implemented as in Figure 10. For vehicle 2, constant velocity is used.

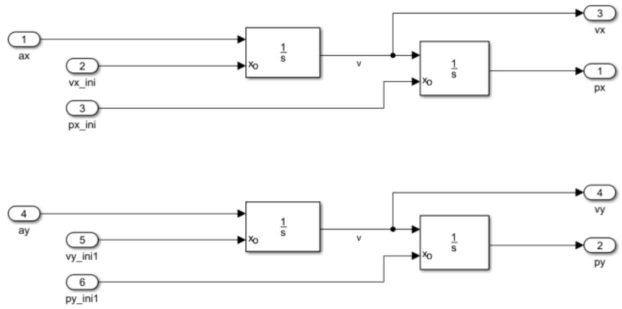

Figure 10: Vehicle kinematics

The main implementation of the method 2 is implemented on block in Figure 11. In this block, new velocity that is closest to preferred velocity outside the collision region is calculated.

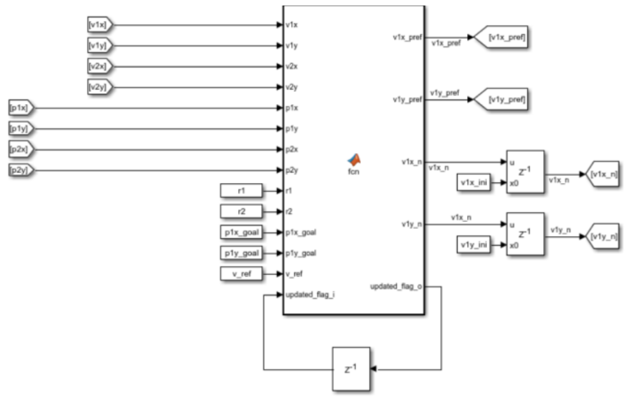

Figure 11: New velocity calculation block

According to new velocity and current velocity of the vehicle, required acceleration is calculated by block given in Figure 12. Note that, in this study, controller is not designed but acceleration value is calculated in very basic way.

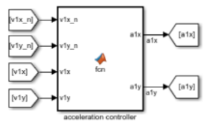

Figure 12: Acceleration controller

When vehicle 1 is sufficiently close to the target, it is decided as success and simulation is stopped (Figure 13).

Figure 13: Simulation success condition

Distance between two vehicles is observed to see if it exceeds collision limit (Figure 14).

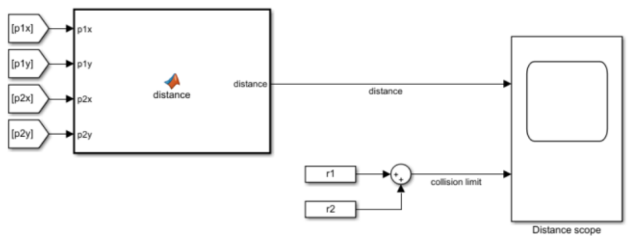

Figure 14: Distance between two vehicles and collision limit

Other parameters such as current velocities, calculated velocities, accelerations and positions are also observed by scope but there is no need to show those blocks.

V. Results

Figure 15 and Figure 16 show how velocity is updated during simulation. Vehicle 1 successfully avoided collision (Figure 17) and reached its target (200,100) (Figure 18).

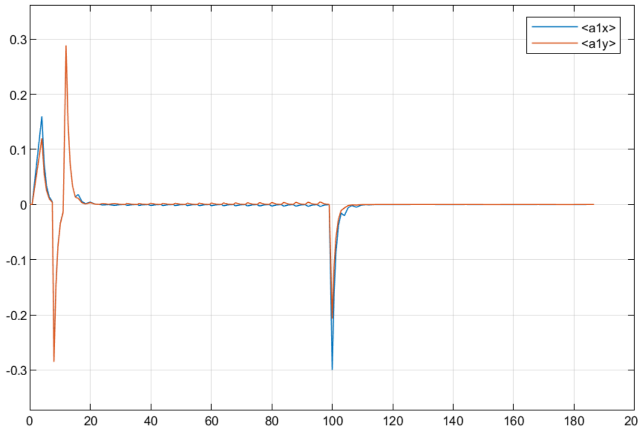

Figure 15: Acceleration vs time (Vehicle 1)

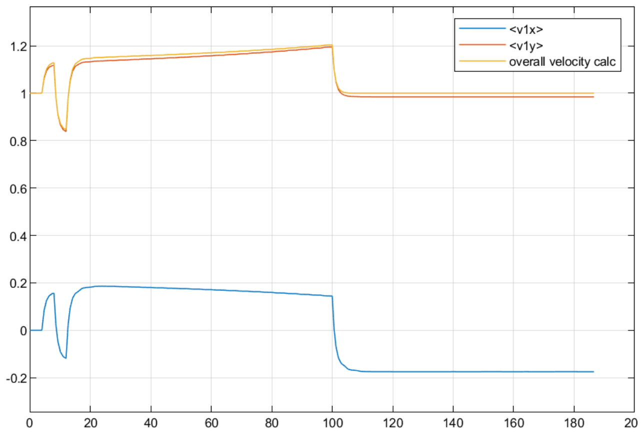

Figure 16: Velocity vs time (Vehicle 1)

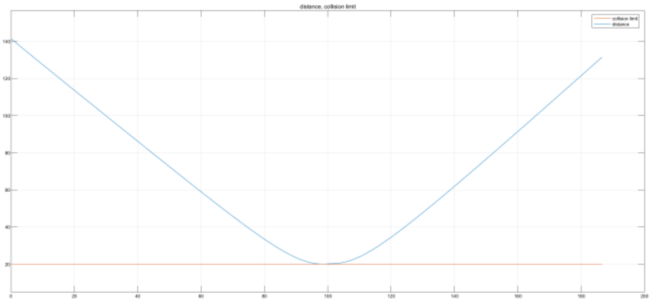

Figure 17: Distance between vehicles vs time

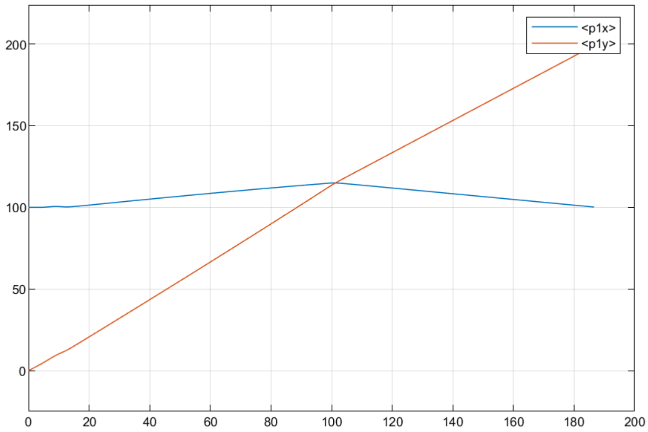

Figure 18: Position vs time (Vehicle 1)

.

VI. Results & Illustration

Illustration script is firstly designed on GNU Octave. Then, it is moved to MATLAB. Vehicles are defined as circles. Using position and time arrays obtained from simulation, we illustrate how vehicle 1 (red) avoids collision and reach the target while vehicle 2 (blue) is moving with constant speed (Figure 19).

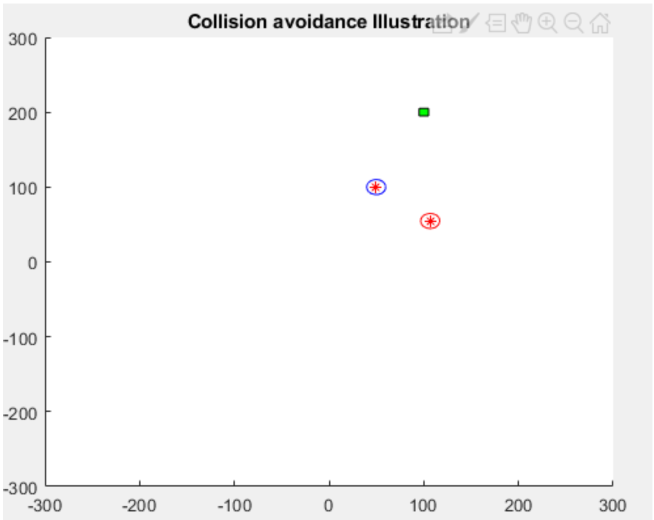

Figure 19: Collision Avoidance Illustration Example

VII. Further Study

This topic can be studied further by improving and adding several factors to problem. For example, advanced steering mechanisms (Ackermann) and vehicle dynamics can be added to the problem. Controllers can be designed to reach updated velocities.

Also updating velocity by only steering rather than changing overall speed of the vehicle can be implemented. In that case, new velocity value would be calculated not by the nearest point outside the collision region but calculated by rotating preferred velocity until it reached outside the collision region.

REFERENCES

[1] Jamie Snape, Jur van den Berg, Stephen J. Guy, and Dinesh Manocha, "The hybrid reciprocal velocity obstacle," IEEE Trans. Robot., vol. 27, 2011.

[2] Jamie Snape, Jur van den Berg, Stephen J. Guy, and Dinesh Manocha, "Independent navigation of multiple mobile robots with hybrid reciprocal velocity obstacles," IEEE/RSJ Int. Conf. Intelligent Robots and Systems, St. Louis, Mo., 2009.

[3] X. Sun, G. Wang, Y. Fan, D. Mu and B. Qiu, "Fast Collision Avoidance Method Based on Velocity Resolution for Unmanned Surface Vehicle," 2019 Chinese Control And Decision Conference (CCDC), 2019, pp. 4822-4827, doi: 10.1109/CCDC.2019.8832993.

[4] C. Suo, W. Xu, Y. Guan, L. He and Y. Yang, "Real Time Obstacle Avoidance and Navigation with Mobile Robot via Local Elevation Information," 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2019, pp. 2896-2901, doi: 10.1109/ROBIO49542.2019.8961624.

[5] R. Chancharoen, V. Sangveraphunsiri, T. Navakulsirinart, W. Thanawittayakorn, W. Boonsanongsupa and A. Meesaplak, "Target tracking and obstacle avoidance for mobile robots," 2002 IEEE International Conference on Industrial Technology, 2002. IEEE ICIT '02., 2002, pp. 13-17 vol.1, doi: 10.1109/ICIT.2002.1189851.