Abstract

The goal of this project is the development of a system that contains N robots and M moving obstacles – humans. The main challenge of this problem comes from the restrictions. The first one is that the N robots should remain in a uniform chain. This means that the robots should move all together towards the same point (the second one follows the first one, the third one follows the second one and so on). The second restriction is that the M people should not intersect with the robot chain. At this point, it is important to note that the people have independent direction and speed from the robots. Also the first robot of the chain makes an independent movement. This movement can be either completely random or takes it from point A to point B. The rest of the robots aim to follow the robot in front of them and at the same time avoid people in near distance (their field of view).

Introduction

In this project we analyse a widely studied project which deals with the navigation of robots autonomously in an environment. More specifically, the goal is the simulation of a system that contains N robots that move in a chain shape and M people – objects of avoidance.

This algorithm can be used in many real world applications, for example the development of a truck chain. In this particular application, only the first truck is been driven by a human being and the rest of them automatically follow the truck in front of them. At the same time, they need to avoid any obstacles and cars.

There are numerous studies that propose algorithms for the solution of this problem. One of the methods consists of a gradient tracking algorithm based on artificial harmonic potential fields [1].

The proposed method uses a vector analysis for both the robots and the people. This is really important when this algorithm needs to be used with really small robots that have weak processors. This method uses only additions, multiplications and basic trigonometry, without even inverse trigonometry which is time consuming, especially for a small processor.

Problem Formulation

The proposed system consists of N autonomously moving robots and M moving objects of avoidance in a two-dimensional space. At any given time t the objects can be characterized by position (x, y) speed U and rotation θ.

The goal of the robots is to move in a platoon formation by following one another. At the same time, they need to avoid collision with obstacles and other robots.

Main Assumptions

With the purpose of creating this program in a reasonable amount of time, we were forced to make some basic assumptions. This does not mean that the program is not expandable. The code is modular and the mathematical theory is general in order to have capabilities for future improvements. More specifically, during the development of the program and the creation of the mathematical foundation we assume that:

1. Robots - humans are represented with a dot in a 2 dimensional space.

2. Every object can be represented with a position (x, y) speed U and rotation θ.
3. Each robot has restrictions in the amount of acceleration and the angle of rotation.
4. Every robot has a camera that can identify where is the robot in front of it and what is his current speed it. This camera can only identify robots in a certain distance and angle, and these variables are parameterized in the algorithm.
5. Due to the above restriction, the robots need to be initialized in distances and angles that are visible from the following robot.
6. Also every robot has a sensor dedicated to identify humans in every possible angle and in a distance bigger than the distance between robots. This information is used from the corresponding robot to start avoiding as soon as possible intersection with a human.
7. Every robot knows where exactly is in the space and what is its current speed.
8. Also due to real world limitations, every robot has limitations in its movements. More specifically the robots cannot move-rotate more than , which again is parametrized in the algorithm.

Methodology

After an extensive research we decided that one of the best approaches for the development of this particular problem is a vector analysis.

In this problem, every object in the two dimensional space has a rotation matrix that comes from the angle of rotation Θ in relation to $x'x$ axes.

$$R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

Also every object has a position (x, y) two axis (R_x, R_y) and a speed u which is always parallel to the R_x.

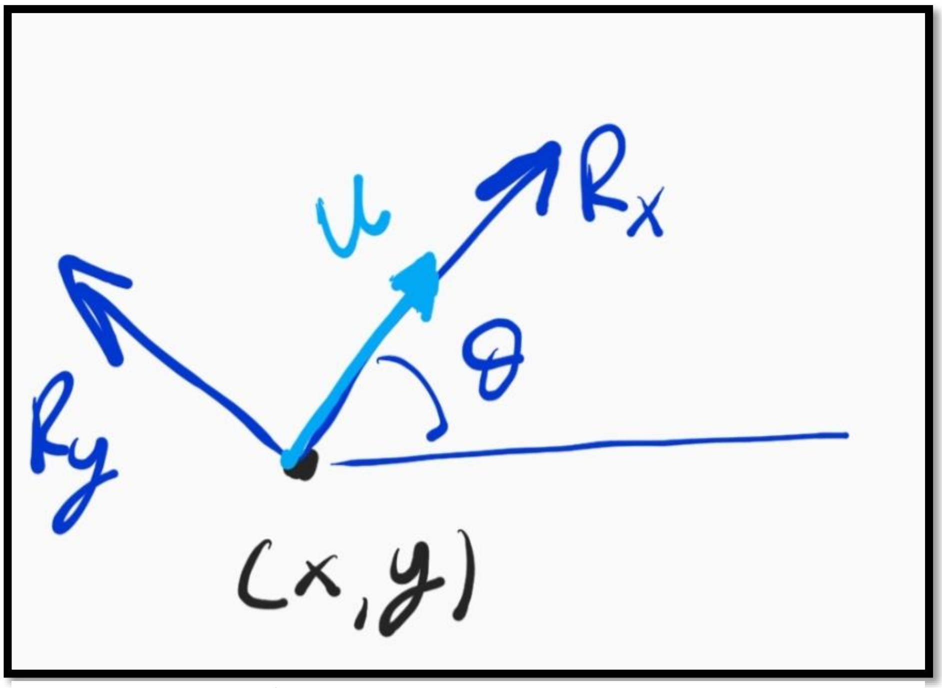


Figure 1. Features in any Object

With that being said, every object in the space is described with nine points: position (x, y) , speed $(u_x, u_y) = (u, 0)$ and rotation matrix $(R_{xx}, R_{xy}, R_{yx}, R_{yy},)$.

Furthermore, we can calculate the next position of the object by adding the old position of the object with the multiplication of the speed array, rotation array and time that have passed between the two points.

$$\begin{bmatrix} x_{new} \\ y_{new} \end{bmatrix} = \begin{bmatrix} x_{old} \\ y_{old} \end{bmatrix} + \begin{bmatrix} u \\ 0 \end{bmatrix} \times \begin{bmatrix} R_{xx} & R_{xy} \\ R_{yx} & R_{yy} \end{bmatrix} \times T$$

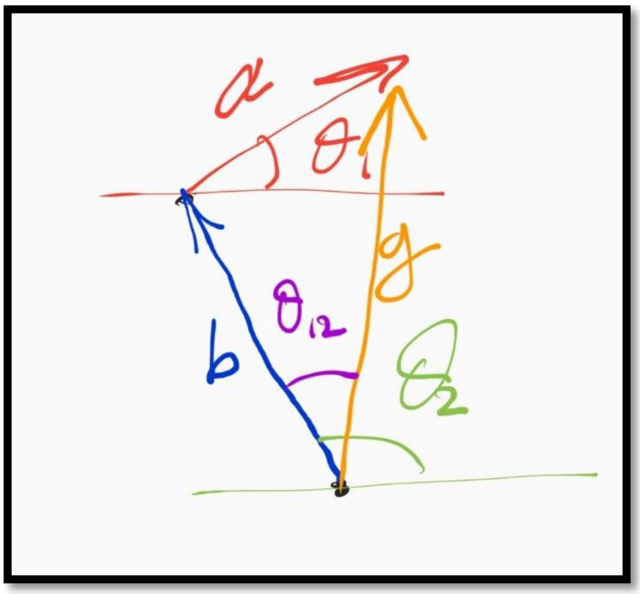
Relation between Robots

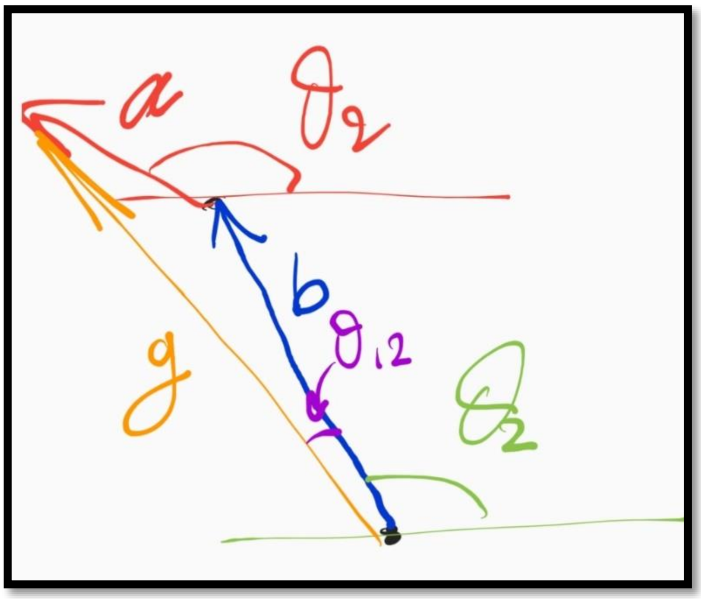
At this point, it is really important to analyse the relation between robots. We can assume that the robot "B" needs to follow the robot "A".

After an extensive research for a realistic model that enables the robots to follow one another, it was decided that addition in the vector space will be used. This is a really good model for three main reasons:

- Takes into consideration the speed and rotation of both robots in the system.
- They can be used easily with the current analysis that has already been done (position, rotation matrix, etc.)
- Has linear results that seem reasonable to be used in the real world.

For the implementation of this method we need the position, speed and rotation of the robot "A" and the position, speed and rotation of the robot "B". With an addition in the vector space, we will calculate the direction "G" which is the new direction of "B".





$$\boldsymbol{g} = \boldsymbol{a} + \boldsymbol{b} = (x_1 + x_2, y_1 + y_2)$$

One of the most important parts of this analysis is the angle θ_{12} between "B" and "G". More specifically, the following equation of inverse tangent needs to be used:

$$\theta_{12} = \tan^{-1}\frac{y_2 + y_1}{x_2 + x_1}$$

$$If \;\; (g_x > 0 \;\&\; g_y > 0)$$

$$\theta_{12} = |\theta_{12}|$$

$$If \;\; (g_x < 0 \;\&\; g_y > 0)$$

$$\theta_{12} = 180 - |\theta_{12}|$$

$$If \;\; (g_x < 0 \;\&\; g_y < 0)$$

$$\theta_{12} = 180 + |\theta_{12}|$$

$$If \;\; (g_x > 0 \;\&\; g_y < 0)$$

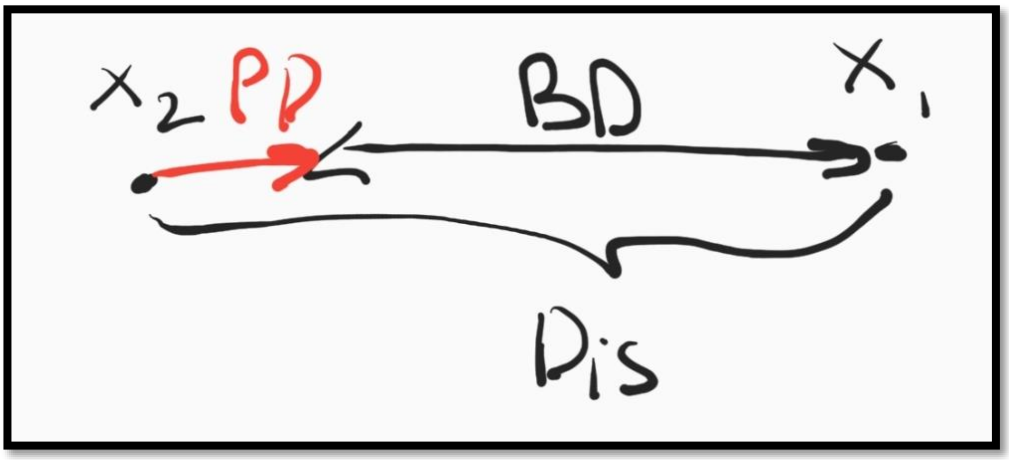
$$\theta_{12} = 360 - |\theta_{12}|$$

Also an analysis has been made about the speed of the robot correlated with the percentage of distance from the rear robot.

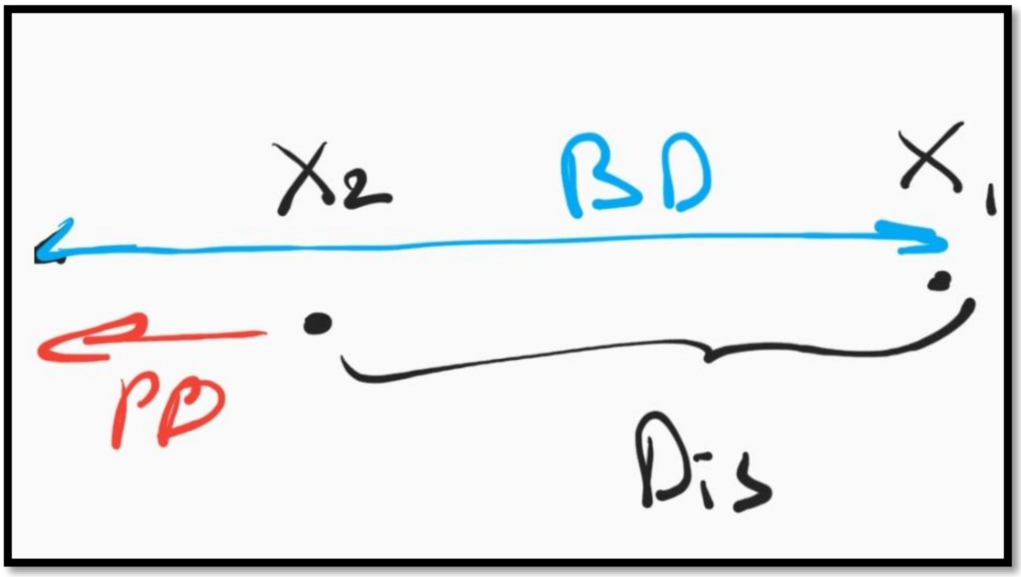
First of all, the distance of the robot is calculated with the euclidean distance:

$$Distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Also depending on the robots and applications that will be used, we can initialize a corresponding "$BestDistance$" which indicates the best possible distance between robots.

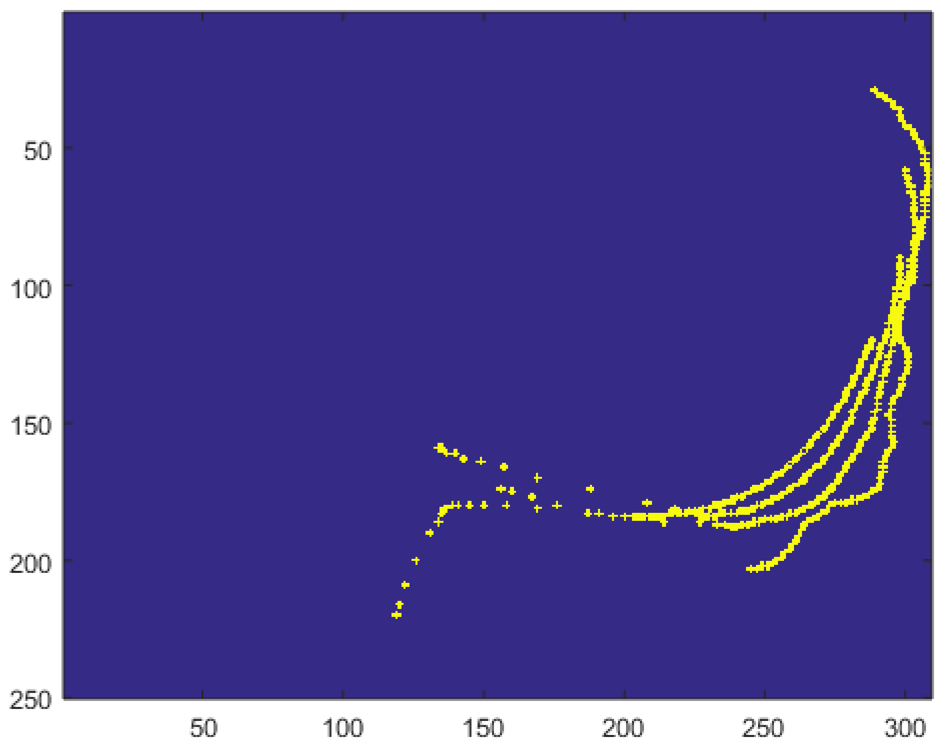


Finally PD indicates the distance that the robot needs to move with the purpose of finding the "$BestDistance$".



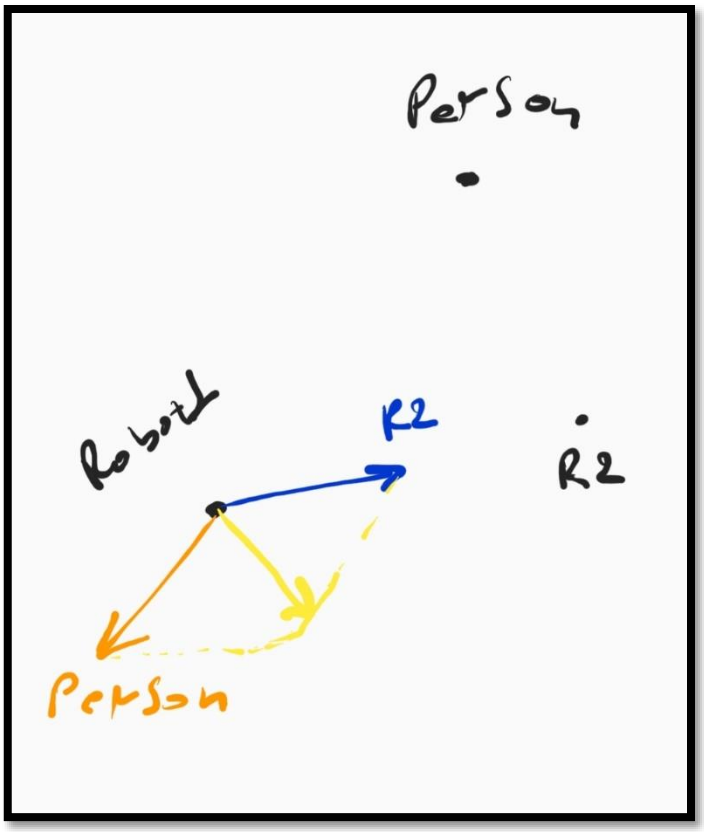
$$u = \frac{PD}{T} = \frac{Dis - BD}{T}$$

There is a more complex algorithm that calculates the actual speed of the robot based on previous speed and maximum acceleration and can be found in the actual code in MATLAB.



Relation Between Robots and People

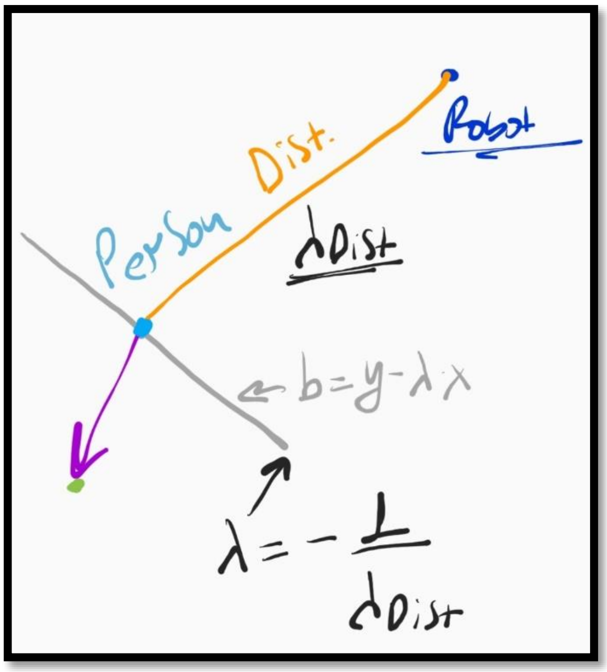
Every robot needs to make a parallel movement that satisfies both the following of a robot and the avoidance of people. More specifically, if a robot detects a person, it creates a force in the opposite direction disproportionate to the distance.



The direction of the Person Force is calculated with the unit vector of person vector $\boldsymbol{p_1} = \frac{\boldsymbol{p}}{|\boldsymbol{P}|}$, which is multiplied by $(\,CorrectDistance - Distance\,)$. However, again this result is restricted by the maximum acceleration of the robot.

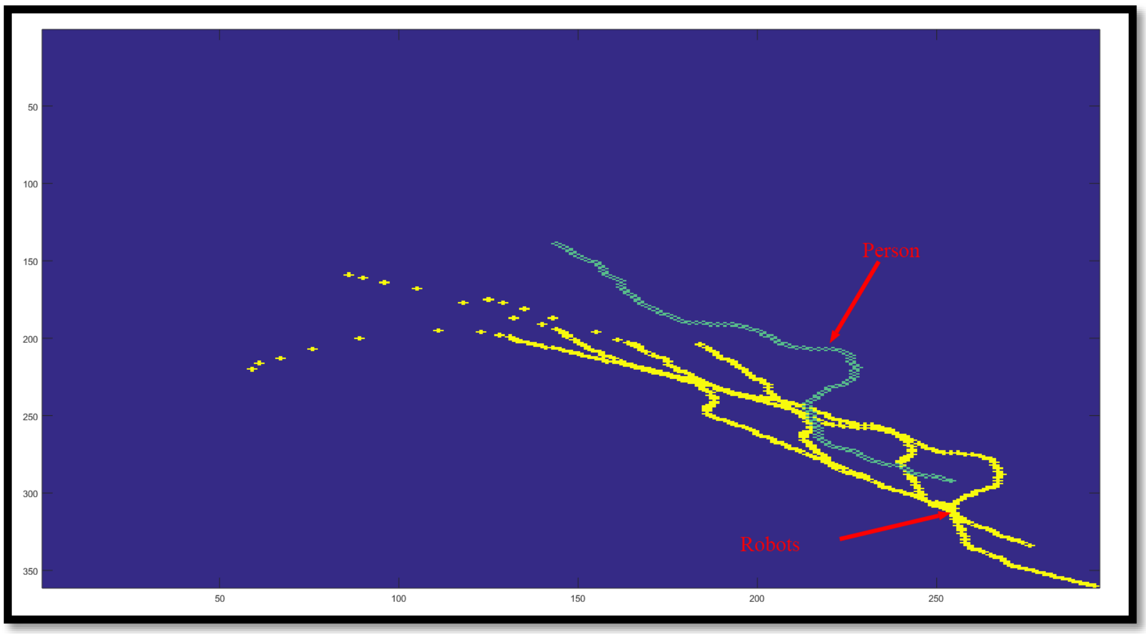
With this analysis, everything works really well and in general as is expected to be. More specifically, the robots follow each other and at the same time avoid the people inside the two dimensional space. However, there is a small detail that was overlooked. If a robot avoid a person, then the robot locks with the person and start to mirror its movements.

For that reason, an analysis has been made for an equation that can identify whether the person moves towards the robot or in the exact opposite direction.

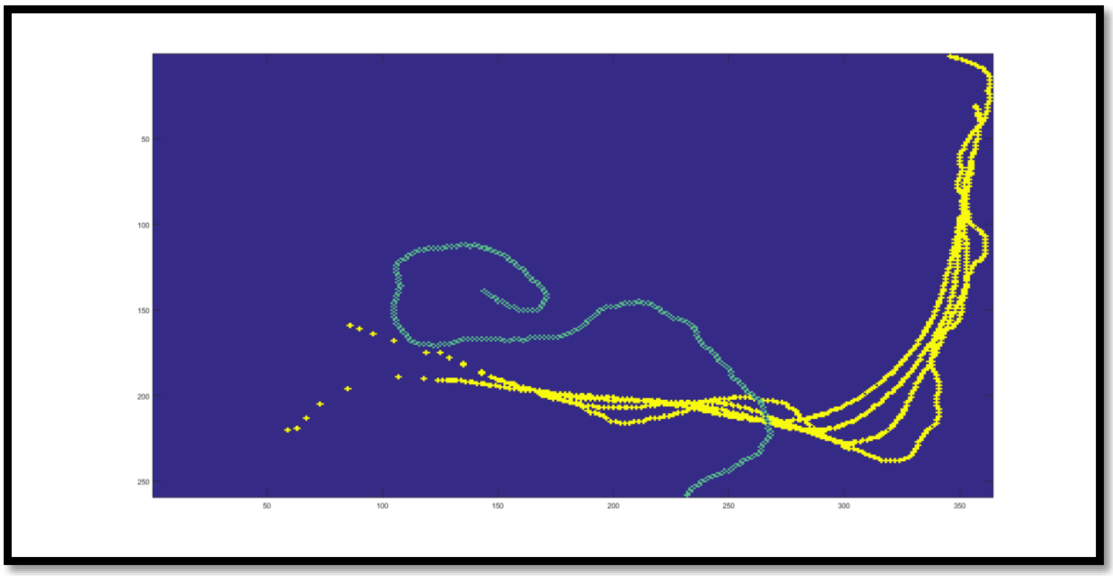


For this analysis we used a line perpendicular to the line that is created between robot and person. If the movement of the person is on the same side as the robot, we need to avoid the person. In any other case, the person is not being taken into consideration.

Simulation Results



In the above example, the background is in yellow color, blue for the robots and light blue for the people. The robots in the above example try to follow one another and at the same time avoid the person.



In this example, the robots are in a chain and the person does not influence even the last robot, because he does not move towards the robot but in the opposite direction.

Conclusion and Future research

An autonomous robot chain algorithm has been analyzed in this report. The proposed method was based on classical vector analysis with the purpose of achieving a really fast processing algorithm that can be used in every processing unit. This project had interesting final results however there are several improvements that can be made in the future:

- Improvement of this algorithm for use in a more realistic environment that contains different shapes and properties of obstacles.

- More realistic representation of the problem in a three Dimensional simulation.
- Actual development of this algorithm with real robots and moving obstacles.

References

[1] Platooning control of autonomous nonholonomic mobile robots in a human-robot coexisting environment, Marco D., Matteo R., Tullio F. and Antonella F., 2012, American Control Conference

[2] Platooning strategy of mobile robot: simulation and experiment, K. Baarath, Muhammad Aizzat Zakaria,*, M.V. Suparmaniam, and Mohd Yazid bin Abu, 2016, The 2nd International Conference on Automotive Innovation and Green Vehicle

[3] Mast, M., et al., Design of the Human-Robot Interaction for a Semi-Autonomous Service Robot to Assist Elderly People, in Ambient Assisted Living: 7. AAL-Kongress 2014 Berlin, Germany, January 21-22, 2014, R. Wichert and H. Klausing, Editors. 2015, Springer International Publishing: Cham. p. 15-29.

[4] Kelly, A. and N. Seegmiller, A Vector Algebra Formulation of Mobile Robot Velocity Kinematics, in Field and Service Robotics: Results of the 8th International Conference, K. Yoshida and S. Tadokoro, Editors, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 613-627,(2014)