AS5570 Project Report

Topic: Modeling and Controlling a Robotic Convoy using Guidance Laws Strategies,

IEEE Transactions On Systems, Man, And Cybernetics, August 2005

Authors: Fethi Belkhouche, Member, IEEE, and Boumediene Belkhouche, Member, IEEE

Guide: Prof. Satadal Ghosh

Submitted by :
Shreeniwas Mahesh Jagdale (ME17B115)
S Tarun Prasad (ME17B114)
Sarin Jacob John (CH17B069)
Vaibhav Shivaji Tarphe(ME17B116)

Problem Statement

- To simulate the results for the solutions proposed in the given paper to solve the research problem
- To suggest improvisations to improve the guidance schemes

Abstract of the problem

This paper deals with the problem of modeling and controlling a robotic convoy given the initial alignment of the convoy with periodic inputs given to the leader robot. Guidance laws techniques are used to provide a mathematical formulation of the problem. The guidance laws proposed in the paper for this purpose are **velocity pursuit**, **deviated pursuit**, and **proportional navigation**. The guidance law equations model the robot's path under velocity sensor based control laws. A systematic study of the tracking problem based on this technique is undertaken. These guidance laws are applied to derive decentralized control laws for the angular and linear velocities. For the angular velocity, the control law is directly derived from the guidance laws after considering the relative kinematics equations between successive robots. The second control law maintains the distance between successive robots constant by controlling the linear velocity. This control law is derived by considering the kinematics equations between successive robots under the considered guidance law. Properties of the method are discussed and proven. Simulation results confirm the validity of our approach, as well as the validity of the properties of the method.

Approach to Simulations:

The paper arrives at equations for maintaining the robotic convoy intact such that the relative velocity along the line joining two consecutive robots is zero and such that each follower robot turns with the same angular velocity as its leader robot about the follower robot. This is however an instantaneous law. For the purpose of simulating this numerically for different shaped trajectories we sample a discrete amount of points with small time intervals between them for each trajectory. For each of these points we compute each robot's velocities using the guidance laws and displace them in the next time step with the computed velocities.

Simulations of the solutions proposed in the paper:

- Assumptions in simulations:
    - Constant distance is maintained within each consecutive member of the convoy due to the inputs from guidance laws
    - The time interval of discretisation is small enough to assume generality of the analytical guidance solutions to the discrete interval simulations.
    - Each follower robot is able to sense the absolute velocity of its own leader robot without any time lag and using it can compute the leader coordinate accurately by knowledge of time.
    - The follower robots can keep track of its own velocities and thereby its coordinates as well.

Velocity Pursuit (Pure Pursuit) Guidance Law

- Equations used in the simulation
    - The equations shown below are the result of the assumptions mentioned above and the various preliminary equations of Pure Pursuit:

$$\dot{x}_{i+1} = \frac{v_{i+1}}{d_{i0}}(x_i - x_{i+1})$$
$$\dot{y}_{i+1} = \frac{v_{i+1}}{d_{i0}}(y_i - y_{i+1})$$
$$\dot{\theta}_{i+1} = \dot{\sigma}_{i,i+1}.$$

- After getting the above equations, we have simulated the results in MATLAB (All MATLAB codes can be found here: MATLAB Codes)
- Results of the simulations
    - For circular leader trajectory: Circular Trajectory Video

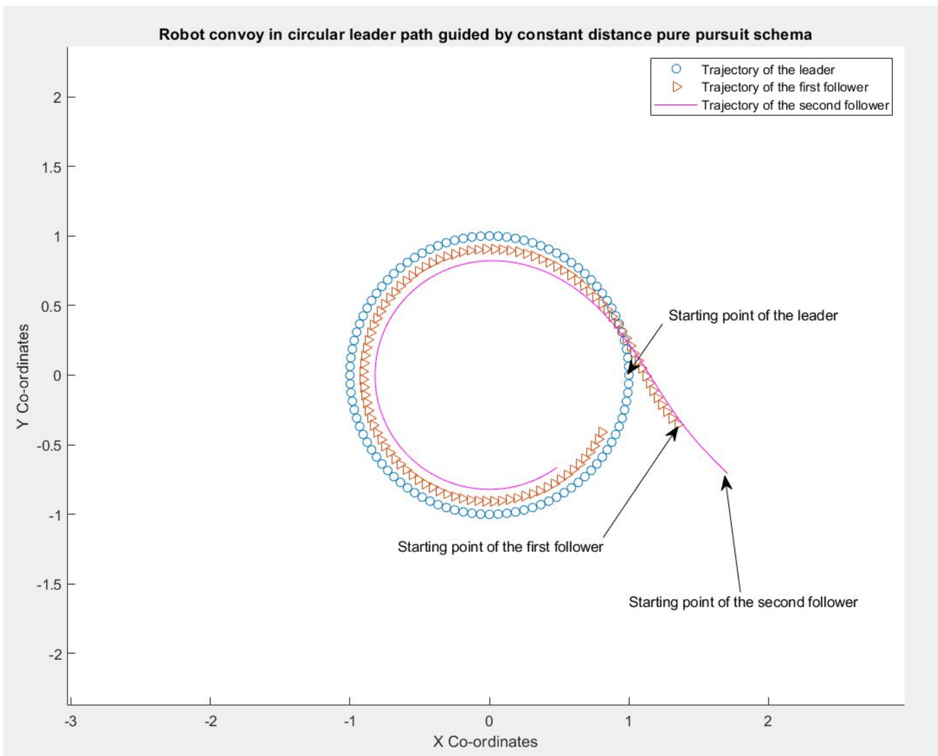


Fig.1: Simulation of PP for circular leader path with 2 followers in the convoy

- For sinusoidal leader trajectory: [Sinusoidal Trajectory Video]

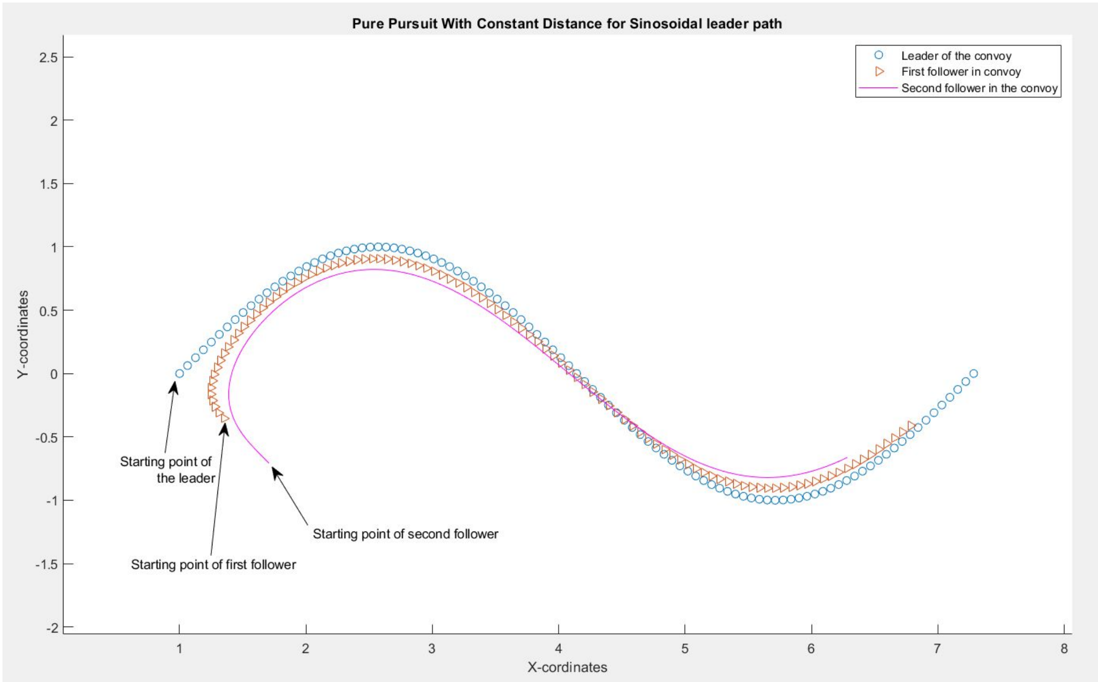


Fig.2: Simulation of PP for sinusoidal leader path with 2 followers in the convoy

- For spiral leader path: [Spiral Trajectory Video]

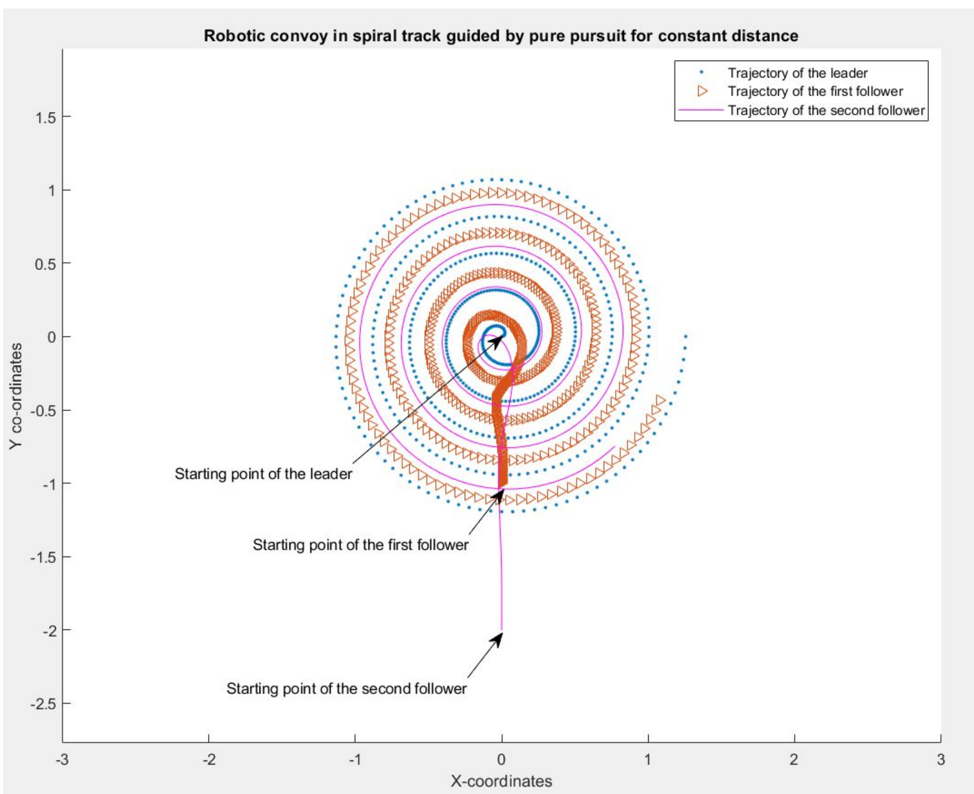


Fig.3: Simulation of PP for spiral leader path with 2 followers in the convoy

- Observations from simulations:
    - The follower doesn’t follow the exact path of the leader, there is a slight error in their trajectories in curved parts of the trajectories
    - The path of the follower matches the trajectory of the leader when the portion of the leader’s trajectory is linear, e.g see in sinusoidal case, the path is almost followed in linear regions
    - If the initial trajectory constraints are not followed then there might be initial chaos between the trajectories
    - Initial parameters must be fed carefully to the algorithm

- Comparison of the simulations with the results shown in the paper:
    - The trajectories simulated by us exactly match the results of the given paper.
    - Pure pursuit is likely to cause offset errors in curved trajectories
    - Following figures of simulated trajectories in the paper clearly tells that our solution matches with the solution of the author
    - Fig.5 shows that in sinusoidal trajectories there is very less offset error in the linear region while in fig.4 you can see a decent amount of offset error. Which are the exact results of the simulations provided by us

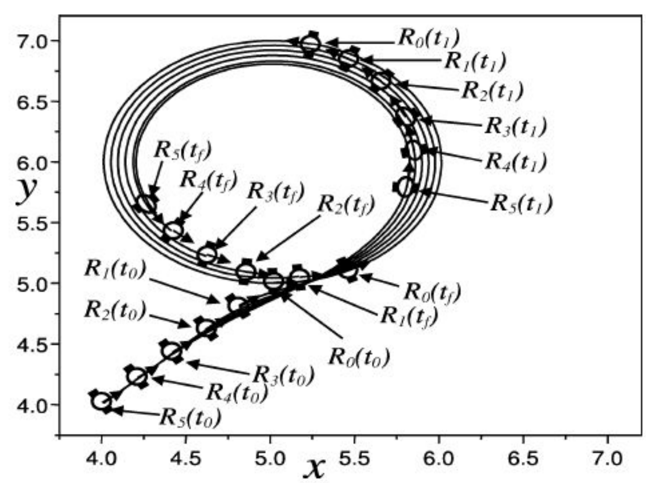


Fig. 5. Path traveled by the convoy, lead robot moving in a circle. The following robots are moving using the velocity pursuit.

Fig.4: Showing simulation results of author for circular trajectories of the leader(Image is taken from the given paper)

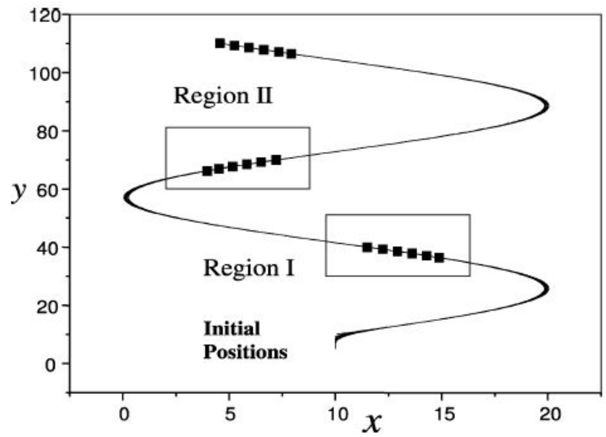


Fig. 8. Robotic convoy moving in a sinusoidal motion. The following robots move according to the velocity pursuit.

Fig.5: Showing simulation results of author for sinusoidal trajectories of the leader(Image is taken from the given paper)

Deviated Pursuit Guidance Law:

- Equations used in the simulation
  - The equations shown below are the result of the assumptions mentioned above and the various preliminary equations of Deviated Pure Pursuit:

$$\dot{r}_{i,i+1} = v_i \cos\delta_i - v_{i+1}\cos\alpha_{i+1}$$
$$r_{i,i+1}\dot{\sigma}_{i,i+1} = v_i \sin\delta_i - v_{i+1}\sin\alpha_{i+1}.$$

$$\dot{x}_{i+1} = v_{i+1}\cos(\alpha_{i+1} + \sigma_{i,i+1})$$
$$\dot{y}_{i+1} = v_{i+1}\sin(\alpha_{i+1} + \sigma_{i,i+1})$$
$$\dot{\theta}_{i+1} = \omega_{i+1} = \dot{\sigma}_{i,i+1}.$$

- The simulations were performed on Python and the link to the code folder can be found here.
- The simulations were generated as videos and the link to the folder containing them can be found here.
- Simulation Outputs were observed as follows:

- Circular Trajectory:

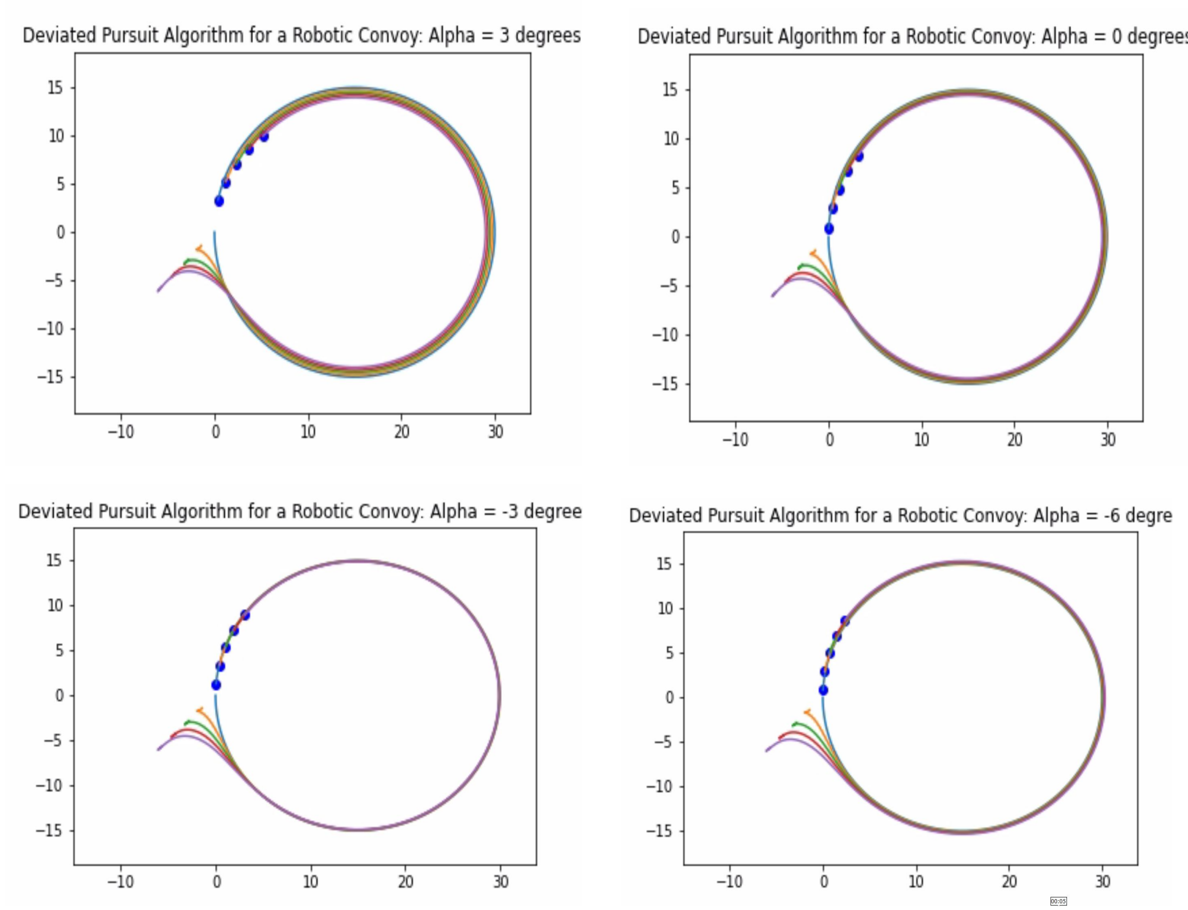


Fig.6: Simulation of DPP for circular leader path for different values of deviation alpha

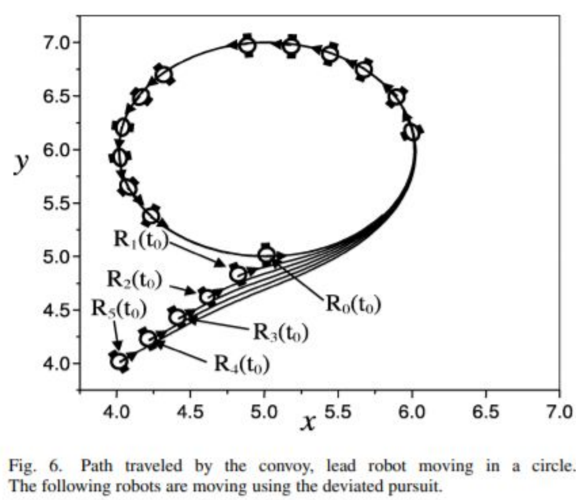


Fig.7:Result obtained in the paper for deviated pursuit for a circular trajectory.

○ Sine Trajectory:

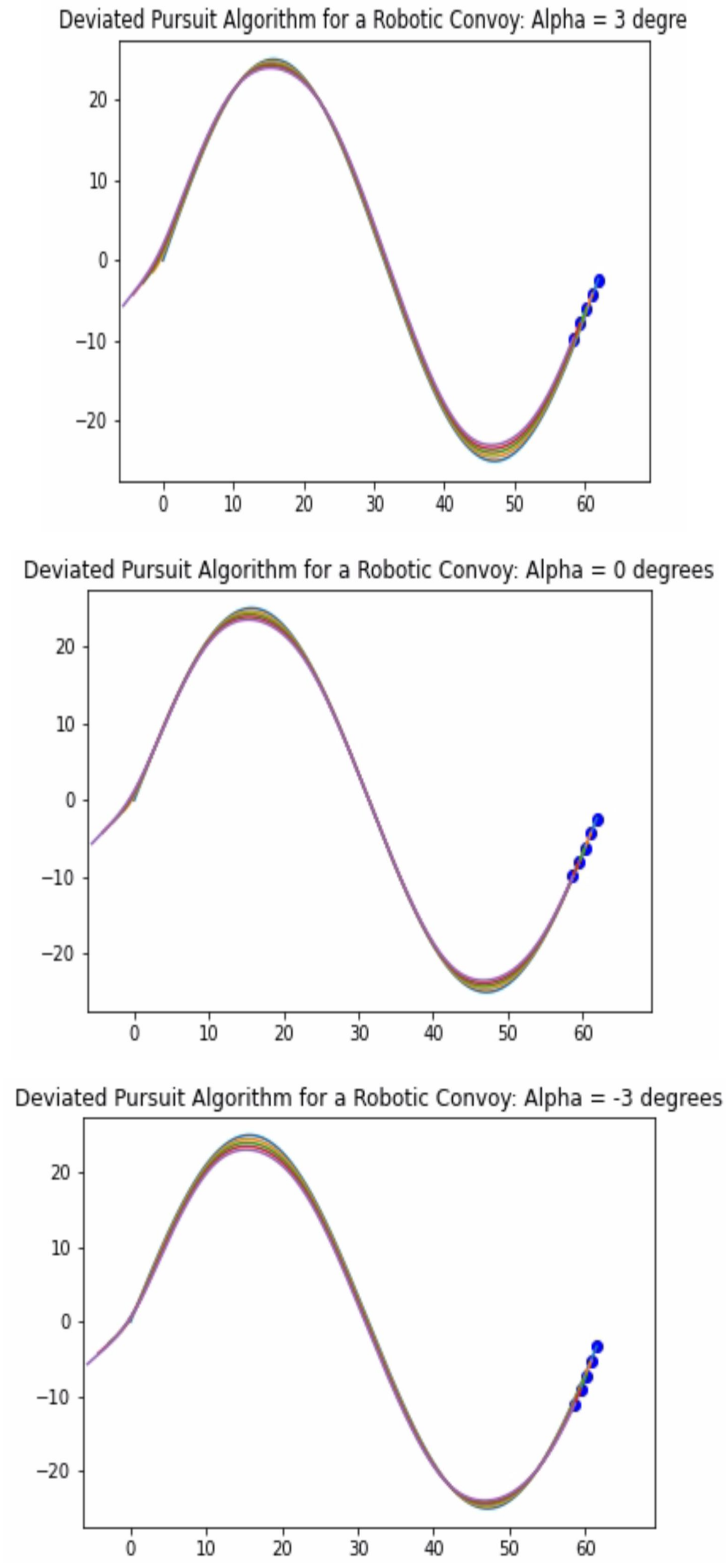


Fig.8: Simulation of DPP for sine leader path for different values of deviation alpha

- Spiral Trajectory:

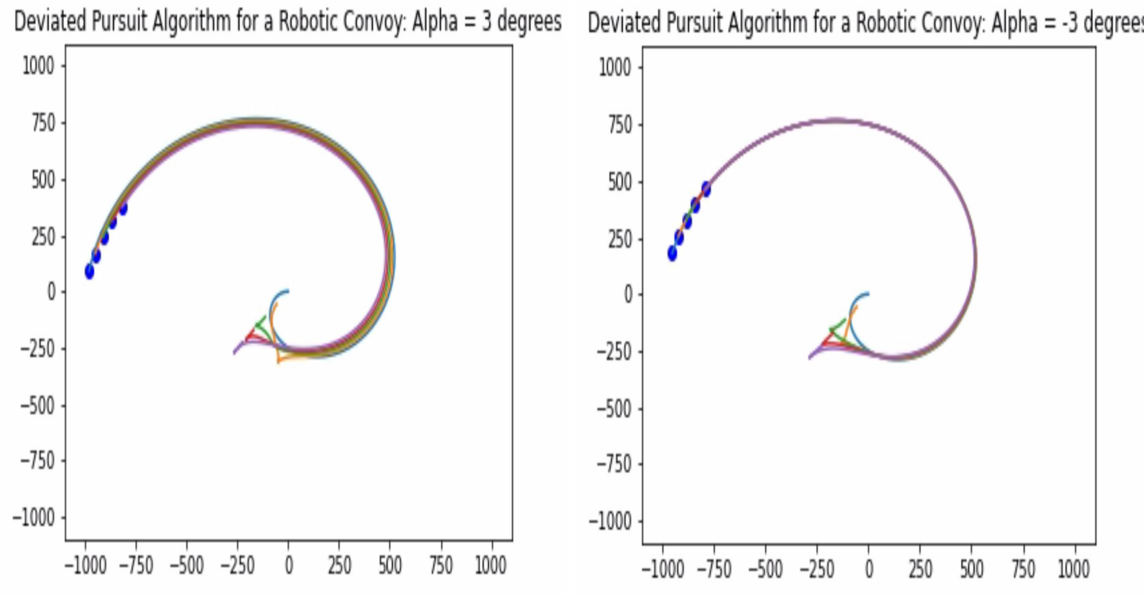
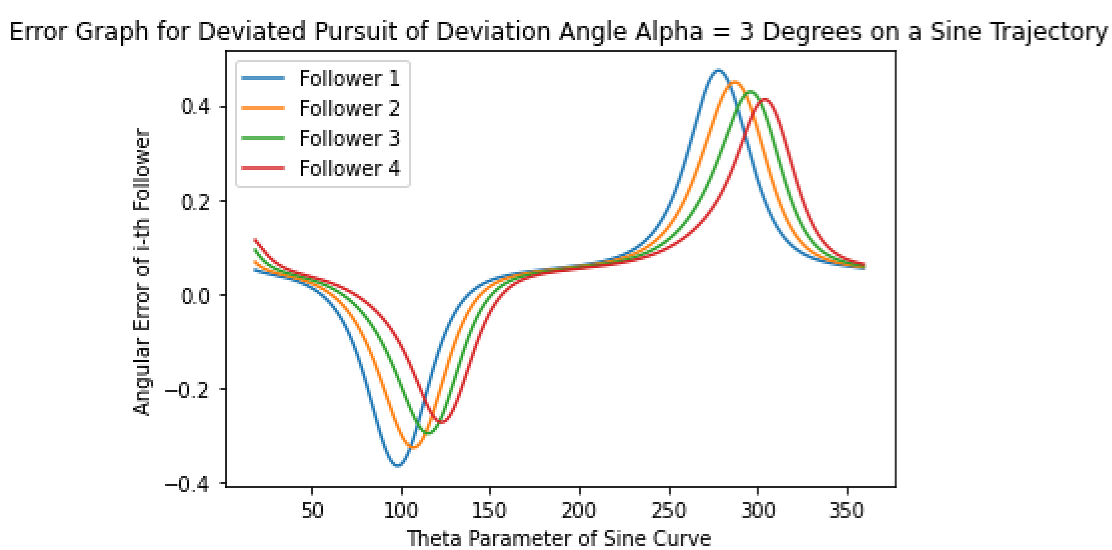
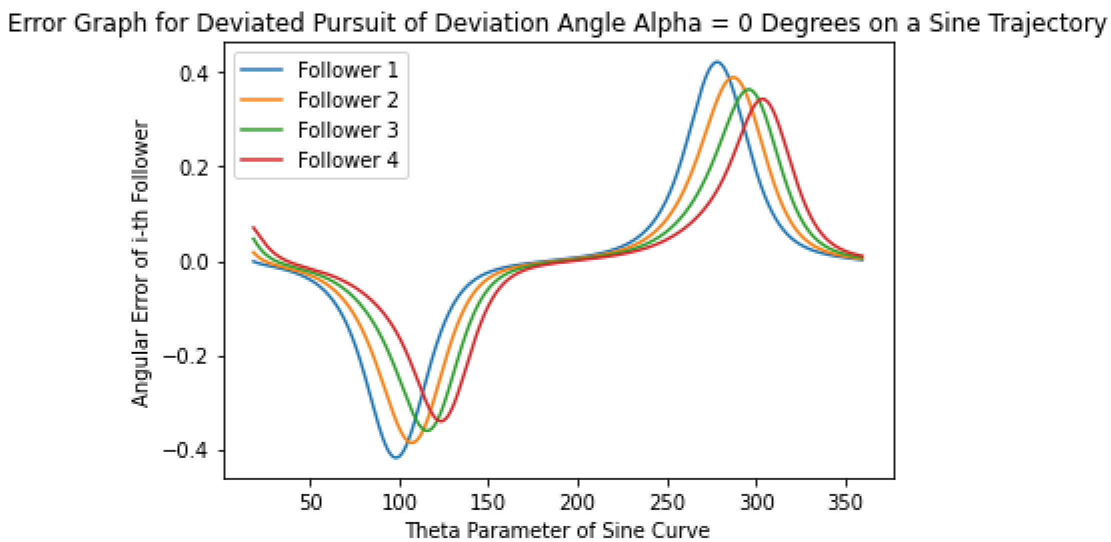
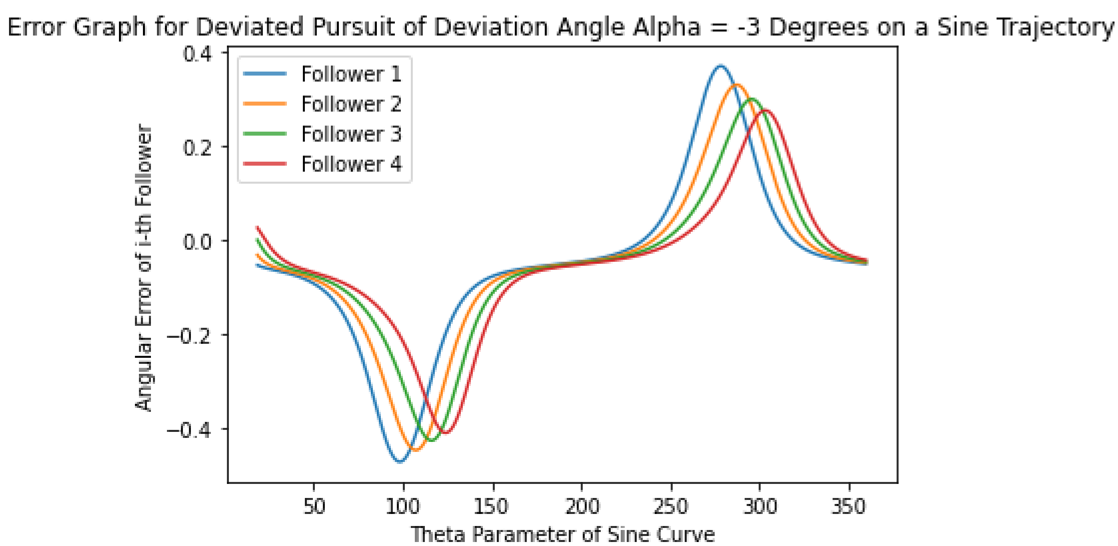


Fig.7: Simulation of DPP for spiral leader for different deviations

- Error Graphs:







- Observations:
  - The results observed in our simulations are exactly as stated in the paper.
  - For the circular case we are able to get rid of the error build up by using an appropriate deviation angle of alpha = -3 degrees in our case. This can be clearly seen in comparison with the results for other values of alphas.
  - For the sine case we can observe the simulation space as the maxima and minima regions and the other approximately linear regions. In the linear regions the performance is nearly the same or a little less when compared to velocity pursuit. When it comes to the maxima and minima regions it can be clearly seen that using one value of alpha improves the convoy following performance in one direction while it degrades the performance in the opposite direction of turning.

- This clearly shows that there is no improvement in the performance over pure pursuit for a sine trajectory and also that a single deviation angle parameter works beneficially only when the trajectory is turning in one specific direction alone and is detrimental in the opposite direction of turning.
- In the spiral case it can be clearly seen that the deviation from leader trajectory can be minimised using appropriate alphas.
- The convoy behaves erratically during the initial part of the spiral trajectory as the scale of the trajectory is much smaller when compared to the convoy scale initially.
- The error graphs clearly indicate the unsymmetric performance of deviation algorithms.

Robot convoy based on the proportional navigation guidance law

- In Proportional Navigation, the pursuer follows the relation:

$$\omega_i = K\dot{\sigma}_{i,i+1}$$

- This derives the kinematic equations for two successive robots as

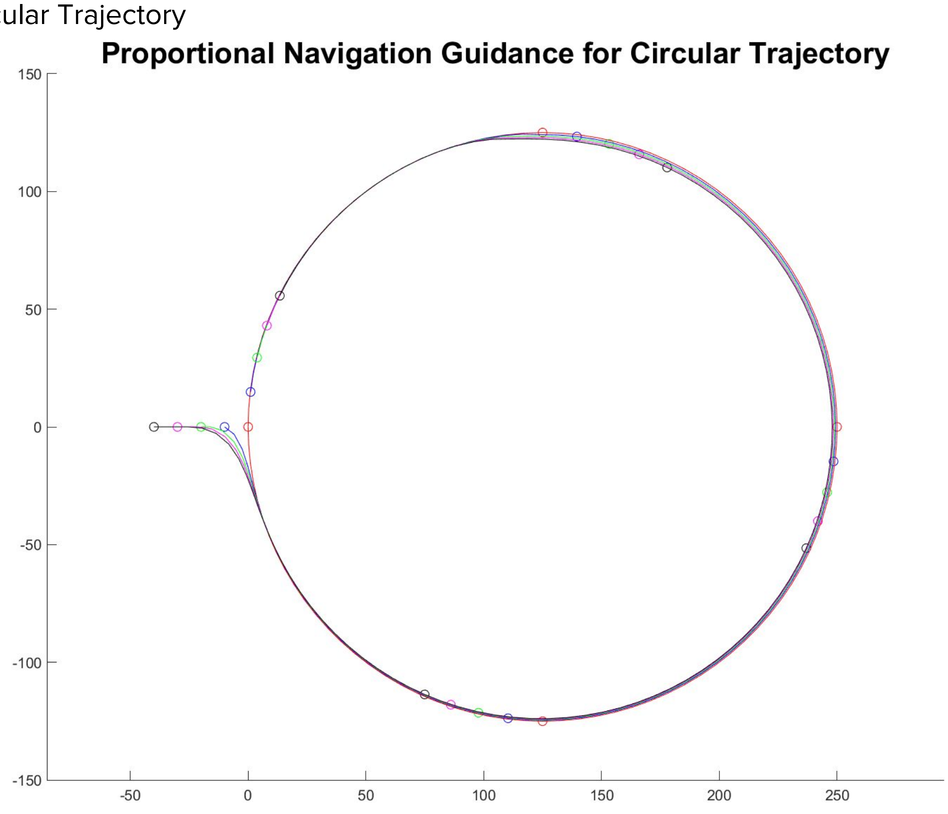
$$\dot{r}_{i,i+1} = v_i \cos \delta_i - v_{i+1} \cos(M\sigma_{i,i+1})$$

$$r_{i,i+1}\dot{\sigma}_{i,i+1} = v_i \sin \delta_i - v_{i+1} \sin(M\sigma_{i,i+1})$$

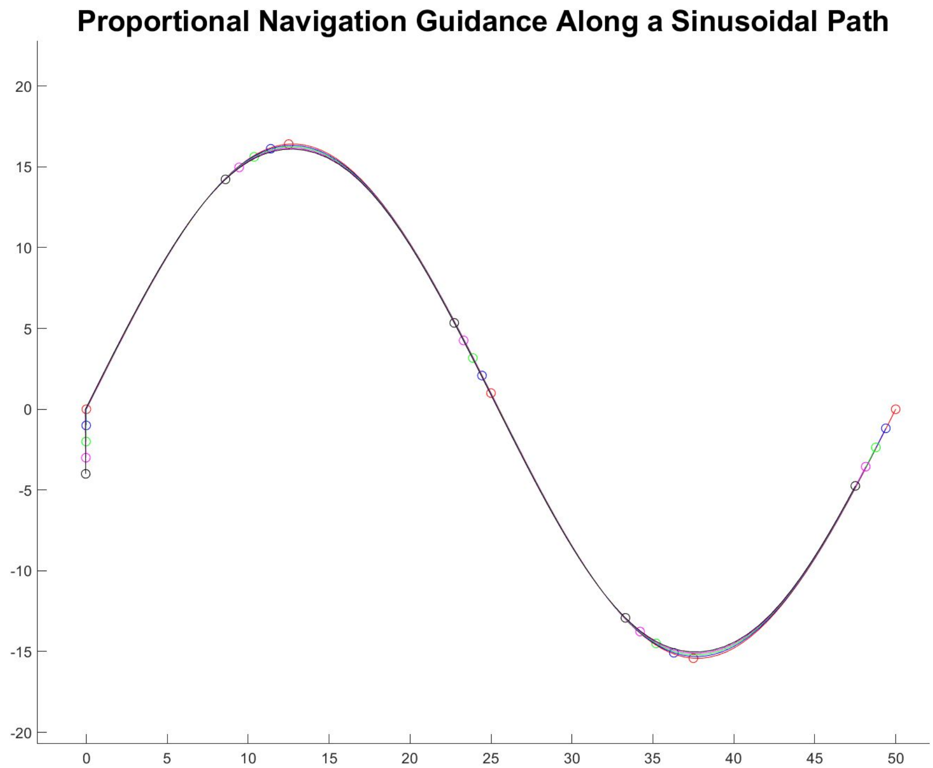
- The constant clearance between two robots leads to the final equation:

$$v_{i+1} = v_i \cos(\theta_i - \sigma_{i,i+1}) / \cos(M\, \sigma_{i,i+1})$$

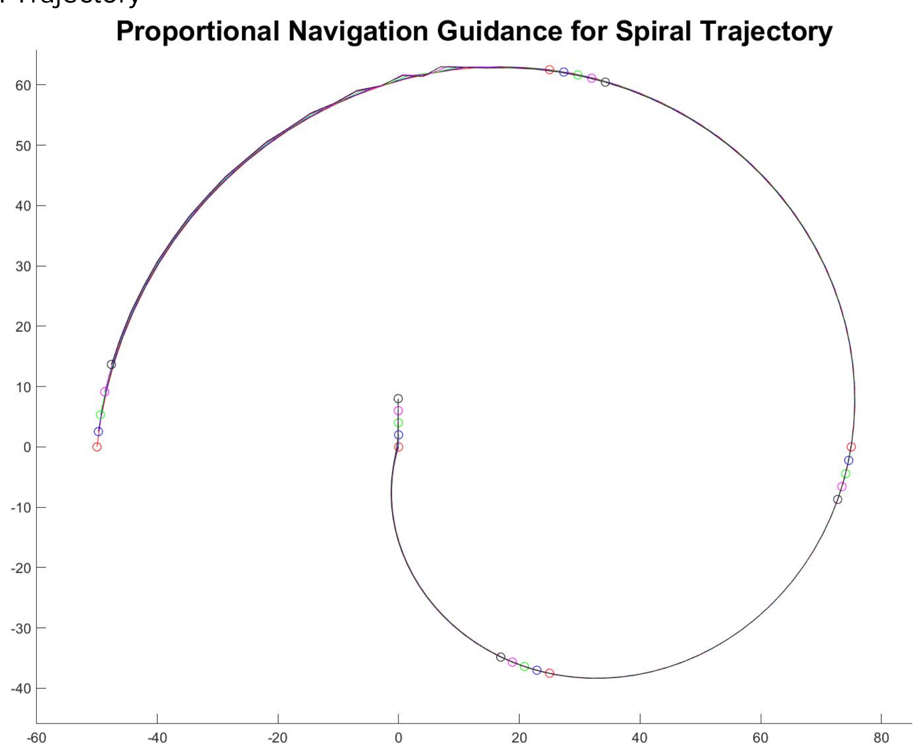
- Results:
  - Circular Trajectory



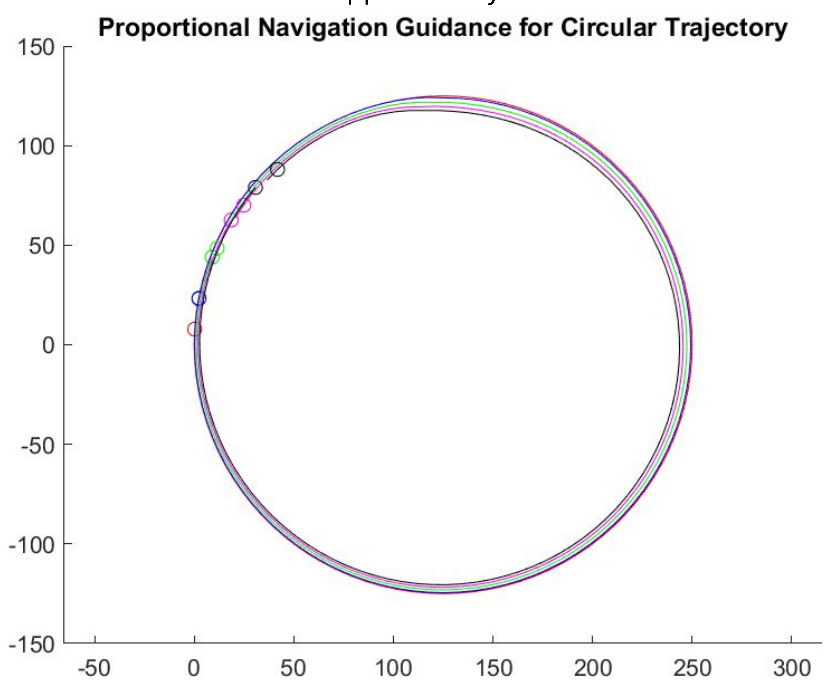
- Sinusoidal Trajectory



- Spiral Trajectory



- **Observations from simulation:**
    - It is noticed that Proportional Navigation is a generalisation of Velocity Pursuit and Deviated Pursuit, given that at K = 1, PN behaves like a PP guidance, and at higher K, it behaves like a DPP with varying deviation angle.
    - The simulation plots can be interpreted as the thinner lines being regions where PN has properly guided the convoy along the same path, whereas thicker regions are where PN hasn't been able to guide the convoy along the same path.
    - Good behaviour of the guidance is heavily dependent on the initial conditions such as clearance, navigation constant K, size of path.
    - There is an erratic behaviour in the circle and spiral simulations at a relative angle of $-\pi$ (top part of the plot), which we believe is due to the sudden shift of angle from positive to negative as output from arctan function.
    - As suggested by the professor, the PN simulation was continued for more revolutions to notice the effect of time on the behaviour of the guidance of convoy robots, and the result of the 4th revolution is shown below. It was expected that the guidance would be better as the robots settled on the path, but the simulation shows it to behave in the opposite way.



Effect of sensor noise on the path of the following robots.
Sensor noise can affect all measured quantities, such as the line of sight angle, or the lead robot's orientation angle or linear velocity. We are assuming that error is due to the measurement of the line of sight angle alone.

$$\sigma_{M01}(t) = \sigma_{01}(t) + \eta(t)$$

Where η (t) is sensor noise.
Assumptions:

1) Sensor Noise is additive.
2) Noise is proportional to the line of sight angle.(η (t)= $\sigma_{01}/3$)

I. Proportional noise used for simulation:

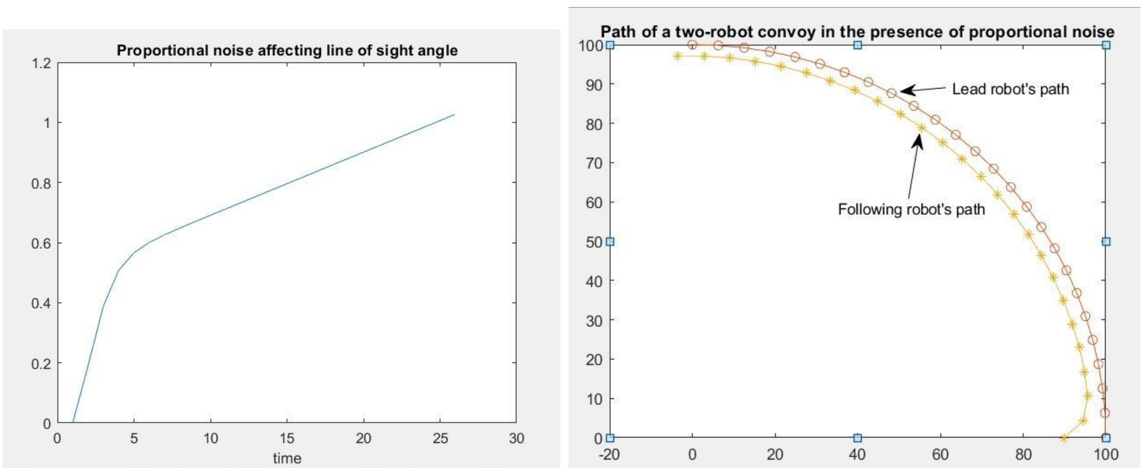


Fig: Two-robot convoy in the presence of proportional noise

Consider a another case where $\eta(t)$ is random function.often,it is taken as white gaussian noise with zero mean and unknown or known variance.
II. Random noise used for simulation:

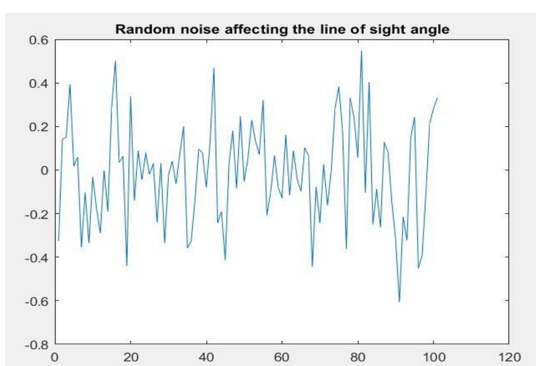


Fig: Random Noise

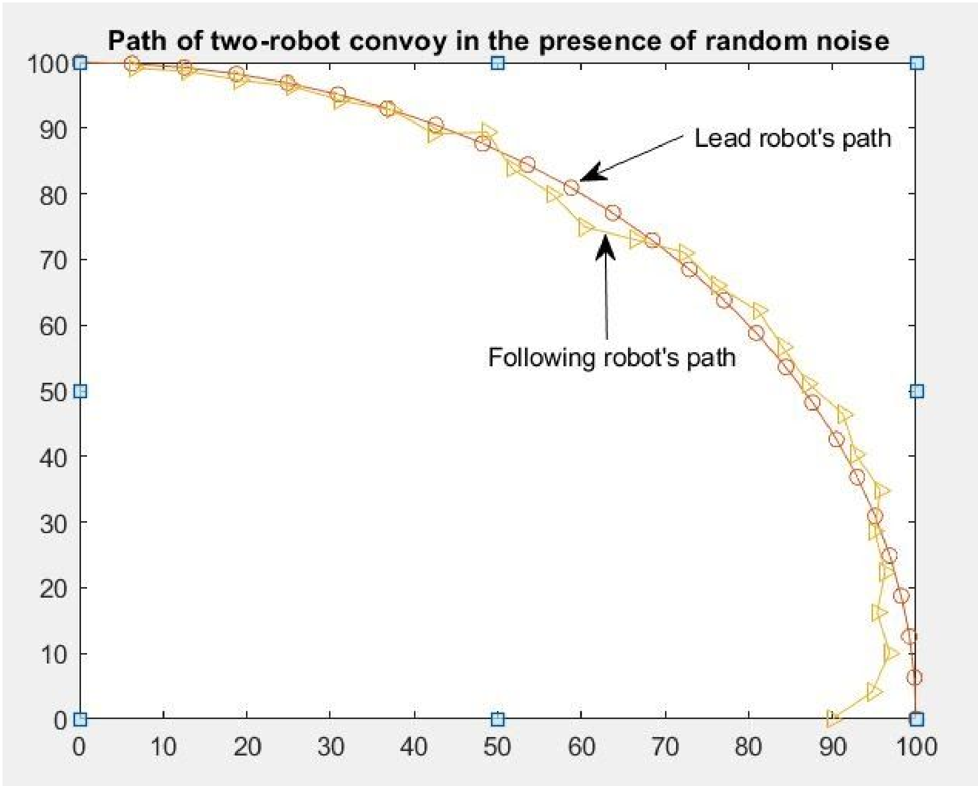


Fig: two-robot convoy in the presence of random noise.

Error in the proportional noise case is quite larger than the random noise. It is roughly ⅓ rd of σ, while in case of error due to the random noise case is bounded and small in magnitude compared to the proportional noise. It is expected that the path-tracking will be more efficient in case of random noise compared to the proportional noise case and the same can be seen in the above simulations.

Improvisations:

- **Deviation Error-Proportional Alpha Deviated Pursuit:**

  One of the issues in the solutions proposed in the paper are that velocity pursuit works fairly well only for straight line type of trajectories while deviated pursuit works well mostly for trajectories which involve a monotonous change of orientation. Moreover a single value of this deviation hyperparameter doesn't suffice for different types of trajectories or even different directions of change of orientation. Proportional navigation tries to address this in some way but it also works only in a monotonous orientation changing trajectory. Thus we need to come up with a new type of deviated pursuit which generates the deviation angle as a function of the deviation from the trajectory it is supposed to follow inorder to handle trajectories with non monotonous change of orientation.

For a pair of follower and leader robots we define the error as the angle made by the current negative line of sight vector with the vector from the current leader coordinate to its previous coordinate. We use a variable deviation angle pursuit algorithm where the deviation angle is computed as the product of a proportionality constant and this error angle. The code for the same can be found here. The videos of the simulations generated can be found here.

The simulations yield much better results. For the circular case it performs as well as deviated pursuit and for the sine trajectory it improves performance both at the maxima and minima in contrast with plain deviated pursuit which improves and degrades the performance respectively. The performance is on par with velocity pursuit in the linear region as well.

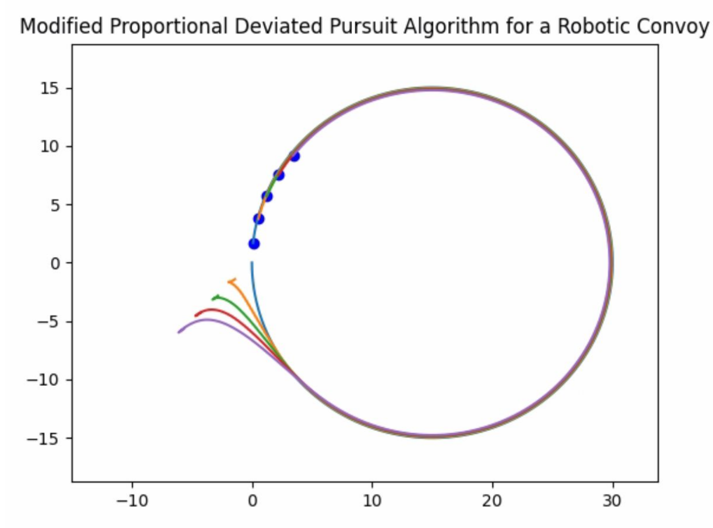


Fig: Modified Proportional Deviated Pursuit for Circle

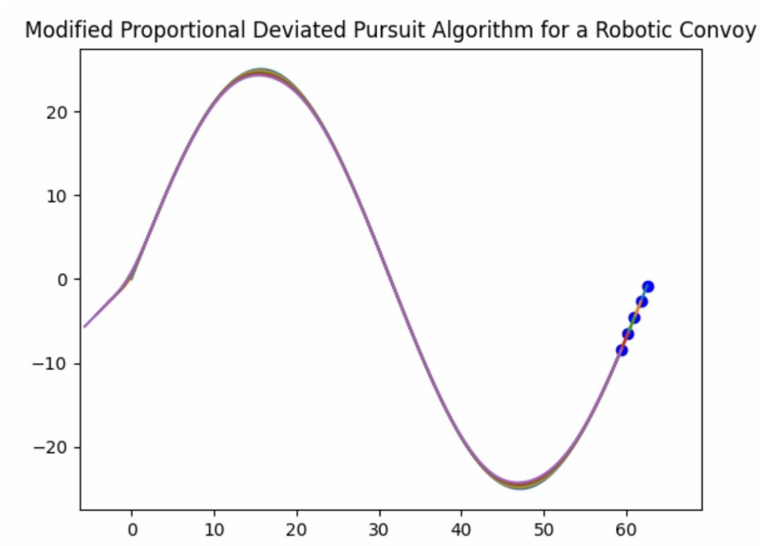
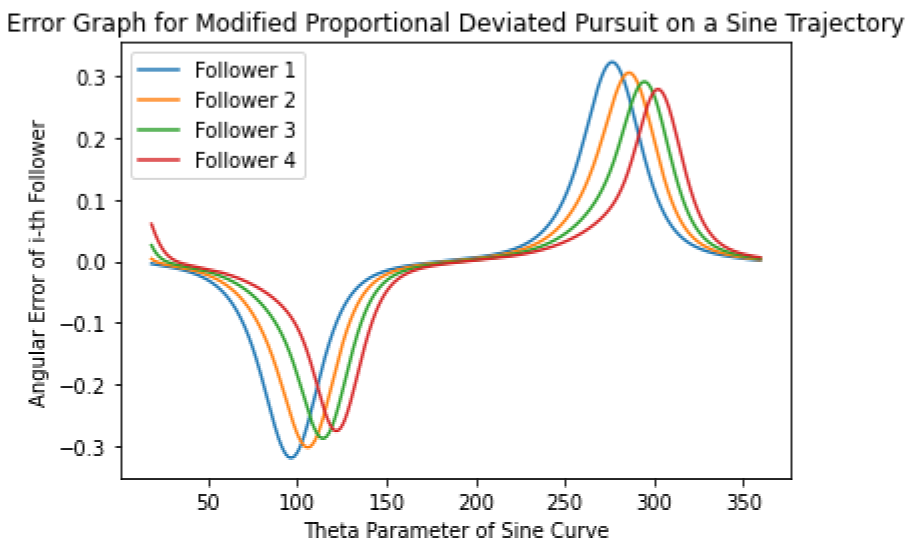
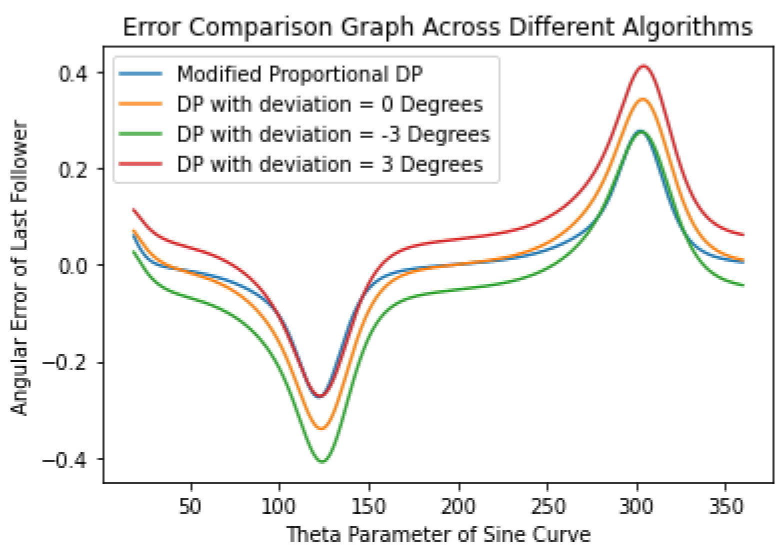


Fig: Modified Proportional Deviated Pursuit for Sine





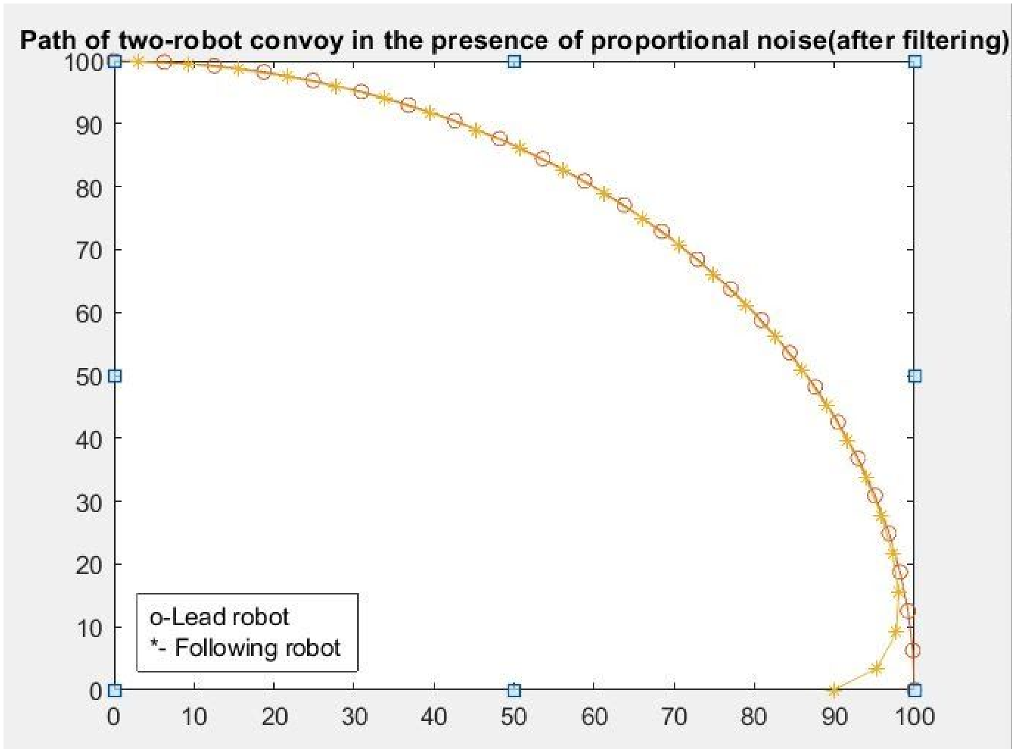
The above two graphs clearly illustrate that this algorithm beats 0 deviation algorithm and superimposes the better performance of the two positive and negative deviation algorithms at maximas and minimas.

- **Filtering**

  We have considered two cases for studying the influence of sensor noise on the path of the following robot. In the first case, $\eta(t)$ is a random function. In the second case , $\eta(t)$ is proportional to the line of sight angle. In both cases,it has been found that the path of the following robot is deviated from the desired path of the lead robot in the presence of noise.this deviation from desired trajectory is dependent on the magnitude of noise. Different techniques such as filtering can be used to improve this aspect.

  In second case,where noise is proportional to the line of sight angle($\eta(t) = \sigma_{01}(t)/3)$,

  An appropriate correction factor can be used to filter the LOS angle.



$$\sigma_{M01}(t) = \sigma_{01}(t) + \eta(t)$$
$$\sigma_{M01}(t) = \sigma_{01}(t) + \sigma_{01}(t)/3$$
$$\sigma_{M01}(t) = 4/3 * \sigma_{01}(t)$$

So,before updating the LOS command to the following robot,it can be filtered by pre-multiplying by correction factor(¾ in this case). And then true LOS command can be fed to the following robot for efficient tracking.

$$\sigma_{CM01}(t) = 3/4 * \sigma_{M01}(t)$$

$\sigma_{CM01}(t)$ = corrected Los angle

Fig: two-convoy robot in the presence of proportional noise.

In the first case,noise is a random function.unlike in the case of proportional noise, we don't know(or can't predict accurately) the magnitude of error at every time step.generally,random noise is assumed to be bounded white gaussian noise.so,we do know the bounds of random noise.
It can be seen that the path of the following robot is fluctuating around the desired path of the lead robot in the presence of random noise. To account for this fluctuation,we have

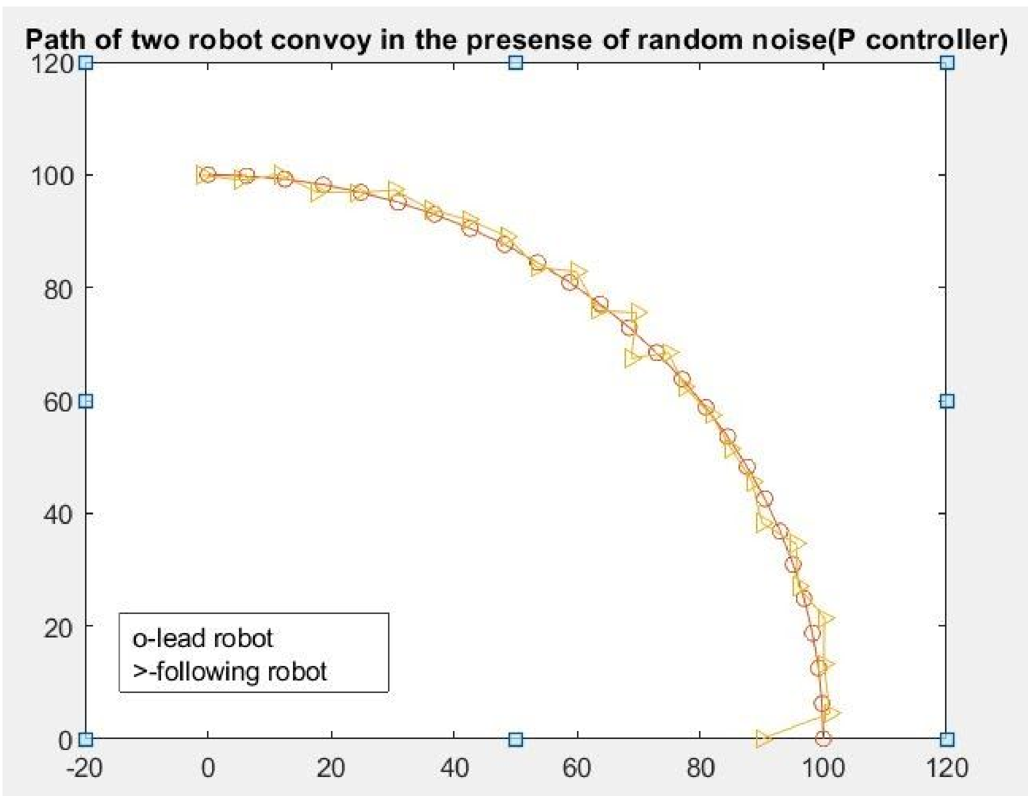
taken error between following robot's position and target lead's robot position in control formulation for efficient tracking.

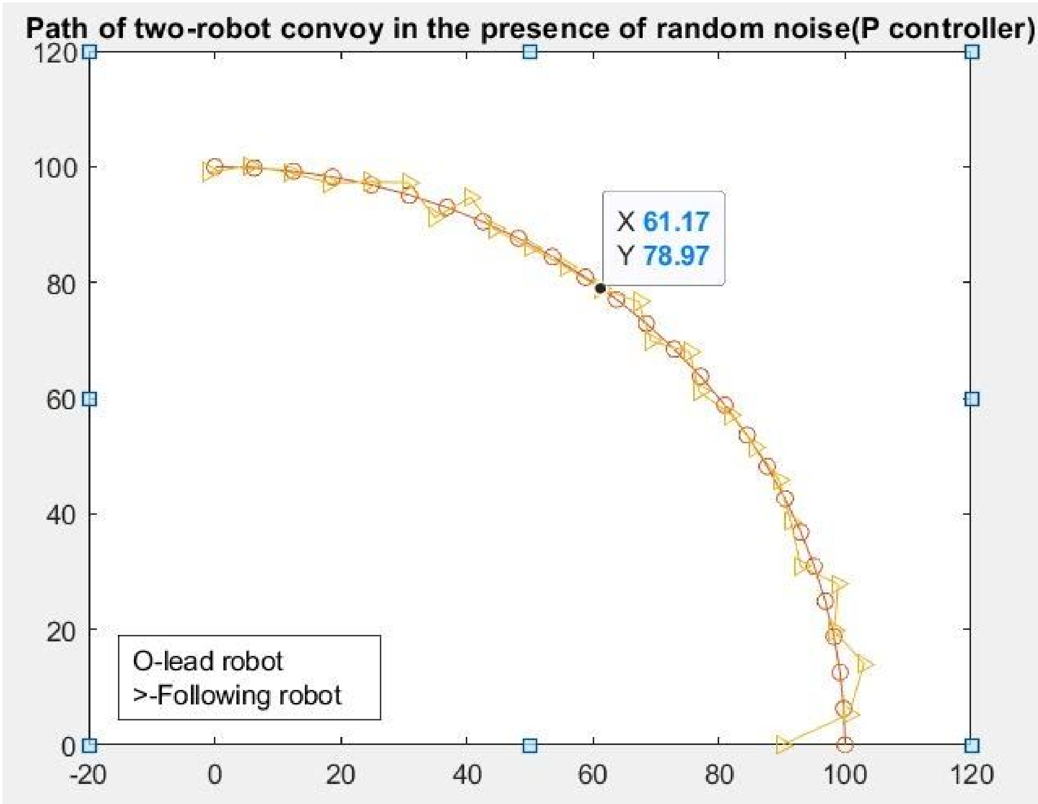
$$xf(i) = xf(i-1) + k * d(i) * cos(\sigma_{i,i+1}(i)) + v * cos(\sigma_{i,i+1}(i)) * dt$$
$$yf(i) = yf(i-1) + k * d(i) * sin(\sigma_{i,i+1}(i)) + v * sin(\sigma_{i,i+1}(i)) * dt$$

d(i) = cross-track error

K has to be chosen with care,otherwise it would shoot up the error and lead to high deviation from desired path.

$$k(i) = (\sigma_{i,i+1}(i) + LB(\eta(t))) / (\sigma_{i,i+1} + UB(\eta(t)))$$





- **CNN based approach to determine alpha(deviated angle) in case of DPP**
    - We have faced difficulties while getting optimum trajectory in case of deviated pursuit guidance scheme i.e we had to do various iterations to get the value of optimum deviated angle which optimizes the final trajectories of the convoy.
    - One more observation was the value of optimum deviated angle differs from trajectory to trajectory, it depends on various other parameters too.
    - So the solution we are trying to provide here will be able to predict the values of deviated angle in advance(In this case we will only take circular trajectories to simplify the expressions)
    - CNNs can be used to predict the value of deviated angle using various parameters, as the value of deviated angle differs for different trajectories and different values of the parameters
    - Parameters to be considered while modeling CNN
        - The radius of the circular trajectory
        - The distance between consecutive robots
        - Speed of the leader
        - Speed of the follower
    - Using the above parameters and from trials, we can train a neural network, which will, in turn, predict the value of the optimum deviated angle for new trajectory and parameters

Applications of knowledge:

- Adaptive cruise control (ACC) is an available cruise control ADAS(advanced driver-assistance system) for road vehicles that automatically adjusts the vehicle speed to maintain a safe distance from vehicles ahead.basically.its problem of tracking the lead car and regulating the speed of following car such that it maintain constant distance between them.control for linear velocity of following has to be considered , based on the engagement geometry and suitable guidance algorithm.
- The concept of connected vehicles is proposed and tested to reduce the traffic intensity on the road. Connected vehicles will move in the same fashion as convoy.lead vehicles will guide all following vehicles such that optimum safe distance and velocities maintain between them.this will reduce the traffic intensity and accident threats.
- The problem of convoy driving can be seen as a special case of group formation. Military applications of convoy driving are the most obvious, where a given number of autonomous vehicles follow each other while keeping a safe constant distance. Other applications can be found, for example, in flexible factories, where an automated robotic convoy is used for product transportation.

Learning Outcomes:

- The robot convoy can be guided through numerous guidance schema
- We have used few from paper and tried to implement some improvisations
- As we go with more sophisticated algorithms, errors reduce but it comes with high computational costs
- This knowledge can be applied in various day to day life activities, such as autonomous cars and automatic signals for cars to follow the convoy.
- We have learnt various new things about guidance schemas and saw their real life application as a part of this project, the experience during the project was enriching and encouraging
- **References:**

    - **https://ieeexplore.ieee.org/abstract/document/1468252**