Modeling and Controlling a Robotic Convoy Using Guidance Laws Strategies

Fethi Belkhouche, Member, IEEE

Boumediene Belkhouche, Member, IEEE

Group Members

- S Tarun Prasad ME17B114
- Shreeniwas Mahesh Jagdale ME17B115
- Vaibhav Shivaji Tarphe ME17B116
- Sarin Jacob John CH17B069

Abstract/Theory of the paper:

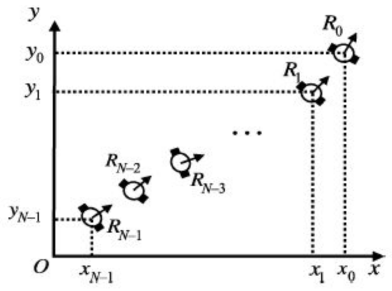


Illustration of the robotic convoy.

- 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.
- 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.

Approach to Simulations:

- Relative Velocity along the line joining 2 consecutive robots is zero.
- Each follower robot turns with the same angular velocity as its leader robot about the follower robot.
- 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.

Simulation Assumptions:

- 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.

Pure Pursuit

- **Guidance Law**:

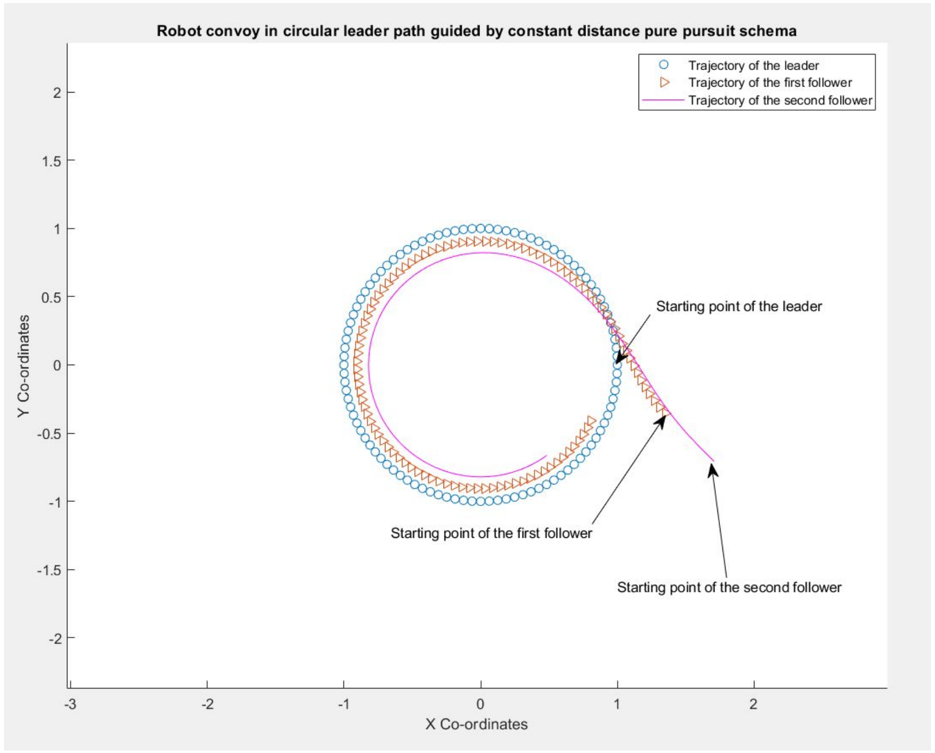
$$\dot{r}_{i,i+1} = v_i \cos\delta_i - v_{i+1}\cos\alpha_{i+1}$$
$$r_{i,i+1}\dot{\sigma}_{i,i+1} = v_i \sin\delta_i - v_{i+1}\sin\alpha_{i+1}.$$

- After simplifying above equations for pure pursuit case we get following equations, which can be used to generate trajectories in MATLAB

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

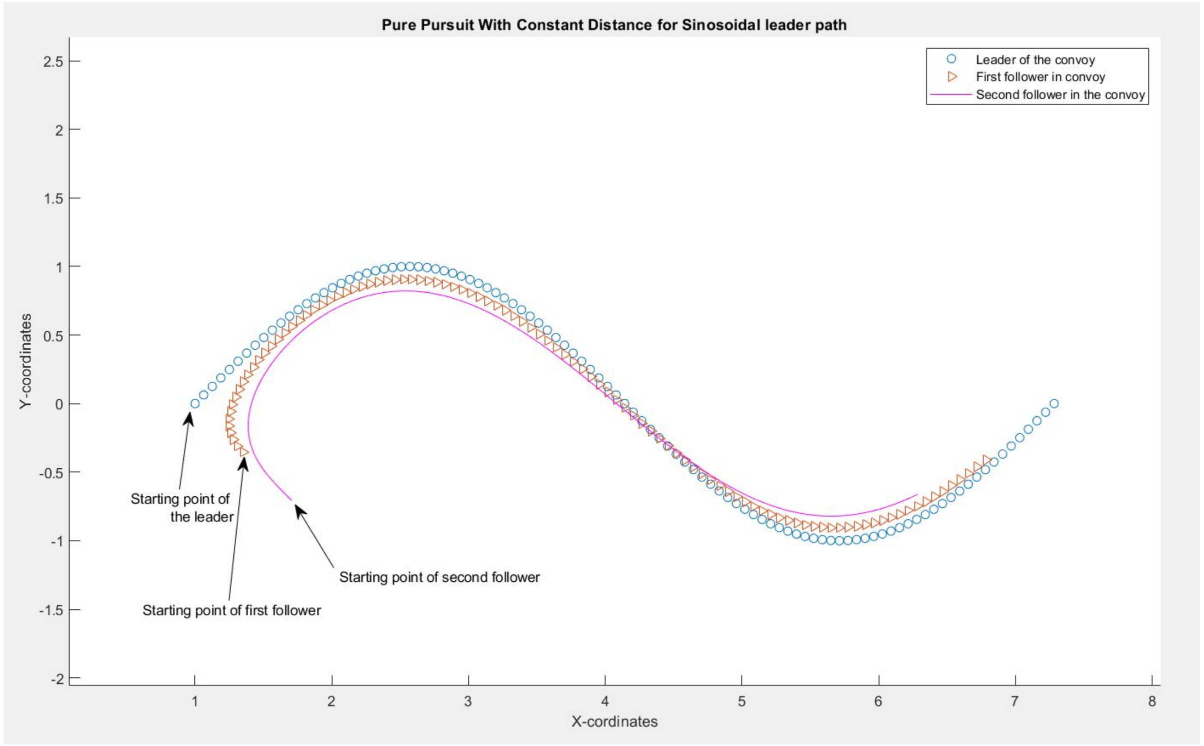
- The simulations were performed on MATLAB for an arbitrary convoy with n=3 robots and a fixed distance between each robot and the link to the code folder can be found here.
- The simulations were generated as videos and the link to the folder containing them can be found here.

Pure Pursuit Circle Simulation



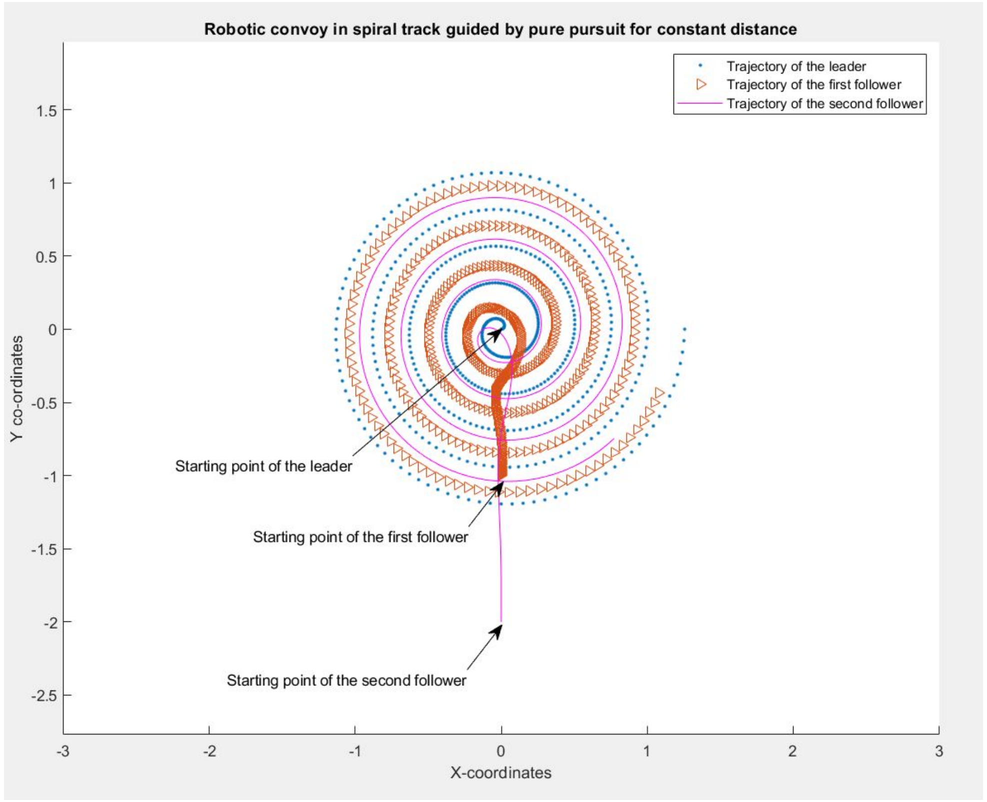
For circular leader trajectory: [Circular Trajectory Video]

Pure Pursuit Simulation for sinusoidal leader trajectory



For sinusoidal leader trajectory: Sinusoidal Trajectory Video

Pure Pursuit Simulation for spiral leader trajectory



For spiral leader path: Spiral Trajectory Video

❖ Pure Pursuit Simulation Observations:

- 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
- The offset error can be reduced by using DPP or advanced improvisations

❖ Pure Pursuit Simulation Comparison with Author's Results:

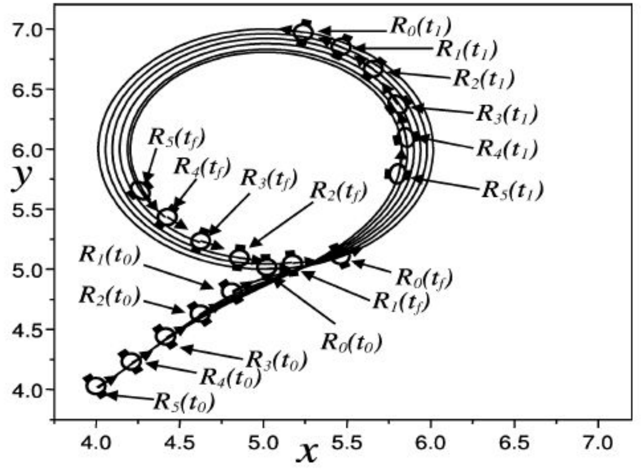
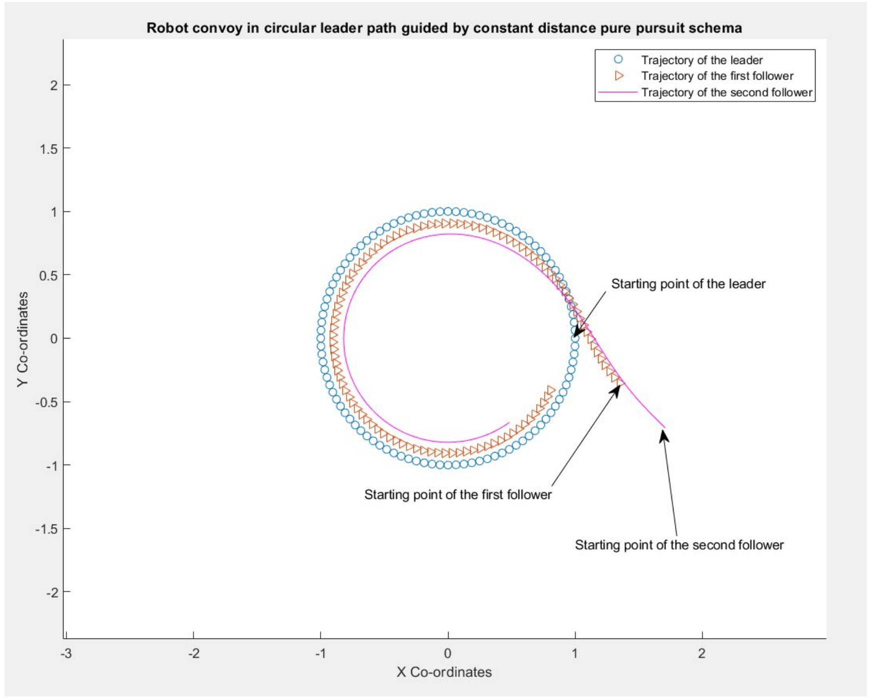


Fig. 5. Path traveled by the convoy, lead robot moving in a circle. The following robots are moving using the velocity pursuit.

Solution provided by the author in the paper



Simulation results of our code

❖ Pure Pursuit Simulation Comparison with Author's Results:

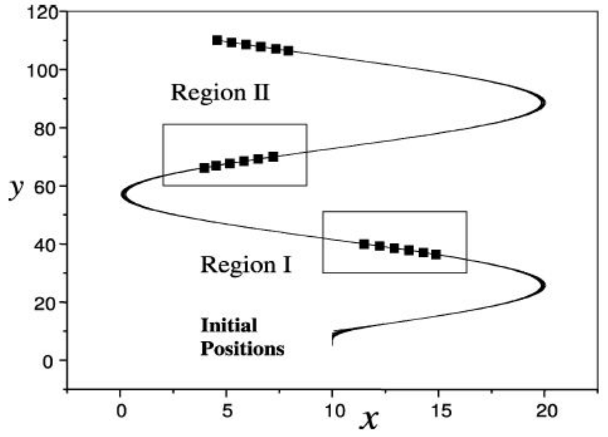
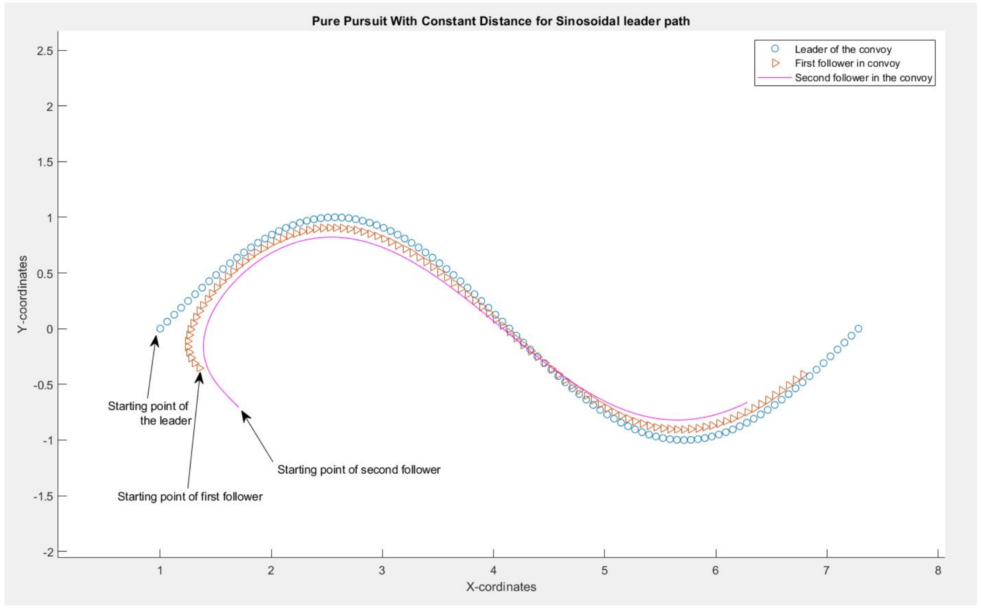


Fig. 8. Robotic convoy moving in a sinusoidal motion. The following robots move according to the velocity pursuit.



Solution provided by the author in the paper

Simulation results of our code

❖ Pure Pursuit Simulation Comparison with Author's Results:

- The trajectories simulated by us exactly match the results of the given paper.
- Pure pursuit is likely to cause offset errors in curved trajectories
- Previous slides clearly shows results of simulated trajectories in the paper and compares it with trajectories simulated by us

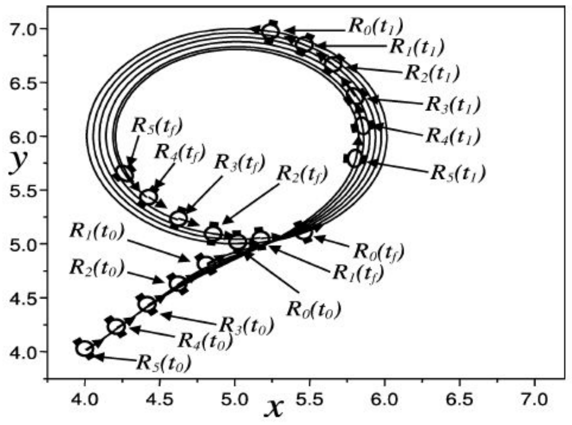


Fig. 5. Path traveled by the convoy, lead robot moving in a circle. The following robots are moving using the velocity pursuit.

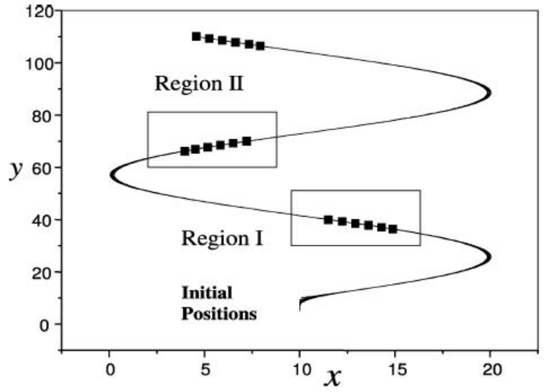


Fig. 8. Robotic convoy moving in a sinusoidal motion. The following robots move according to the velocity pursuit.

Deviated Pursuit:

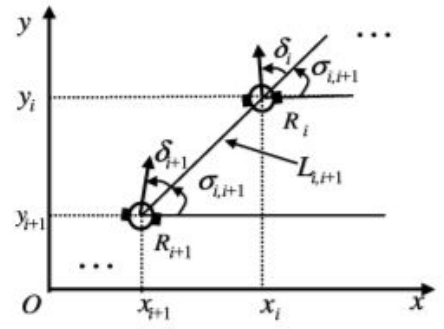


Fig. 3. Geometry of the tracking problem.

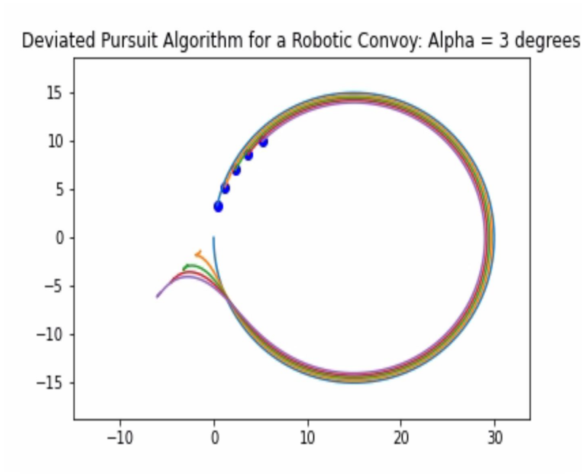
- Guidance Law:

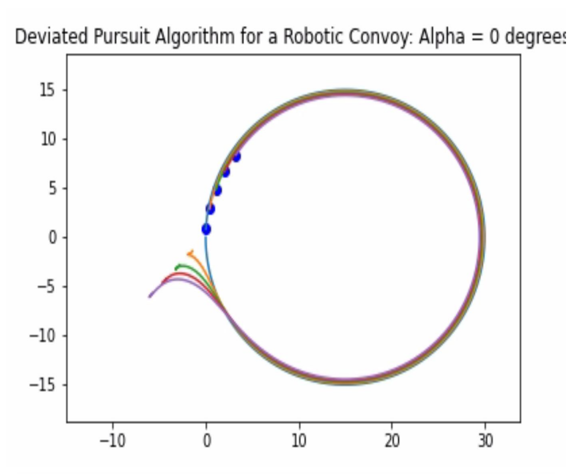
$$\dot{r}_{i,i+1} = v_i \cos\delta_i - v_{i+1}\cos\alpha_{i+1}$$
$$r_{i,i+1}\dot{\sigma}_{i,i+1} = v_i \sin\delta_i - v_{i+1}\sin\alpha_{i+1}.$$

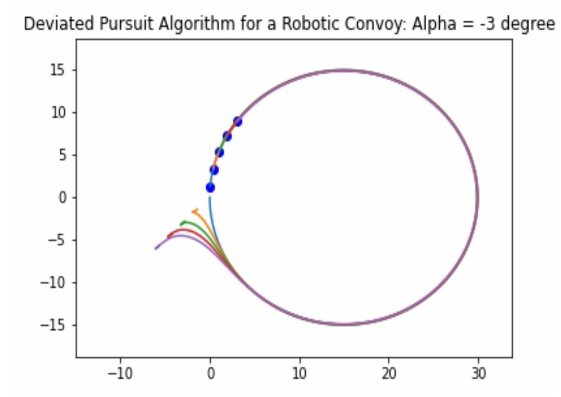
$$\dot{x}_{i+1} = v_{i+1}\cos(\alpha_{i+1} + \sigma_{i,i+1})$$
$$\dot{y}_{i+1} = v_{i+1}\sin(\alpha_{i+1} + \sigma_{i,i+1})$$
$$\dot{\theta}_{i+1} = \omega_{i+1} = \dot{\sigma}_{i,i+1}.$$

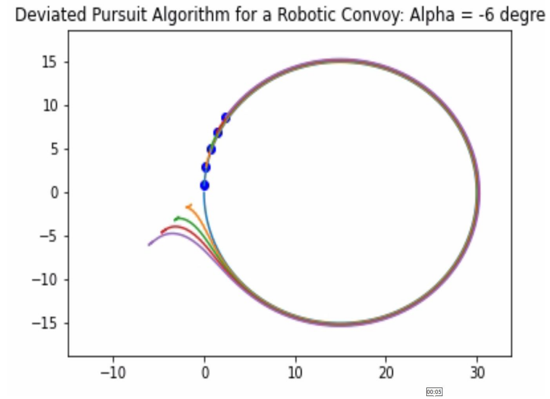
- The simulations were performed on Python for an arbitrary convoy with n robots and a fixed distance between each robot and the link to the code folder can be found here.
- The simulations were generated as videos and the link to the folder containing them can be found here.

Deviated Pursuit Circle Simulation

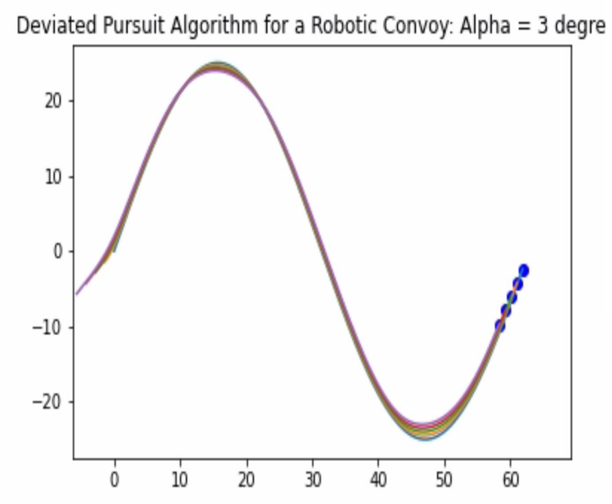


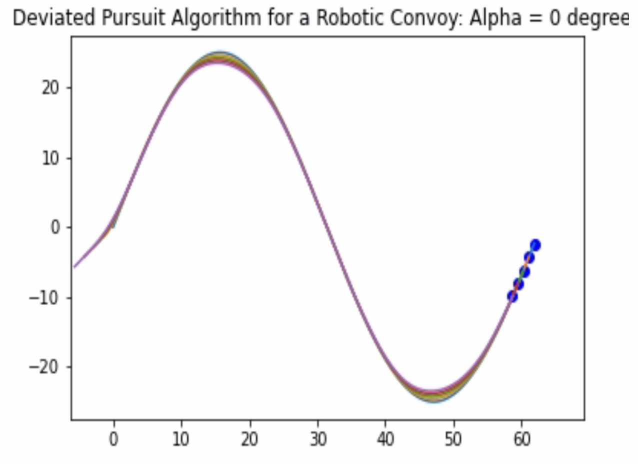


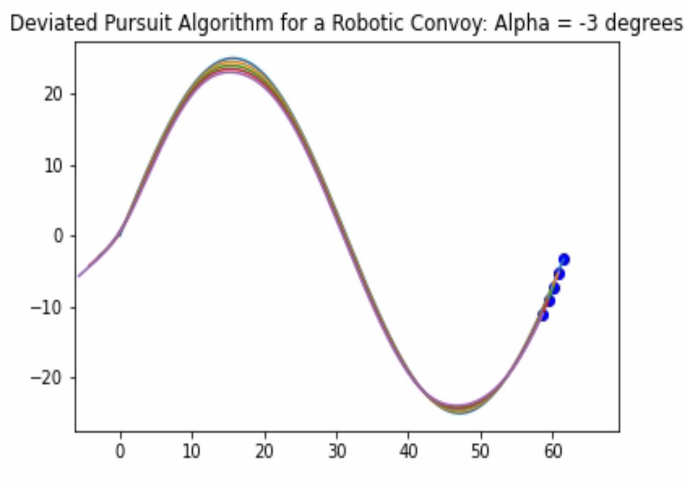




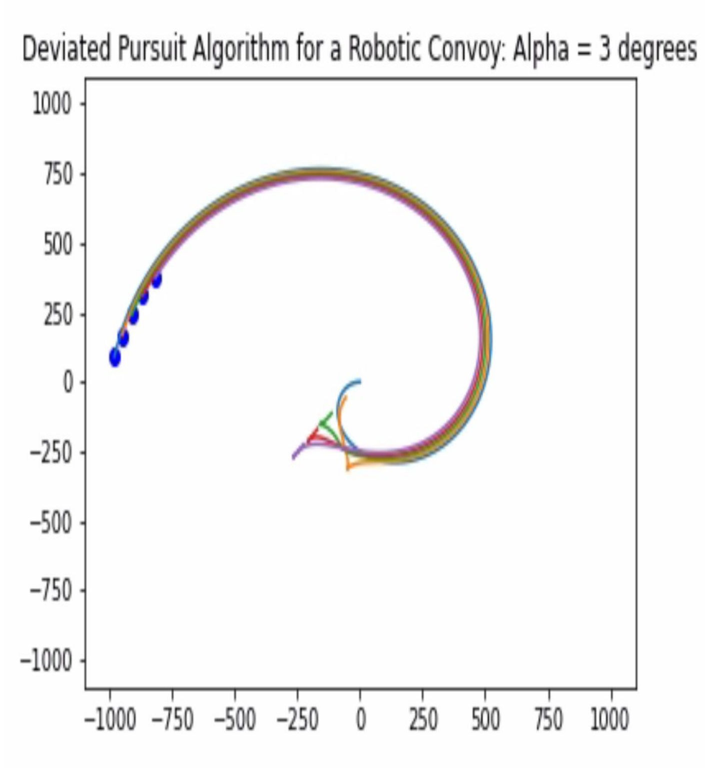
Deviated Pursuit Sine Simulation

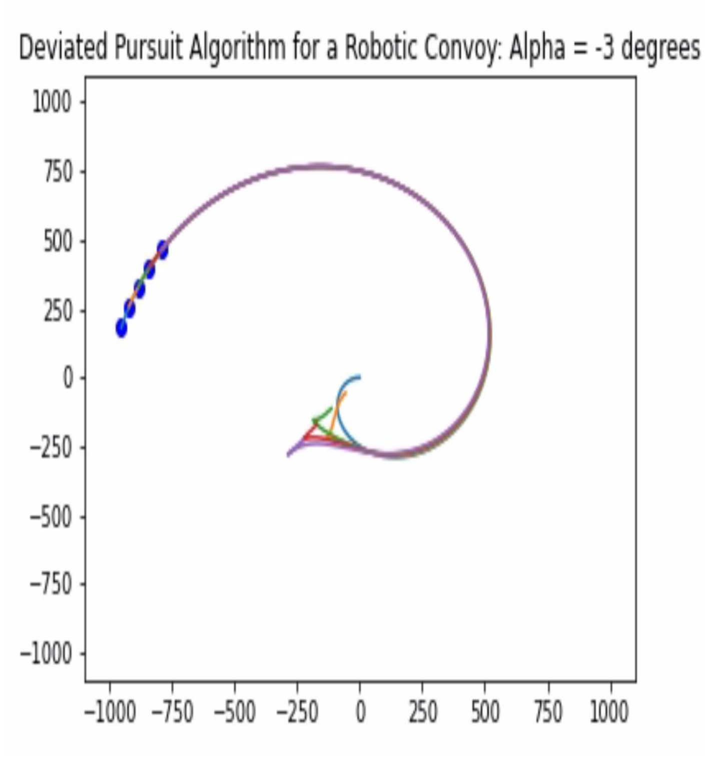




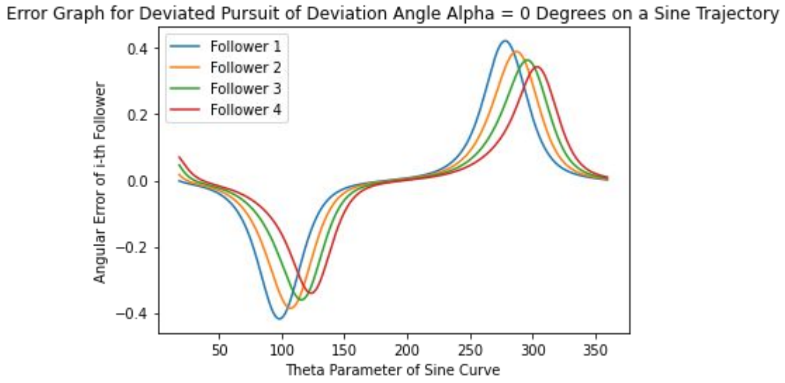


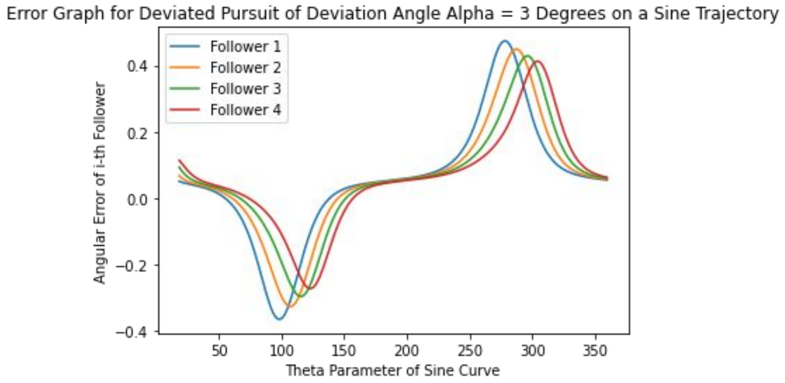
Deviated Pursuit Spiral Simulation

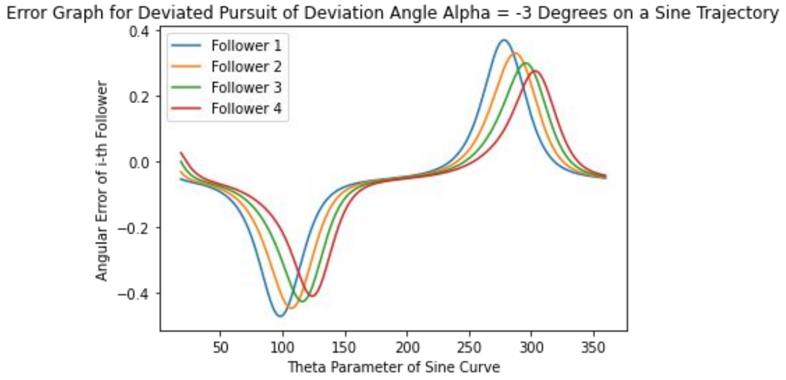




Error Graphs for different alpha(deviation) values:







Deviated Pursuit Simulation Observations:

- The results observed in our simulations are in line with 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 divide the simulation space into 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 of turning while it degrades the performance in the opposite direction of turning.

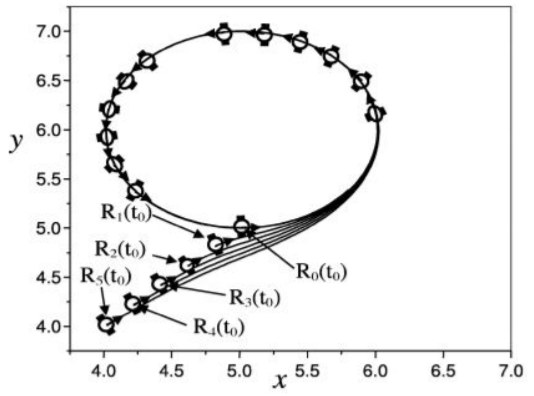


Fig. 6. Path traveled by the convoy, lead robot moving in a circle. The following robots are moving using the deviated pursuit.

Deviated Pursuit Simulation Observations:

- 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 other direction.
- 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.
- The error graphs clearly indicate the unsymmetric performance of deviation algorithms.

Proportional Navigation

- The relation satisfied is: $\omega_{i+1} = K\dot{\sigma}_{i,i+1}$
- The PN guidance is defined by the system:

$$\theta_{i+1} = K\sigma_{i,i+1}$$
$$1 < K \leq 1.1$$

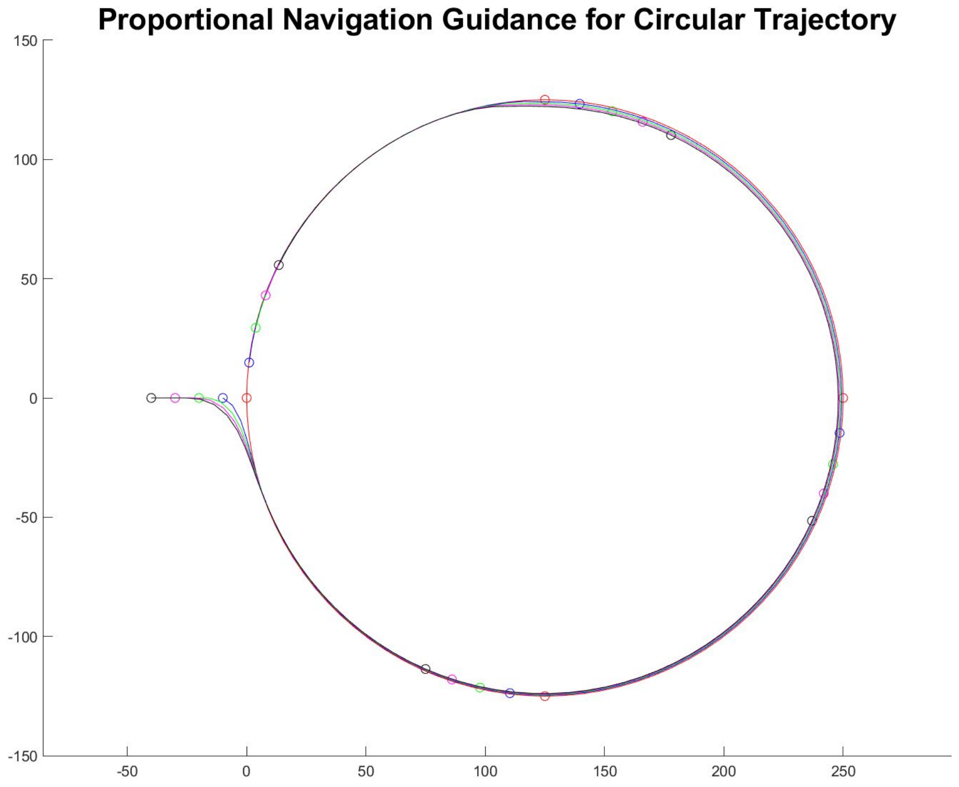
- This system gives the kinematic equations as:

$$\begin{aligned} \dot{r}_{i,i+1} &= v_i \cos\delta_i - v_{i+1}\cos(M\sigma_{i,i+1}) \\ r_{i,i+1}\dot{\sigma}_{i,i+1} &= v_i \sin\delta_i - v_{i+1}\sin(M\sigma_{i,i+1}) \end{aligned}$$

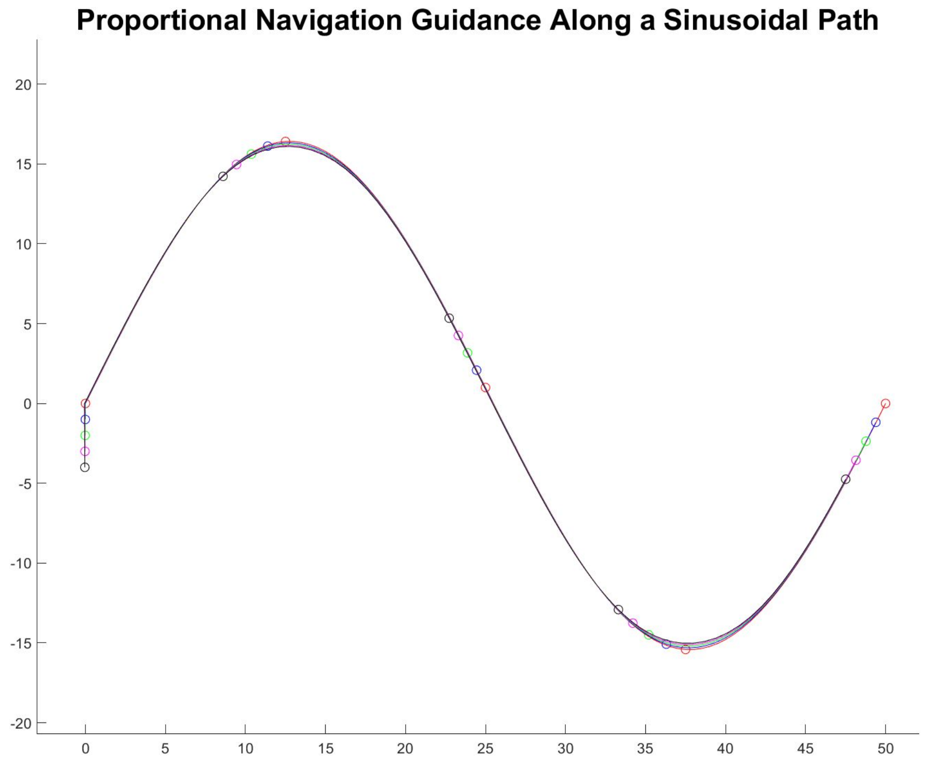
- Finally, adding the constant clearance condition gives our PN formulation as:

$$v_{i+1} = v_i \frac{\cos(\theta_i - \theta_{i+1} + \alpha_{i+1})}{\cos\alpha_{i+1}}$$

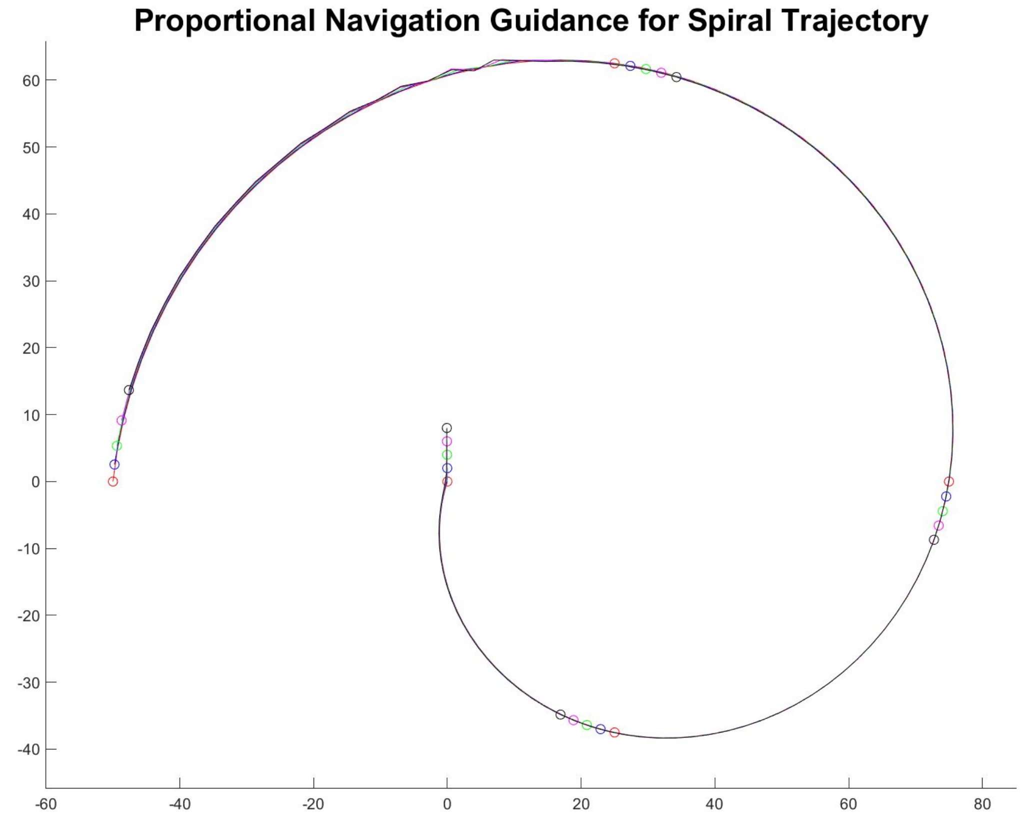
Simulation of PN for Circular Trajectory



Simulation of PN for Sinusoidal Trajectory



Simulation of PN for Spiral Trajectory



Interpretations and Observations

- It is noticed that Proportional Navigation is a generalisation of Velocity Pursuit and Deviated Pursuit,
  - At K = 1, PN behaves like a PP guidance,
  - 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 leader's path, whereas thicker regions are where PN hasn't been able to guide the convoy accurately along the leader's 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 -180 deg (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.
- These simulations are not performed explicitly in the paper, thus not giving a benchmark to validate against. However the interpretations made were quite clear when compared with the PP and DPP laws.

Improvisation 1

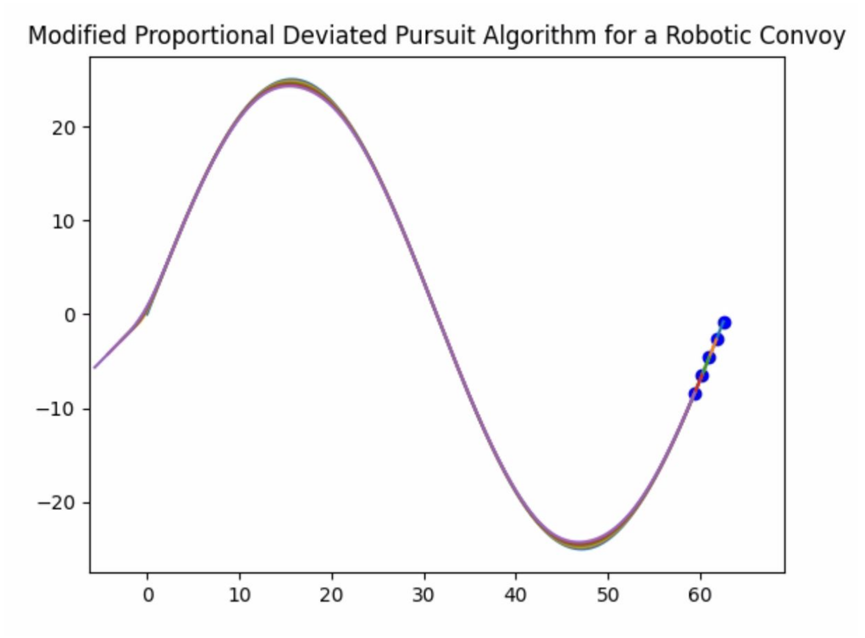
Deviation Error-Proportional Alpha Deviated Pursuit:

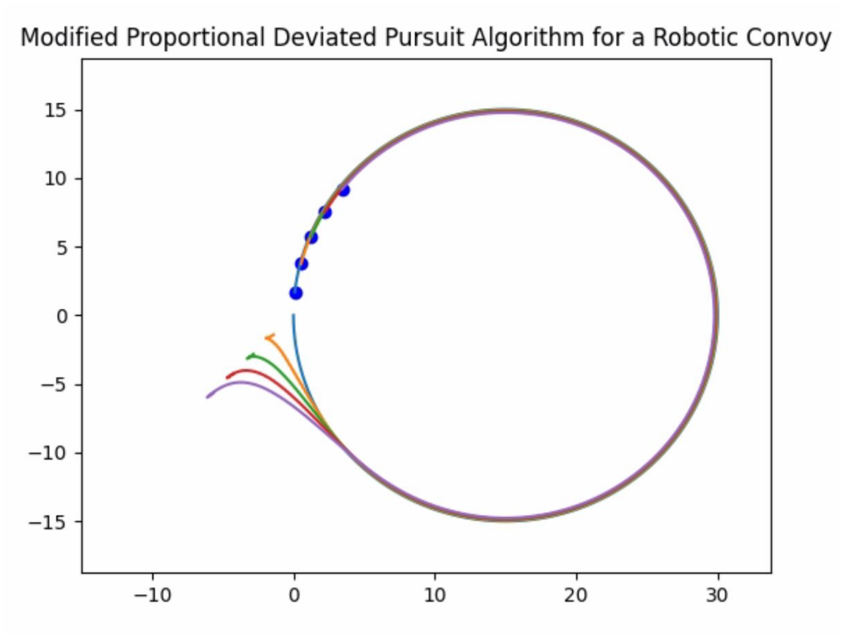
- Velocity pursuit works fairly well only for straight line type of trajectories
- Deviated pursuit works well mostly for trajectories which involve a monotonous change of orientation
- A single value of this deviation hyperparameter doesn't suffice for different types of trajectories or even different directions of change of orientation
- The same issue lies with proportional navigation
- 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 in order to handle trajectories with non monotonous change of orientation

Deviation Error-Proportional Alpha Deviated Pursuit - Problem Definition

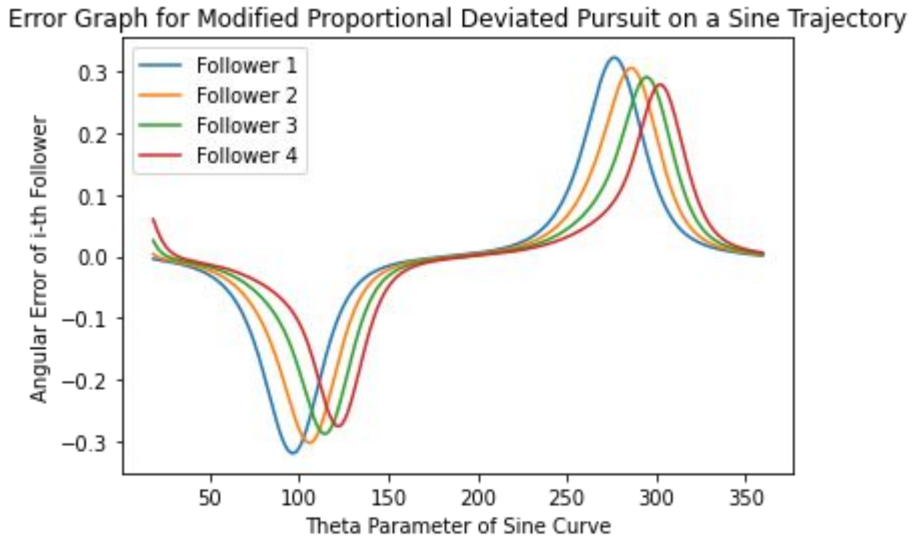
- For a pair of follower and leader robots we define the error as the angle made by the current negative line of sight vector with the vector from the current leader coordinate to its previous coordinate.
- We use a variable deviation angle pursuit algorithm where the deviation angle is computed as the 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.

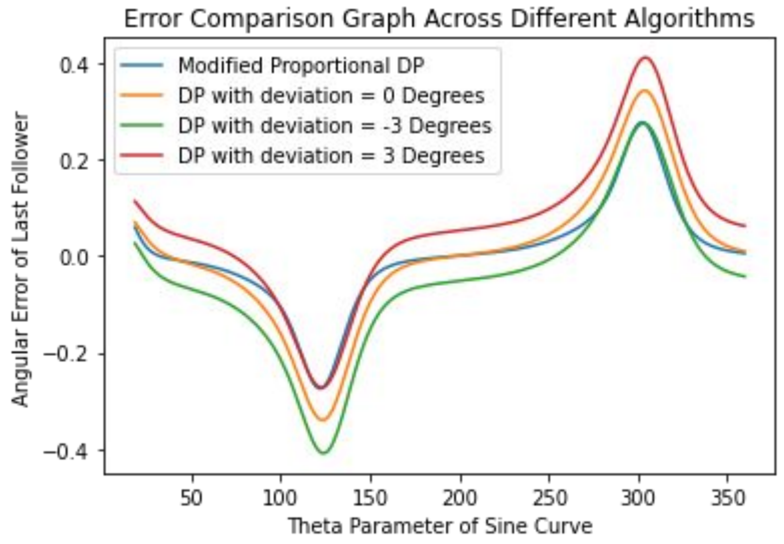
Deviation Error-Proportional Alpha Deviated Pursuit Simulation Plots





Error Graph and Comparison with other Algorithms





Deviation Error-Proportional Alpha Deviated Pursuit - Simulation Observations

- The simulations yield much better results.
- For the circular case it performs as well as deviated pursuit.
- 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.
- The error and cross algorithm comparison graphs clearly illustrate that this algorithm beats 0 deviation algorithm and superimposes the better performance of the positive and negative deviation algorithms at maximas and minimas.

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.

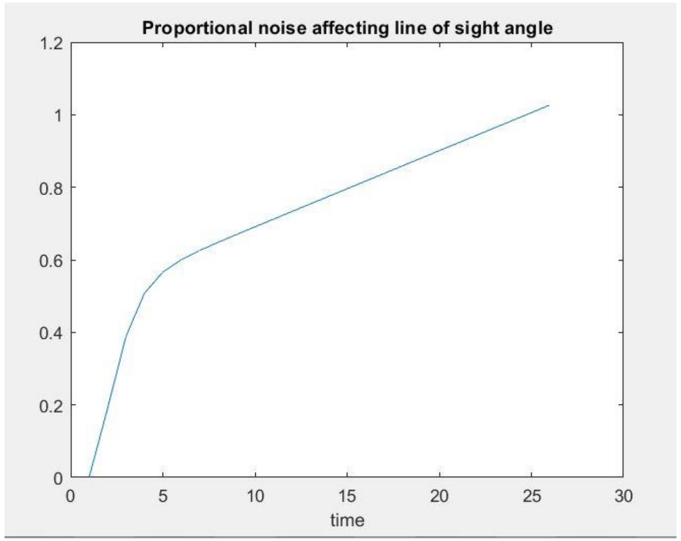
Where n(t) is sensor noise

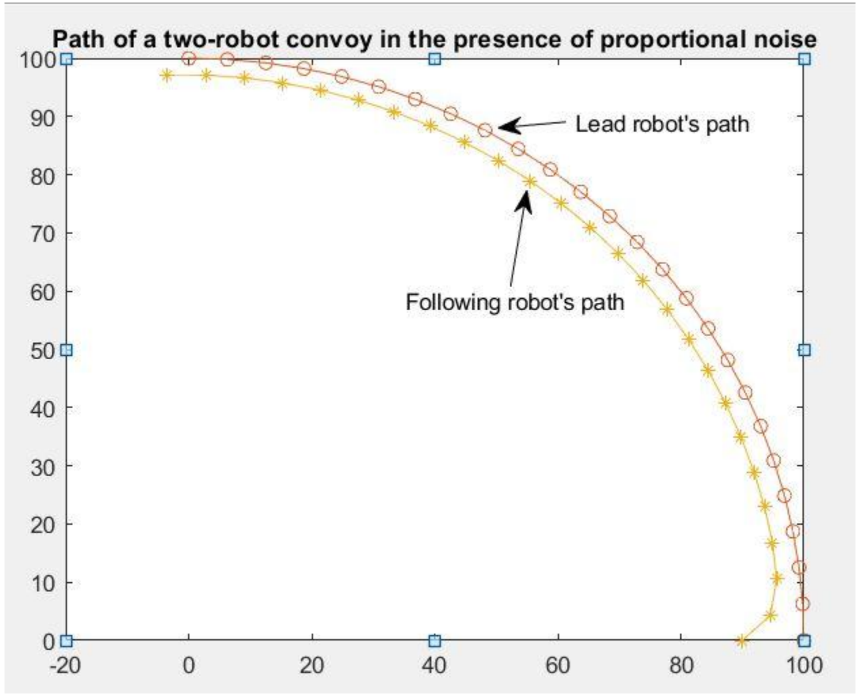
$$\sigma_{M01}(t) = \sigma_{01}(t) + n(t).$$

Case 1: Proportional noise

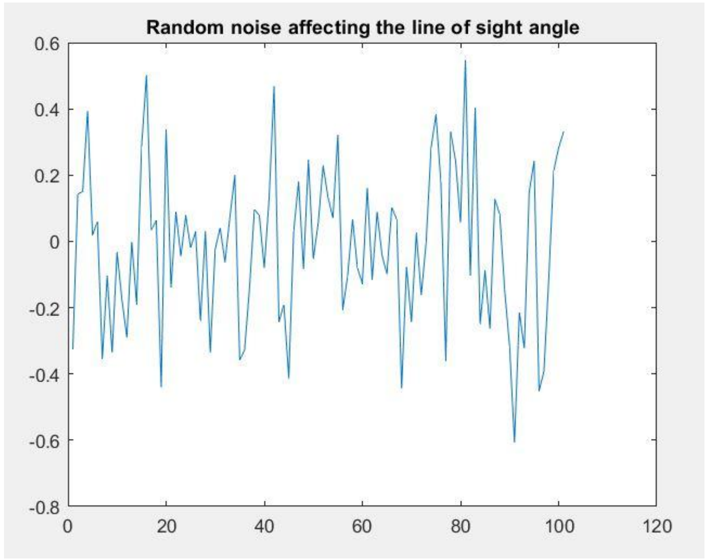
- Noise is proportional to the line of sight angle.

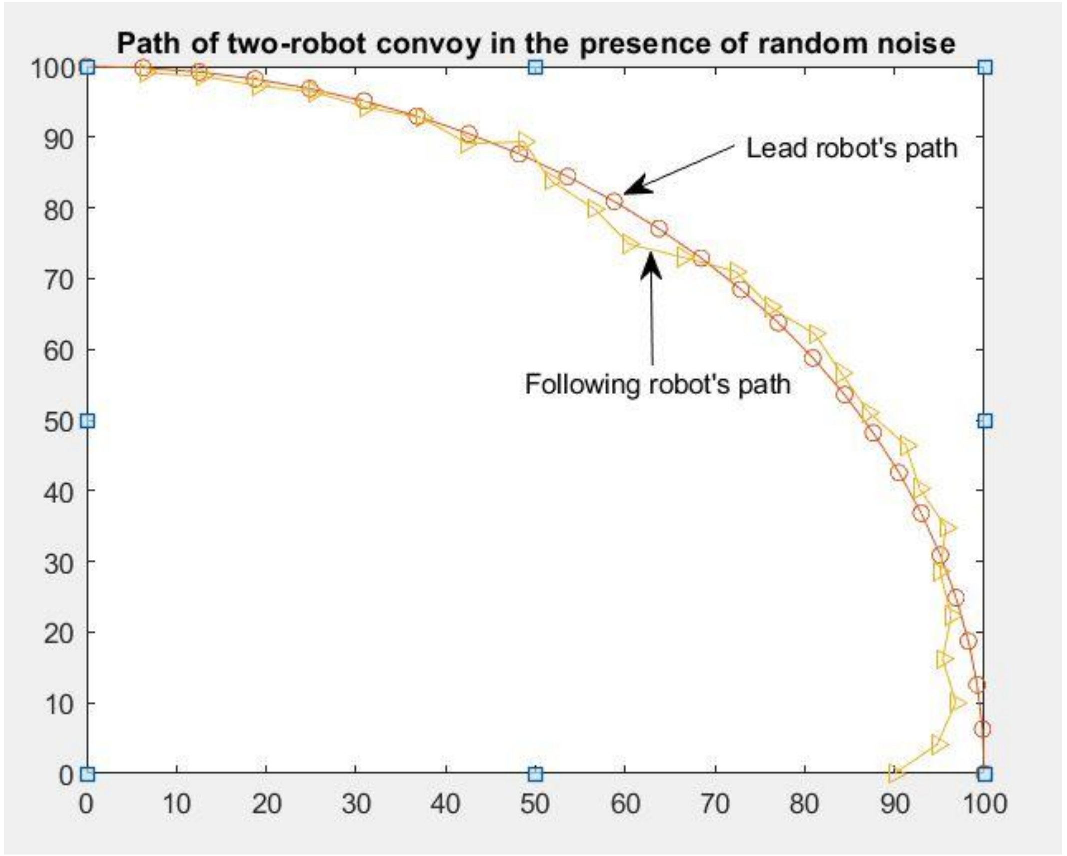
$$n(t) = \sigma_{01}(t)/3.$$





Case 2: Random Noise





Noise Observations

- Error in the proportional noise case is quite larger than the random noise. It is roughly ⅓ rd of , 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 of following robot will be less deviated from desired path of lead robot, in case of random noise compared to the proportional noise case and the same can be seen in the above simulations.

Proportional Noise attenuation by filtering

- In first case,where noise is proportional to the line of sight angle: $n(t) = \sigma_{01}(t)/3.$ An An appropriate correction factor can be used to filter the LOS angle.

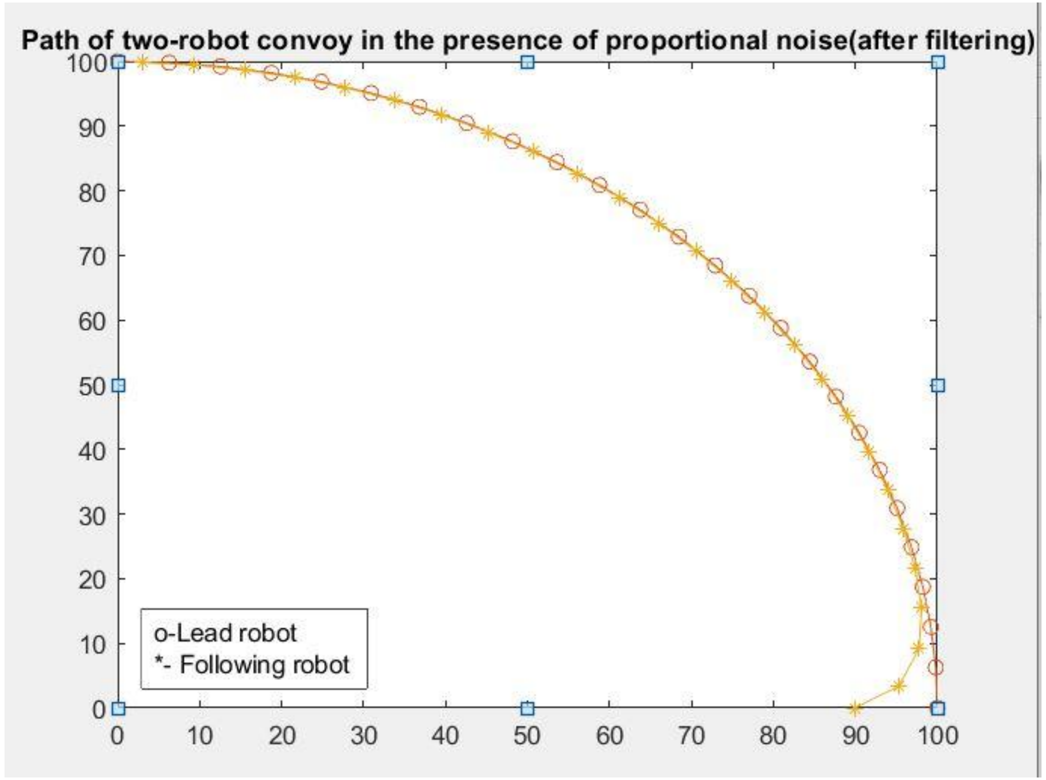
$$\sigma_{M01}(t) = \sigma_{01}(t) + \eta(t)$$
$$\sigma_{M01}(t) = \sigma_{01}(t) + \sigma_{01}(t)/3$$
$$\sigma_{M01}(t) = 4/3 * \sigma_{01}(t)$$

- 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.

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

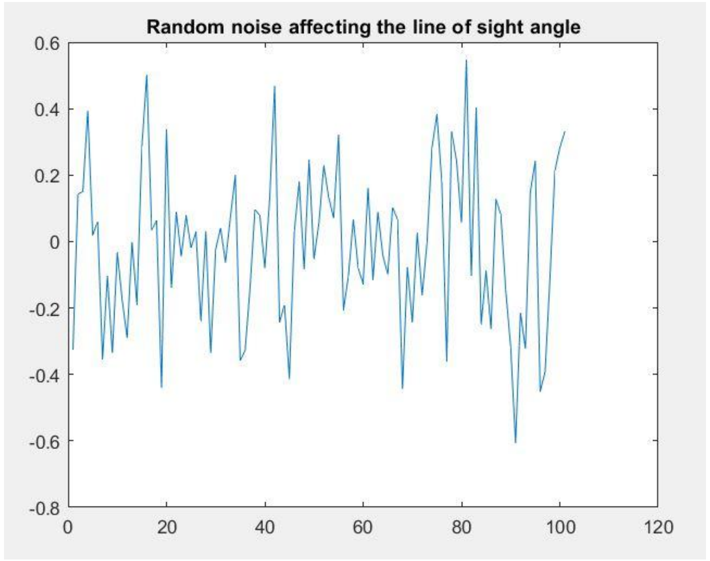
$\sigma_{CM01}(t)$ = corrected Los angle

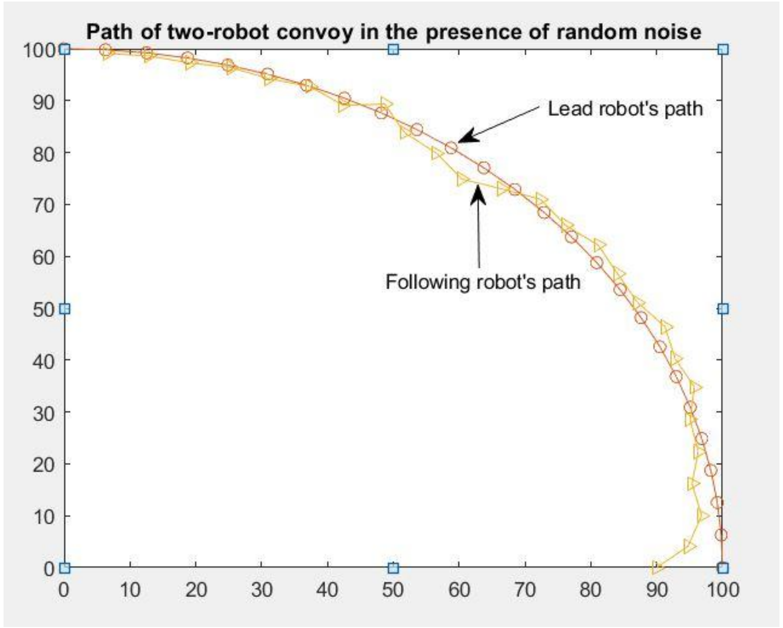
Simulation for two-robot convoy



Random Noise Attenuation

- 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.

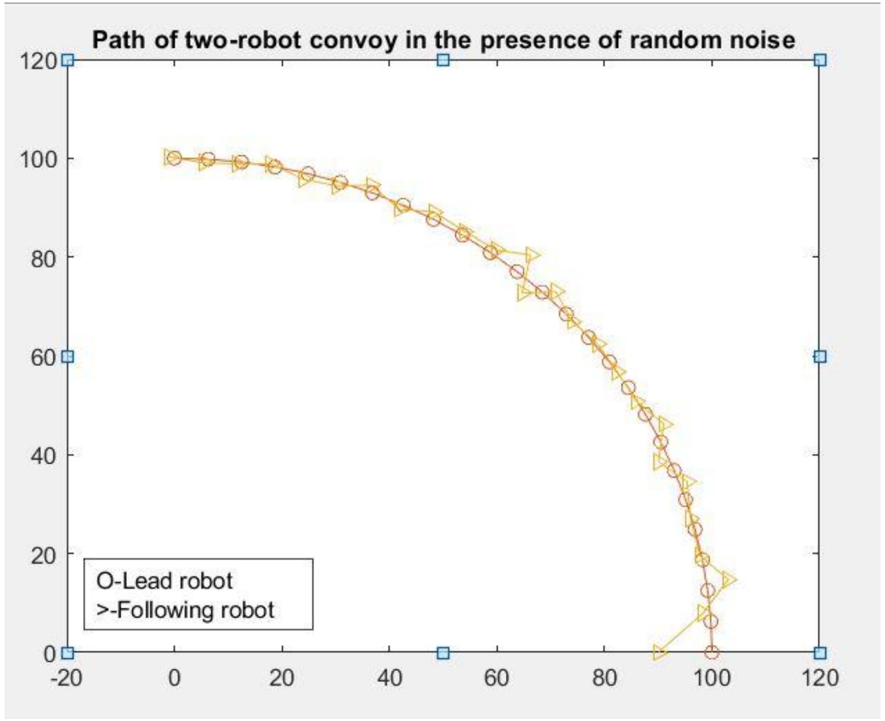
$$xf(i) = xf(i-1) + k * d(i) * cos(\sigma_{i,i+1}(i)) + v * cos(\sigma_{i,i+1}(i)) * dt$$
$$yf(i) = yf(i-1) + k * d(i) * sin(\sigma_{i,i+1}(i)) + v * sin(\sigma_{i,i+1}(i)) * dt$$

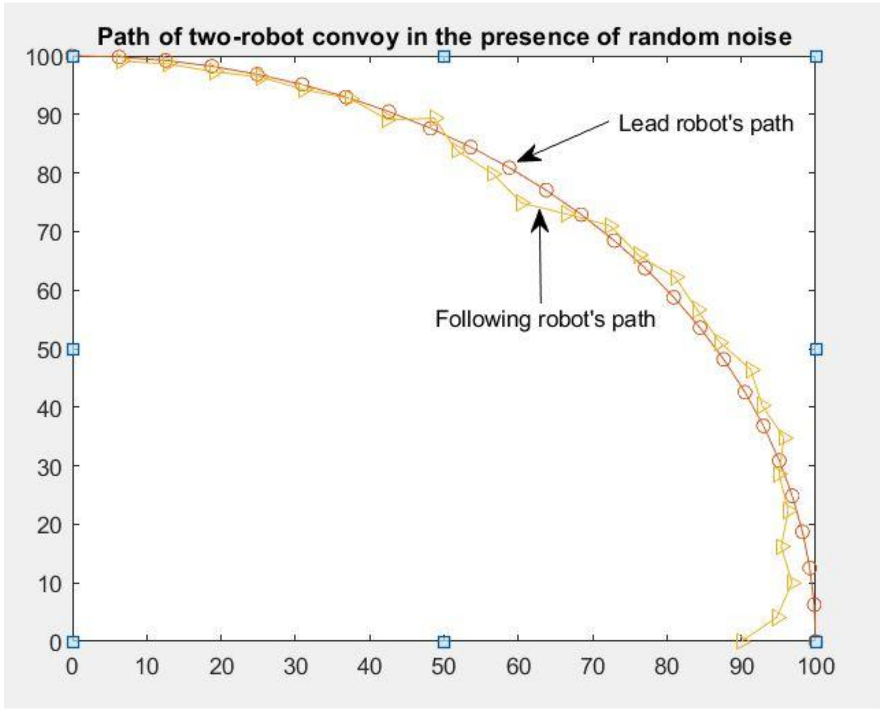
Where d(i) is the cross-track error.

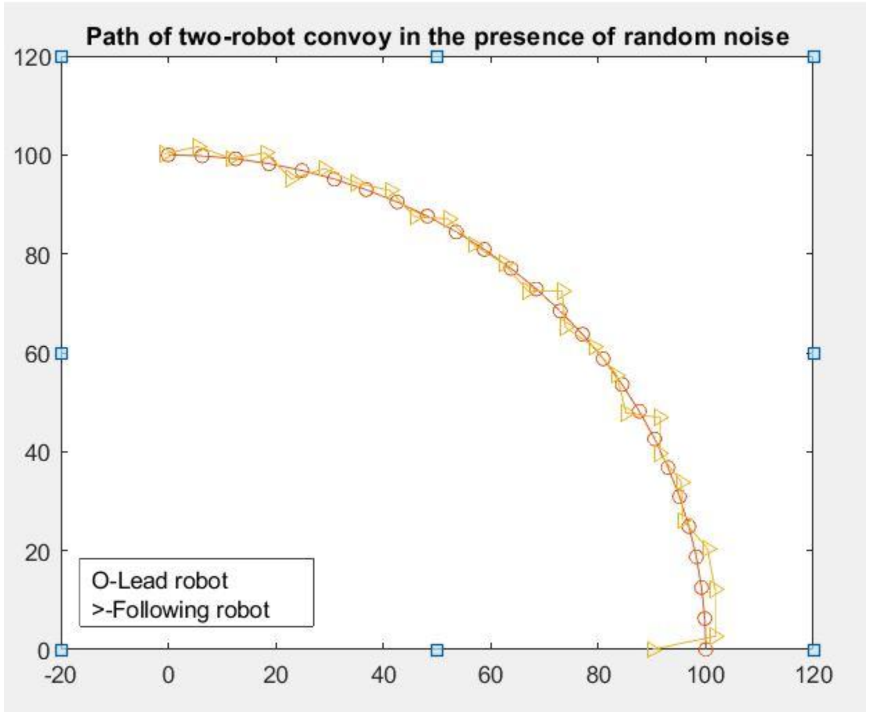
- K has to be chosen with care,otherwise it would shoot up the error and lead to high deviation from desired path.

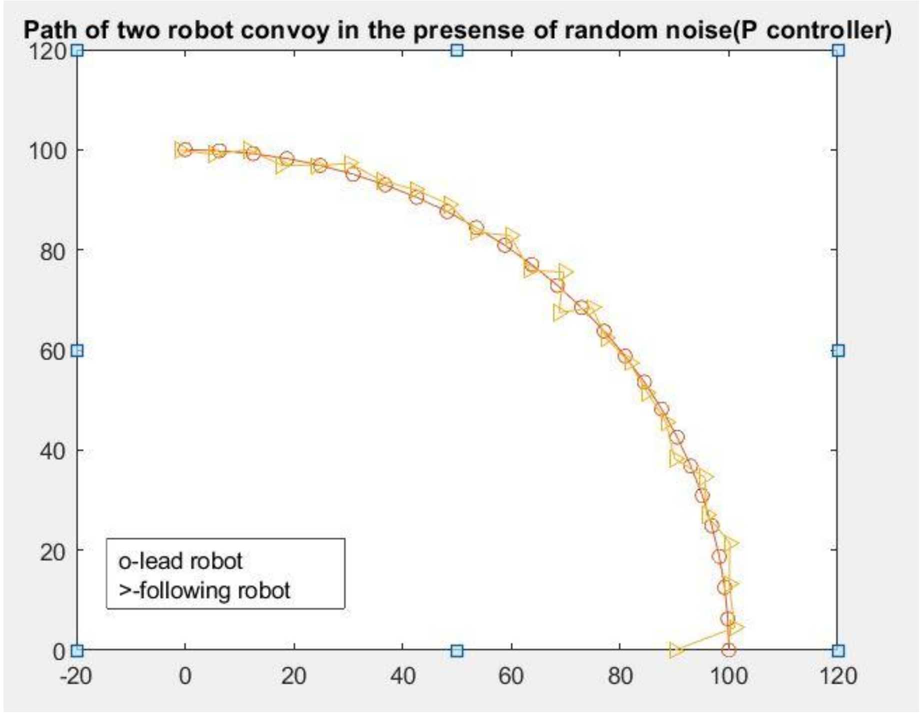
$$k(i) = (\sigma_{i,i+1}(i) + LB(\eta(t))) / (\sigma_{i,i+1} + UB(\eta(t)))$$

Simulation of two-robot convoy in the presence of random noise






Path of two-robot convoy in the presence of random noise
O-Lead robot
>-Following robot
120
100
80
60
40
20
0
-20
0
20
40
60
80
100
120


Path of two robot convoy in the presense of random noise(P controller)
o-lead robot
>-following robot
120
100
80
60
40
20
0
-20
0
20
40
60
80
100
120

Conclusion:

- 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

Contributions:

- **S Tarun Prasad (ME17B114)**: Deviated Pursuit Simulation and Interpretation, Improvisation 1 Deviation-Error Proportional Deviated Pursuit Ideation and Simulation, Presentation, Report.
- **Shreeniwas Jagdale (ME17B115):** Pure Pursuit Simulation and Interpretation, Improvisation 3, Presentation, Report
- **Sarin John (CH17B069):** Proportional Navigation Simulation and Interpretation, Presentation, Report
- **Vaibhav tarphe (ME17B116):** Noise Simulation and Interpretation,Noise Attenuation,Presentation,Report

THANK YOU