# A Simulation Framework for Decentralized Formation Control of Non-Holonomic Differential Drive Robots

Fahad Tanveer
PAF-Karachi Institute of Economics and Technology
fahadtanveer84@gmail.com

Muhammad Bilal Kadri
PAF-Karachi Institute of Economics and Technology
bilal.kadri@pafkiet.edu.pk

***Abstract*—Decentralized cooperative control schemes are a prime research focus due to their resemblance to biological systems and many advantages over centralized schemes. This paper presents a simulation framework for a decentralized cooperative control scheme for differential drive mobile robots, with focus on formation control and obstacle avoidance. The framework employs a hierarchical three layered model. The highest layer is responsible for defining intermediate waypoints followed by a navigation layer and a trajectory tracking layer. The navigation layer employs virtual and behavioral structures along with artificial potential field functions using non-linear systems theory to generate robot trajectories. Due to non-holonomic nature of the differential drive robots a robust sliding mode controller is employed for trajectory tracking. Simulation results for individual layer and for the integrated platform are presented, for formation control with obstacle avoidance in a practical scenario with reasonable assumptions. Simulation results validate the working of the proposed scheme.**

***Keyword—Decentralized control; cooperative control; formation control; obstacle avoidance; differential drive robots; artificial potential field; virtual structures; behavioral structures; non-holonomic; sliding mode control***

## I. INTRODUCTION

Advances in embedded systems and associated technologies, have paved the way for research on multi-agent robot systems to effectively co-ordinate assigned tasks. The primary motivation drawing attention to this problem can be linked to biological inspirations to develop multi-robot solutions to cater problems. There are two main areas associated with cooperative control for multi-agent systems; categorized either as formation control or non-formation control problems [1]. Typical approaches include leader follower, behavioral and virtual structure/leader via artificial potential fields and graph rigidity [2]. On the other hand non-formation cooperative control problems focus on applications such as task/role assignment, payload transport, traffic control, search and rescue, cooperative timing, reconnaissance, surveillance, etc.

Cooperative control can be deployed either in a centralized or a decentralized scheme with latter having several advantages [3]. In distributed schemes there is reduced communication and sensing requirement with improved robustness, reliability and scalability as things evolve in a parallel fashion and only neighborhood information is more necessary.

The application of cooperative formation control schemes for non-holonomic mobile robots such as differential drive robots poses additional challenges due to motion constraints. The methods used in literature to cater these motion problems can be categorized as [4]: 1) sensor-based reactive control approach to navigation problems where the emphasis is on interactive motion planning in dynamic environments; 2) decomposition of navigation problem into a path planning phase and a path execution phase, where the path is generated by optimization algorithms; 3) the third category employs a similar path planning phase with trajectory tracking where temporal requirement is necessary as compared to path following.

One of the many ways to tackle the problem is designing multi-layered controllers by employing separate navigation schemes along with robust trajectory tracking controllers. This work is part of an initial study to realize such a test platform. The overall task was divided into four sub-problems: 1) development of a framework, presented in Section II; 2) investigation of a navigation layer for formation control and obstacle avoidance to generate required trajectory, presented in Section III; 3) investigation of a robust trajectory tracking controller for following the generated path with temporal constraints, presented in Section IV; and 4) simulating the integrated system for a practical scenario under reasonable assumptions, presented in Section V.

## II. SIMULATION FRAMEWORK

Formation control and obstacle avoidance of mobile robots is a navigation problem. Like all navigation systems its can also be decoupled along three abstraction layers [5], [6].

*1) High-Level Planning*: This is the highest level of navigation planning and comprises of intermediate goals/waypoints to complete the mission. These intermediate points can be generated by an intelligent or by an operator input along with other initialization parameters or environment inputs.

*2) Low-Level Planning:* This intermediate layer incorporates navigation scheme for reference trajectory generation, based on intermediate way-points and obstacle

information and other constraints/objectives such as formation constraints.

*3) Execution Layer:* The execution layer is responsible for generating the actuator commands for either trajectory tracking or path following. It is based on the dynamic and/or kinematic model of the mobile robot.

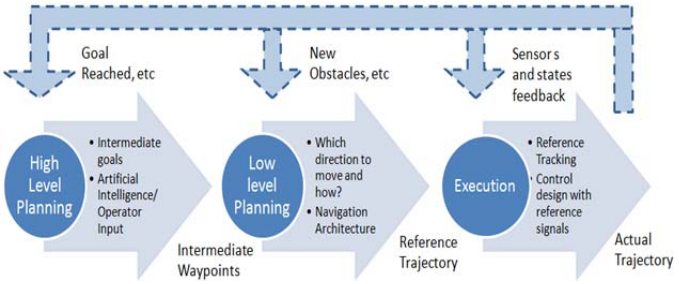


Fig. 1. Navigation abstraction layers for mobile robots.

The proposed framework is illustrated in Fig. 1. The framework is developed in MATLAB/Simulink®.

## III. Navigation Layer

The proposed navigation layer has to achieve two primary objectives. Firstly it has to achieve a geometric formation for varying number of robots, from arbitrary initial positions and then to maintain it during swarming. Secondly it has to avoid obstacles during swarming. The layer is based on virtual and behavioral structures along with artificial potential field functions using non-linear systems theory to generate robot trajectories.

### A. Navigation Model

The navigation model is based on the idea presented in [7]. For the purpose of trajectory generation robots are assumed as point mass holonomic system which simplifies the navigation model and underlying algorithms for on-board implementation. For a system of N robot agents that are assumed to be fully actuated in $R^2$, the model of the $k^{th}$ agent in the system is given by the following dynamics equations in Cartesian coordinates.

$$M\ddot{x_k} + B\dot{x_k} = f_{xk} - k_d\dot{x_k} \qquad (1)$$
$$M\ddot{y_k} + B\dot{y_k} = f_{yk} - k_d\dot{y_k}$$

Where, $(x_k, y_k)$ is the position of the $k^{th}$ mobile robot, M is mass of the agent, B is the damping co-efficient of the mobile robot and $k_d$ is the introduced in the system model to control the transient response of the agents. $f$ is the force on the robot.

### B. Formation Control

The scheme in [7] proposes a regular polygon formation, inscribed in a circle. A potential field strategy is proposed using an attractive and a repulsive force. A law generates trajectories that make the agents attract to the periphery of a circle of radius α and center $(x_c,y_c)$. Assuming robots as virtual identical electric charges governed by the laws of electro-static forces produces the repulsive force. It is ensured that the attractive force to maintain the formation is greater than the overall repulsive force experienced by the agent.

The complete formation controller is given as follows:

$$M\ddot{x}_k + B\dot{x}_k = f_{xk}(FC) - k_d\dot{x}_k \qquad (2)$$
$$M\ddot{y}_k + B\dot{y}_k = f_{yk}(FC) - k_d\dot{y}_k$$

Where,

$$f_{xk}(FC) = f_{xk}(E) - k_{sk}((x_k - x_c)((x_k - x_c)^2 + (y_k - y_c)^2 - \alpha^2))$$
$$f_{yk}(FC) = f_{yk}(E) - k_{sk}((y_k - y_c)((x_k - x_c)^2 + (y_k - y_c)^2 - \alpha^2))$$

The later component is the force that keeps the robots on the circle of radius α and f(E) is the electrostatic force. The gain $k_{sk}$ is the adjusting gain and

$$f_{xk}(E) = k_r \sum_{i=1,i\neq k}^{N} \frac{q_k q_{ki}}{r_{ki}^2} \cos(\theta_{ki}) \qquad (3)$$

$$f_{yk}(E) = k_r \sum_{i=1,i\neq k}^{N} \frac{q_k q_{ki}}{r_{ki}^2} \sin(\theta_{ki})$$

Where, $\cos(\theta_{ki}) = \frac{x_k - x_{ki}}{|r_{ki}|}$ and $\sin(\theta_{ki}) = \frac{y_k - y_i}{|r_{ki}|}$

Since the trajectory planning controller assumes a virtual point mass in place of the robots, it is valid to assume M=1 kg, B=1 N-sec/m and q= 1 C. In [7] no specific criterion was given to obtain the best response for different values of α. Therefore a multitude of simulations were performed to determine a suitable criterion. An additional practical constraint was imposed, that to achieve the desired formation as quickly as possible. In this regard $k_d$=9 N-sec/m was assumed as in [7], which was found to give good damping characteristics. At first simulations were developed for α=3 with $k_{sk}$=100 and $k_r$=1000 as in [7]. Then $k_{sk}$ and $k_r$ were varied in the range [0.1,50] and [1, 500], respectively for α=1m; it was found that with $k_{sk}$=33.3 and $k_r$=111.1 $Nm^2/C^2$ the trajectories converge to the desired formation with sufficient rate and limited errors. Hence it was concluded that for any value for α, $k_{sk}$ should be scaled proportionally by α and $k_r$ by $1/\alpha^2$.

The simulation results for N=3 and N=4 agents are given in Fig. 2 for $(x_c,y_c)$ =(0,0) for t=3 seconds for α=1m. The simulations confirm good convergence of trajectories to a regular polygon formation.

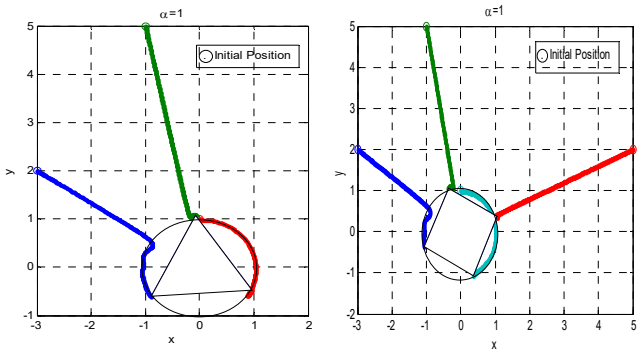


Fig. 2. Simulation results for α=1m and for N=3 and N=4 agents.

### C. Swarming

It is intuitive that moving the center of the circle simply moves the entire formation. Therefore swarming can be

achieved by a higher level controller that merely updates the center of the circle on the desired trajectory.

The same point mass dynamics model is proposed to update the center of the circle. This can be perceived as a virtual robot at the center of the formation.

$$M\ddot{x}_{cv} + B\dot{x}_{cv} = f_{xcv} - k_d\dot{x}_{cv} \qquad (4)$$
$$M\ddot{y}_{cv} + B\dot{y}_{cv} = f_{ycv} - k_d\dot{y}_{cv}$$

Where, $x_{cv}$ and $y_{cv}$ are now the moving center of the circle (virtual robot) and $f_{xcv}$ and $f_{ycv}$ are the virtual forces that can be updated either by the higher layer.

The simulation results for the swarming scenario prove the scheme. The trajectory for center of the formation is based on the sinusoidal control force as input to the point mass model, For α=3: $f_{xcv} = 8$, $f_{ycv} = 24\sin(\frac{x_{cv}}{4})$ and for α=1, $f_{ycv} = 4\sin(x_{cv})$. Moreover the initial coordinates for α=3, of the center $(x_{cv}, y_{cv}) = (4,5)$ and the initial coordinates of the robots are (5,5,), (2,4), (4,4) and (5,1) and for α=1, $(x_{cv}, y_{cv}) = (1,1)$ and the initial coordinates of the robots (0,2), (1,2), (2,0) and (0,1). Moreover it is considered that at midway of the simulation one robot malfunctions and has to leave the formation. Results for α=1are presented in Fig. 3.

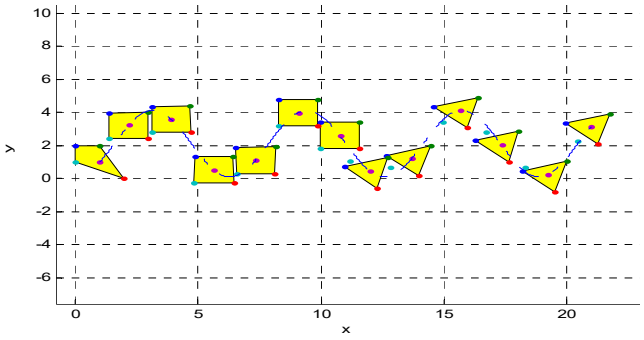


Fig. 3. Simulation for swarming controller α=1.

### *D. Obstacle Avoidance*

In order to avoid local minima problem, the obstacle avoidance control law in [7], is based on rotating fields inscribing the obstacle. A clockwise rotating field is proposed around the obstacle for agents approaching the obstacle from the right hand side and vice versa. In this case the field will never be in the direction of the motion and hence no local minima will be created.

The following equation provides the complete obstacle avoidance force structure.

If $r_k \leq r_a$

$$f_{xk}(OA) = f_{xk}(FC)|_{r_k=r_a} + \frac{k_{OA}(f_k(FC)|_{r_k=r_a})f_{xkrn}}{r_k^2}\left(\frac{1}{r_k} - \frac{1}{r_a}\right)$$

$$f_{yk}(OA) = f_{yk}(FC)|_{r_k=r_a} + \frac{k_{OA}(|f_k(FC)|_{r_k=r_a}|)f_{ykrn}}{r_k^2}\left(\frac{1}{r_k} - \frac{1}{r_a}\right) \qquad (5)$$

Else (on reaching other side of obstacle):

$$f_{xk}(OA) = f_{xk}(FC)|_{r_k=r_a}\exp(-\tau r_k) + f_{xk}(FC)(1 - \exp(-\tau r_k))$$

$$f_{yk}(OA) = f_{yk}(FC)|_{r_k=r_a}\,exp(-\tau r_k) + f_{yk}(FC)(1 - exp(-\tau r_k))$$

Where, $f_k(FC)|_{r_k=r_a} = (f_{xk}(FC)|_{r_k=r_a}, f_{yk}(FC)|_{r_k=r_a})$ and is equal to $f_k(FC) = (f_{xk}(FC), f_{yk}(FC))$ at the moment the kth agent enters in the influence of the obstacle.

In addition we also need to control the center of the formation circle (assumed to be a virtual robot), agents moving around the obstacle slow down since they have to cover a larger distance as compared to the virtual robot that can pass through the obstacle. A new force is defined that tries to keep the virtual robot in the geometric center of the formation.

In order to derive the expression of this force let us define [7].

$$\boldsymbol{r_m}\text{=min of } \boldsymbol{r_k}, \boldsymbol{k} \in \{\boldsymbol{1}, \boldsymbol{2}, .. \boldsymbol{N}\} \qquad (6)$$

And $x_m = \frac{1}{N}\sum_{k=1}^{N} x_k$ and $y_m = \frac{1}{N}\sum_{k=1}^{N} y_k$ , co-ordinates of the geometric center of the formation.

Now the control force in the behavioral structure can be defined as:

If $r_m \leq r_a$,
$$f_{xcv}(OA) = f_{xcvdes} + k_p(x_m - x_{cv})(1 - \exp(-\tau r_m))$$
$$f_{ycv}(OA) = f_{ycvdes} + k_p(y_m - y_{cv})(1 - \exp(-\tau r_m)) \qquad (7)$$
Else (on reaching other side of obstacle)
$$f_{xcv}(OA) = f_{xcvdes}$$
$$f_{ycv}(OA) = f_{ycvdes}$$

Where, again $f_{cvdes} = (f_{xcvdes}, f_{ycvdes})$ are the forces on the virtual robot in absence of the obstacle.

For obstacle avoidance simulation we assume a rectangular obstacle at (5,5.8) with half side lengths (1,1) and input force (2,2) for α=1. Again a multitude of simulations were performed to obtain good values of $k_p$ and $\tau$ for the range [0.1,10] and [0.01,10]. For α=1, $k_p = 3.3$, and $\tau = 0.05$ results in good trajectories. $r_a = 1$ is set as this only defines range proportional to size of the obstacle. Simulation result is shown in Fig. 4. It can be observed from the figure that the obstacle is successfully avoided by all the robots. When the obstacle is encountered the formation breaks. Once the obstacle is avoided all robots come into a formation and start to move towards the goal.

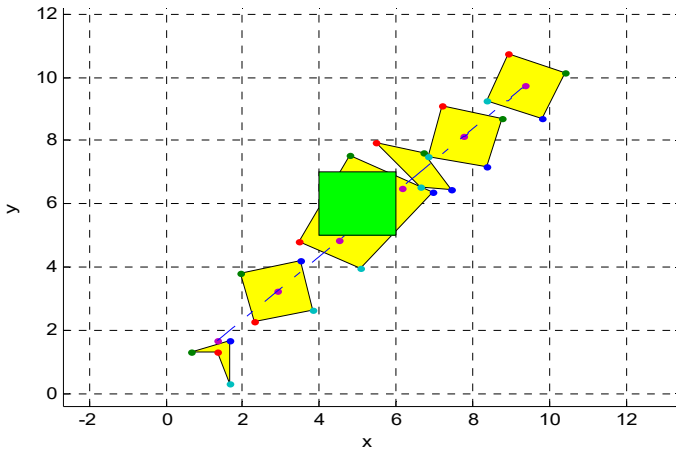


Fig. 4. Simulation results for obstacle avoidance control for α=1.

## IV. Trajectory Tracking Layer

The proposed tracking controller for the said application is a sliding mode controller in polar co-ordinates as presented in [8]. The controller is selected as it helps in tracking the trajectory in forward and reverse directions (in case of heading errors greater than ±90º) so that tracking errors reduce significantly faster and temporal requirements of formation control are met. The controller is globally valid except at a very small region around the origin due to singularity in the polar model of the differential drive robot.

### *A. Mathematical Model of Differential Drive Mobile Robots*

The simplest kind of such robot has two wheels actuated by a set of two different actuators (motors) with a third passive wheel. The dynamic model of the differential drive robot with non-holonomic constraints can be derived from the general Euler-lagrange form as in [8]:

$$M(q)\ddot{q} + V(q,\dot{q})\dot{q} + G(q) = B(q)\tau + A^T(q)\lambda \qquad (8)$$

Where, the non-holonomic constraint is given as:

$$A(q).\dot{q} = 0 \qquad (9)$$

Where, $q \in R^n$ is the state vector, $\tau \in R^r$ is input torque and r<n. $M(q) \in R^{nxn}$ is a positive definite and symmetric inertia matrix, $V(q,\dot{q}) \in R^n$ is the vector of centripetal and coriolis torques, $G(q)$ is the vector of gravitational torque and $B(q) \in R^{nxr}$ input matrix. $\lambda$ is the Lagrange multiplier of constrained forces and $A(q) \in R^{mxn}$ represents the non-holonomic constraint matrix. The dynamics model can be modified by eliminating the non-holonomic constraint term $A^T(q)\lambda$. Considering all disturbances as lumped parameters $\boldsymbol{\tau_d}$ and control torque $\boldsymbol{\tau_{con}}$.

$$\boldsymbol{H(q)\dot{z} + \tau_d = \tau_{con}} \qquad (10)$$

Where, $H(q) = [S^T(q)B(q)]^{-1}S^T(q)M(q)S(q)$, $\dot{q} = S(q)z$ and $\tau_{con} = [\tau_L, \tau_R]^T$, torques exerted by the left and right wheels.

### *B. Trajectory Tracking Sliding Mode Controller*

Under the assumption that $H(q)$ is known; a control torque given by in (11) can be shown to achieve asymptotic position tracking for desired velocity$v_d \geq 0$.

$$\boldsymbol{\tau_{con} = H(q)\dot{z} + H(q)u} \qquad (11)$$

With u given as [8]:

$$u = \dot{M}(\rho_r,\varphi_r,\theta_r)\begin{pmatrix}-k_1\rho_e+\dot{\rho}_r\\-k_2\varphi_e+\dot{\varphi}_r\\-k_3\theta_e+\dot{\theta}_r\end{pmatrix} + M(\rho_r,\varphi_r,\theta_r)\begin{pmatrix}-k_1\dot{\rho}_e+\ddot{\rho}_r\\-k_2\dot{\varphi}_e+\ddot{\varphi}_r\\-k_3\dot{\theta}_e+\ddot{\theta}_r\end{pmatrix} - \dot{M}(\rho_d,\varphi_d,\theta_d)\begin{pmatrix}\dot{\rho}_d\\\dot{\varphi}_d\\\dot{\theta}_d\end{pmatrix} - M(\rho_d,\varphi_d,\theta_d)\begin{pmatrix}\ddot{\rho}_d\\\ddot{\varphi}_d\\\ddot{\theta}_d\end{pmatrix} - \begin{pmatrix}\frac{d}{dt}[k_0(\mathrm{Cr}\ \mathrm{sgn}(\rho_e)|s_\theta| - \mathrm{Sr}\ \mathrm{sgn}(\varphi_e)|s_\theta|)]\\\frac{d}{dt}[k_0(\mathrm{Sr}\ \mathrm{sgn}(\rho_e)|s_\theta| + \mathrm{Cr}\ \mathrm{sgn}(\varphi_e)|s_\theta|)]\end{pmatrix} - Qs - Psgn(s) \qquad (12)$$

Where, the robot trajectory is $(\rho_r, \varphi_r, \theta_r)$ and desired trajectory is $(\rho_d, \varphi_d, \theta_d)$, and

$$\rho_e = \rho_r - \rho_d, \quad \varphi_e = \varphi_r - \varphi_d, \quad \theta_e = \theta_r - \theta_d$$

$$s = M(\rho_r,\varphi_r,\theta_r)\begin{pmatrix}s_\rho + k_0 sgn(\rho_e)|s_\theta|\\s_\varphi + \frac{k_0}{\rho_r}sgn(\varphi_e)|s_\theta|\\s_\theta\end{pmatrix}$$

$$s_\rho = \dot{\rho}_e + k_1\rho_e, \quad s_\varphi = \dot{\varphi}_e + k_2\varphi_e, \quad s_\theta = \dot{\theta}_e + k_3\theta_e$$

$$M(m_1,m_2,m_3) = \begin{pmatrix}\cos(m_2-m_3) & -m_1\sin(m_2-m_3) & 0\\ \sin(m_2-m_3) & m_1\cos(m_2-m_3) & 1\end{pmatrix}$$

Cr=$\cos(\varphi_r - \theta_r)$ and Sr=$(\varphi_r - \theta_r)$

$k_0 > 0$

$Q = diag(q_1, q_2) > 0$

$P = P_0 + P_f = diag(p_1, p_2)$

$P_0 = diag(\eta_1, \eta_2) > 0, \quad P_f = diag(f_{m1}, f_{m2}) > 0,$

$sgn(s) = [sgn(s_1)\ \ sgn(s_2)]^T$

Similarly, for $v_d < 0$, the following control law will similarly achieve asymptotic position tracking [8]:

$$u = \dot{N}(\rho_r,\varphi_r,\theta_r)\begin{pmatrix}-k_1\rho_e+\dot{\rho}_r\\-k_2\varphi_e+\dot{\varphi}_r\\-k_3\theta_e+\dot{\theta}_r\end{pmatrix} + N(\rho_r,\varphi_r,\theta_r)\begin{pmatrix}-k_1\dot{\rho}_e+\ddot{\rho}_r\\-k_2\dot{\varphi}_e+\ddot{\varphi}_r\\-k_3\theta_e+\ddot{\theta}_r\end{pmatrix} - \dot{N}(\rho_d,\varphi_d,\theta_d)\begin{pmatrix}\dot{\rho}_d\\\dot{\varphi}_d\\\dot{\theta}_d\end{pmatrix} - N(\rho_d,\varphi_d,\theta_d)\begin{pmatrix}\ddot{\rho}_d\\\ddot{\varphi}_d\\\ddot{\theta}_d\end{pmatrix} - \begin{pmatrix}\frac{d}{dt}[k_0(\mathrm{Cr}\ \mathrm{sgn}(\rho_e)|s_\theta| - \mathrm{Sr}\ \mathrm{sgn}(\varphi_e)|s_\theta|)]\\\frac{d}{dt}[-k_0(\mathrm{Sr}\ \mathrm{sgn}(\rho_e)|s_\theta| - \mathrm{Cr}\ \mathrm{sgn}(\varphi_e)|s_\theta|)]\end{pmatrix} - Qs - Psgn(s) \qquad (13)$$

Where,

$$s = N(\rho_r,\varphi_r,\theta_r)\begin{pmatrix}s_\rho + k_0\mathrm{sgn}(\rho_e)|s_\theta|\\s_\varphi + \frac{k_0}{\rho_r}\mathrm{sgn}(\varphi_e)|s_\theta|\\s_\theta\end{pmatrix}$$

$$N(m_1,m_2,m_3) = \begin{pmatrix}\cos(m_2-m_3) & -m_1\sin(m_2-m_3) & 0\\ -\sin(m_2-m_3) & -m_1\cos(m_2-m_3) & 1\end{pmatrix}$$

However, the above two position tracking controllers fail to reduce the heading tracking error when the reference trajectory does not change and the position tracking errors have been sufficiently reduced. Therefore another heading controller needs to become active in the above scenario in order to achieve desired heading and the control input u can be given as

$$u = [0 \quad -k_3\dot{\theta}_e - Q_2 s_\theta - P_2 sgn(s_\theta)]^T \qquad (14)$$

Where, $s = [0\ s_\theta]^T$

The above three sliding mode controllers were tested for - point stabilization, vr> 0 and vr<0. As proposed in [8], for position tracking controllers, $k_3$ has been taken relatively small and Q large in order to satisfy the Lyapunov analysis.

As there was no other specific criteria, simulations were carried out with a wide range to values for $k_0$, $k_1$, $k_2$, $k_3$ , P and Q, however following parameters were finally selected for better observed performance, $k_0$=1.1, $k_1$=2, $k_2$=2, $k_3$=0.002 for position tracking and $k_3$=2 for heading controller, P= Diag(1.4,0.9) and Q= Diag(18,15). Moreover the position tracking error deadbands were defined as of 0.05m and 0.005 rads, respectively. In addition saturation limits of 4.5m/s and 4.5 rad/s are also imposed on the mobile robots linear and angular velocities. Also $v_d < 0$ was detected if the sign of the cosine($\theta_e$) <0.

As validated by simulation results shown in Fig. 5 that the controller initially starts off with minimization of $\rho_e$ and $\varphi_e$ and once these are in the dead band of 0.05m and 0.005 rads respectively, than the heading correction controller is switched. Also if $v_r$<0 the controller starts off with minimization of $\rho_e$ and $\varphi_e$ with $\theta_e$ settling towards π, and as position tracking errors are minimized and the reference trajectory stops, the heading errors are also catered for.

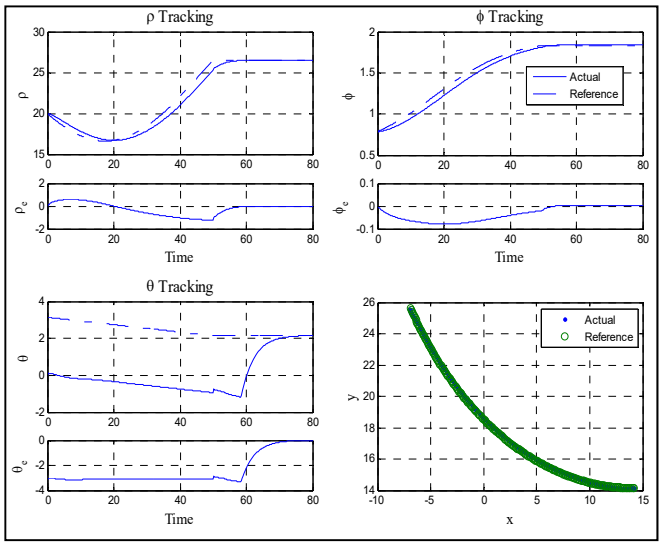


Fig. 5. Trajectory Tracking Controller Simulation Results $v_r$<0.

## V. Integrated Decentralized Co-operative Control System

The integrated system is the same is shown in Fig. 6. In the real world setting each robots embedded controller will incorporate the navigation layer and the tracking layer. The simulation is implemented for a decentralized setting for a total of four actual robots (and one virtual robot). However following assumptions/constraints are set: 1) the initialization parameters other than initial position are set once by the operator(s); 2) initial positions of each robot and obstacle are known beforehand; 3) individual robot generates and tracks its own trajectory; and 4) initial position (and new obstacles) can be updated once after every navigation trajectory tracking is completed.

Simulations have been carried out for the same scenario as given in the obstacle avoidance navigation layer example for α=3. A slight modification in the navigation layer has been made in order to observe the system performance in all modes:

*1)* The system now is implemented with a sample time of 0.01 seconds.

*2)* In the initial 20 seconds the position reference is kept constant so that the desired heading can be matched in order to avoid very large initial errors due to non-holonomic constraints and at the same time the heading controller performance may be analyzed.

*3)* The trajectory way points are generated in batch form for improved performance; with total batch duration of 3 seconds, where in the 1st second, trajectory is generated with 10 way points and the remainder 2 seconds is allocated for the robots to catch up.

Same parameters for the sliding mode tracking controllers have also been used. The simulations results for trajectory tracking have shown good tracking performance. Moreover the formation comparisons are presented in Fig. 7 to 9 with 2 seconds delay for actual trajectory, which reveals good overlapping. Additional results for parameter tracking errors also show good parameter tracking and convergence performance.

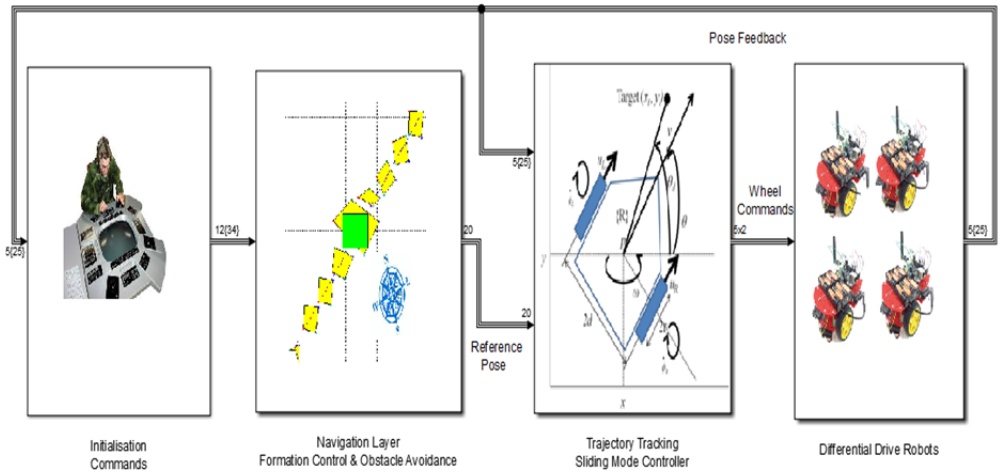


Fig. 6. Proposed framework.

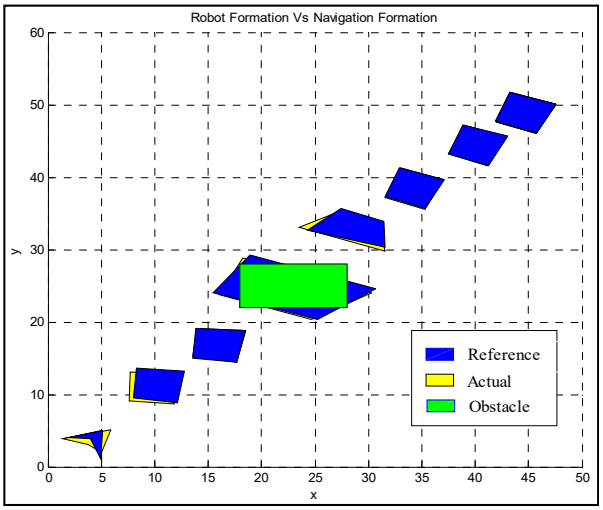


Fig. 7. Integrated system results.

## VI. Conclusion & Future Recommendations

The paper presented a framework for implementation of decentralized cooperative control schemes for formation control and obstacle avoidance scenarios for differential drive robots. Among the multitude of possible future extensions following are proposed in order to realize the system for a more robust performance for practical applications.

*1)* A behavior mode for unknown obstacle avoidance may also be developed for dynamic settings.

*2)* An output feedback trajectory tracking sliding mode controller can also be developed given the nature of sensors employed.

*3)* In addition a module for localization and mapping and pose estimation based on sensory feedback can also be developed to improve the navigation layer performance and pave way for more intelligent and autonomous systems with little user intervention.

*4)* Communication constraints and information uncertainty are critical problems and solutions can be explored for a robust practical implementation.

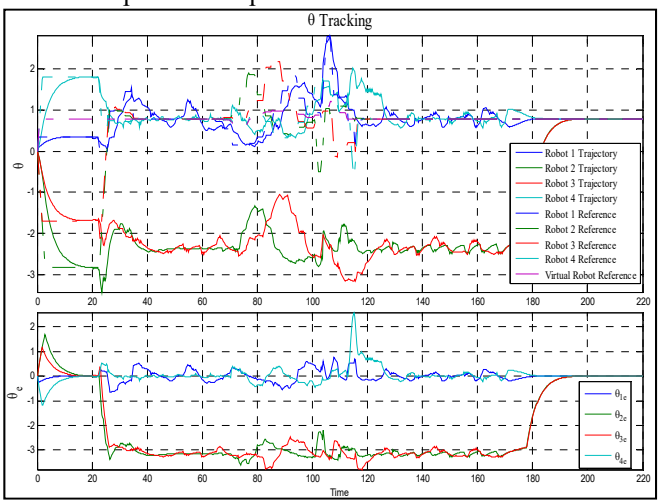


Fig. 8. Rho tracking for integrated simulation platform.

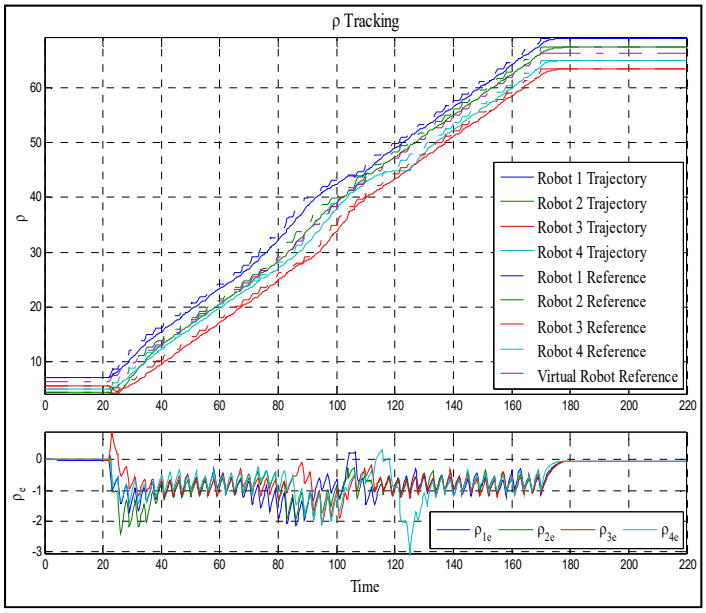


Fig. 9. Heading tracking for integrated simulation platform.

## References

[1] W. Ren, R.W. Beard, E. M. Atkins, *"A Survey of Consensus Problems in Multi-agent Coordination"*, American Control Conference, Portland, OR, USA, June 8-10, 2005.

[2] Y.Q. Chen and Z. Wang, *"Formation Control, A review and a new consideration"*, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005.

[3] A. Loria, J. Dasdemir, N.A. Jarquin, *"Leader–Follower Formation and Tracking Control of Mobile Robots Along Straight Paths"*, IEEE Transactions on Control Systems Technology, 2016, Volume: 24, Issue: 2

[4] X. Chen, Y. Jia, *"Adaptive leader-follower formation control of non-holonomic mobile robots using active vision"*, IET Control Theory & Applications, 2015, Volume: 9, Issue: 8.

[5] R. C. Solea, *"Sliding mode control applied in trajectory-tracking of WMRs and autonomous vehicles"*, MS Thesis, University of Coimbra, 2009

[6] L.F. LEE, *"Decentralised Motion Planning within and Artificial Potential Framework (APF) for Cooperative Payload Transport by Multi-robot collectives"*, MS Thesis, State University of New York, Buffalo, 2004.

[7] H. Rezaee and F. Abdollahi, *"A Decentralized Cooperative Control Scheme With Obstacle Avoidance for a Team of Mobile Robots"*, IEEE Transactions on Industrial Electronics, Vol. 61, No.1, January 2014.

[8] D. Chwa, *"Sliding-Mode Tracking Control of Nonholonomic Wheeled Mobile Robots in Polar Coordinates"*, IEEE Transactions on Control Systems Technology, Vol. 12, no. 4, July 2004.