Shortest Path Planning for Wheeled Mobile Robot using Dummy Targets and MATLAB simulations

Sangita Kamat · D.S. Revankar

Abstract— In robot navigation, obstacle avoidance is major issue of consideration and if not handled properly can result in enormous losses and permanent damages. This paper presents simple algorithm, with a different technique in obstacle avoidance, for devising the shortest path for the robot navigation, while ensuring utmost safety and task competency in reaching the target. The proposed algorithm assumes the obstacles are 2D, stationary and of different shapes and sizes, embedded by virtual, secure elliptical surrounding. This approach combined with the conventional methodology of employing tangential approach in clockwise/counter clock wise directions and Virtual Force Field (VFF), projects it as hybrid model. The novelty of this method is the introduction of intermittent dummy targets to determine the straight-line segments of the shortest path to the goal.

Keywords: Robotic formation · Dummy target · Hybrid model · Lyapunov function

1 INTRODUCTION

Non-holonomic mobile robots have played very important role in control systems. In recent years, the research and development of mobile robots are very active, mostly because of their better potential than other kinds of robots in replacing human beings in civilian, industry, military applications. This paper revisits the problem of obstacle avoidance in mobile robot navigation. A typical dynamic system in robotic application aims at identifying the most efficient motion path from start point to end point (Goal). Obviously, the objective is to have this efficient motion path as close as possible to the reference path.

Real-time obstacle avoidance is one of the challenges to the successful applications of control systems. All mobile robots feature some kind of collision avoidance, ranging from primal algorithms that detect an obstacle and stop the robot short of it, in order to avoid a collision through complex algorithms that permit the robot to detour obstacles. The latter algorithms are much more complex, since they involve not only the detection of an obstacle, but also some kind of quantitative measurements concerning the obstacle's dimensions. Autonomous navigation represents a higher level of performance, since obstacle avoidance and the robot steering toward a given target should go concurrently. Autonomous navigation, in general, assumes an environment with known and unknown obstacles, and it includes global path planning algorithms [1]. Ossama Khatib [2] in 1989 provided a unique real time obstacle avoidance Shortest Path Planning for Wheeled Mobile Robot using Dummy Targets approach to the manipulators and mobile robots based on artificial potential field approach. Initially the method was used for unknown stationary obstacles which further extended for moving obstacles by using time varying artificial potential field. The disadvantage of this method is, it faces problem of local minima in the cluttered environment which can lead to a stable positioning of the robot before reaching the goal. So, this method has local perspective of the robot environment

J. Borenstein [3] gave new concept for obstacle avoidance by using algorithm of combination of Certainty grid for obstacle representation and potential force field for navigation. This method takes care of errors caused by inaccurate sensor data produced by ultrasonic sensors. This allows following continuous motion of the robot without stopping in front of obstacles and solves the local minimum trap problem. The real challenge is to locate and avoid the obstacles in stipulated time for the practical applications of methods so far have been proposed. As such to curb the complexities of heavy computational load, it is possible to implement on low cost, microcontroller-based control structures. For collision Avoidance of non-holonomic single or multi agent systems, in [4] a feedback control law is defined for wheeled mobile robot (WMR) using Lyapunov type analysis to avoid collisions of robot with Stationary obstacles which is further extended to the multi agent system. However, in this, Obstacles are assumed to be circular and of same sizes and shapes, which in reality not. Where as in [5] the author proposed orbital obstacle avoidance algorithm using adaptive limit cycle trajectories. In this, algorithm embedded with well defined control architecture and Lyapunov Synthesis enables the safe and smooth robot navigation in clockwise or counter clockwise direction. Also, the use of limit cycle allows predicting the avoidance region much early and permitting robot to avoid deadlocks or local minima and oscillations. This method helps robot in navigation in less time. But use of limit cycle method makes the algorithm complex. In the present paper, the direction of robot path whether clockwise or counter clockwise is managed by sign of perpendicular drawn from obstacle centre to the line joining dummy Target and Target This paper presents simple algorithm on navigation of robot to the target with the help of two perpendicular axes and dummy targets. These dummy Targets are referred as end points of obstacles or virtual goal in some literatures. In [6] the robot traces both left and right boundaries of the obstacle and chooses the closest one for navigation and to avoid deviation from the target **Kn = min (abs (θck − θT))** is

introduced to direct the robot to the goal. In present paper, misguiding by obstacles left and right boundaries is avoided due to introduction of newly defined dummy target. When there is large obstacle or cluttering of many obstacles and where it exceeds sensors safety length, or if robot faces local minima problem, the obstacle centre co-ordinates are found by taking mean of respective co-ordinates and each time, the dummy targets are relocated as robot moves in tangential direction. Unlike in [7], the present method uses ellipse as safety covering around the obstacle to save robot from the damage. In [8] an area ratio parameter around obstacle is introduced into VFF based approach, whereas present paper introduces ellipse around obstacle which has different area depending on obstacle size and shape.

An artificial potential field method based on Microsoft Kinect sensor is used to measure collision avoidance. Besides this, to determine the shortest paths for transportation tasks, a hybrid planning strategy based on Floyd algorithm and genetic algorithm is used in [9]. Floyd algorithm, gives the reference path based on constrained convex nonlinear optimization which avoids both static and moving obstacles, giving trajectory generation for swarm mobile robots under dynamic environment and local information. This algorithm is predictive method for tracking generated paths so as to reach goal without collisions. Experimental results and numerical simulations proved effectiveness of collision free path planning algorithm. In [10] [11] a parametric approach to trajectory tracking control of robot manipulators is studied where control parameterization method and time scaling transform are used to obtain optimal open loop control. Robust robot navigation using polar coordinates in dynamic environments while avoiding static and dynamic obstacles is presented in [12]. This method employs vector polar histogram (VPH) and Velocity Obstacle method for navigating through dynamic obstacle environment. The method faced difficulties in detecting dynamic obstacles and maneuvering towards the target. Further the method was improvised by combining it with Polar Scan Matching (PSM), which could localize robot's position in better ways that gave improvised detection of dynamic obstacles. Localization is very much important for the mobile robots especially where the performance of the robot is position dependent. Localization is also one of the major problems in obstacle avoidance. In their research work the authors [13] have tried to obtain shortest feasible path in minimum time. Employing Modified Particle Swarm Optimization (MPSO) for mobile robot navigation further increased capability of the optimized algorithms for a global planning. Authors [14] attempted to solve the 2D global path planning problem in a known environment by using a dynamic feedback A* search algorithm and the improved Ant Colony Optimization (ACO) to give shortest path in less time.

2 Problem Statement:

2.1 Robotic environment

The article deals with a typical robotic environment with a set of static obstacles with the objective of determining the shortest path for the robot to reach the target from the start point. Various parameters of the problem at hand are as follows: Robot, obstacles, target are confined to a world frame W, where obstacles are assumed to be 2D, stationary and of different shapes. The target is fixed position and known to the robot. Assuming wheeled mobile robot (WMR) having no proper knowledge of the surrounding environment, it selects and modifies its path towards the goal by ensuring its own safety and safety of surroundings. Wheeled Mobile Robot (WMR) whose position of centre is (xr, yr, θr) is midpoint of mobile vehicle length and is on the axis of direction of the robot, where (xr, yr) are Cartesian co ordinates which gives the state space for the robot and θr is angle of steering of robot with respect to horizontal axis. The world frame W is divided into two spaces obstacle avoidance space and obstacle free space. The WMR is navigated to the target through set of controls [u, ω] where 'u' is linear velocity which is average velocity of right and left wheels of the WMR and 'ω' is angular orientation as shown in figure 1. The robot can move in forward direction and can turn sideways through rotation i.e. motion is non-holonomic. The robot navigation is modeled by Ordinary Differential Equations (ODF):

$$\dot{x}_r = u \cos \theta_r \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

$$\dot{y}_r = u \sin \theta_r \quad \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

$$\dot{\theta}_r = \omega \quad \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

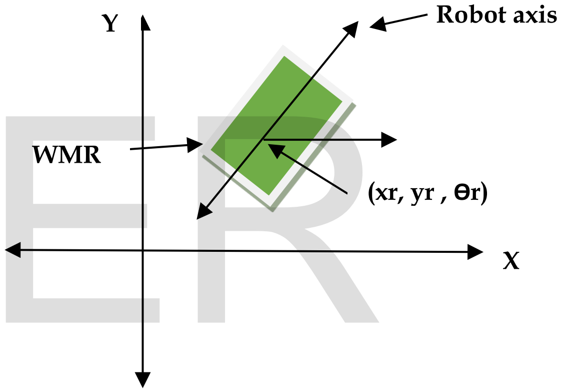

Figure 1: Position of WMR with respect to Cartesian Coordinates

2.2 Mathematically, the parameters are defined as follows: **1.** The Robot start point R = (x0, y0, θ0) Let the robot be enclosed in ball B (R, rR) so that position of robot is safe with respect to active obstacle when it is at distance δ + rR from the obstacle.

2. Target T = (xT, yT, θT) is fixed position and reaching Target (xT, yT) is same as reaching point P where P ∈ B (T, rT) = {P/ | P − T |< rT}

3. Active Obstacle Centre Oi= (Oxi, Oyi).

4. The line joining starting point of Robot R and Target is Reference line RT: lx + my + n = 0.

5. Dummy Targets Di= (xti, yti) are points of intersection of Perpendicular drawn from the active obstacle centre to reference line meeting the ellipse. Dummy Targets are decided after the sign of perpendicular OiPi is determined. Dummy Target Di acts as reference point and direct the robot to the goal. The first dummy Target is assumed to be starting point of robot = (x0, y0) and the last dummy target is Target (xT, yT).

6. The line joining dummy Target Di and Target T is DiT and by default first DiT line is reference line RT as first dummy target is assumed to be starting point of robot = (x0, y0), the subsequent reference lines DiT are obtained for all i >1.

7. Perpendicular from obstacle centre Oi = (Oxi, Oyi) to line RT is OiPi which is extended as the line OiDi to meet dummy target at Di.

8. Safety elliptical embedding around active obstacle is

$$\frac{(x\text{-}Oxi)^2}{\mathbf{ai}^2} + \frac{(y - Oyi)^2}{\mathbf{bi}^2}$$

where' ai 's semi major axis length and 'bi 'is semi minor axis length of the ellipse cover. Where obstacle with centre (Oxi, Oyi) has semi-major axis length (ai −δ) and semi-minor axis length (bi −δ) where 'δ' is safety length. It is safe for the robot to trace the path on ellipse boundary, but not inside ellipse.

3. GENESIS OF THE PROPOSED ALGORITHM:

3.1 Flow Chart: In order to determine the path trajectory for the robotic motion, the procedural logic is as depicted in the flow chart given below:

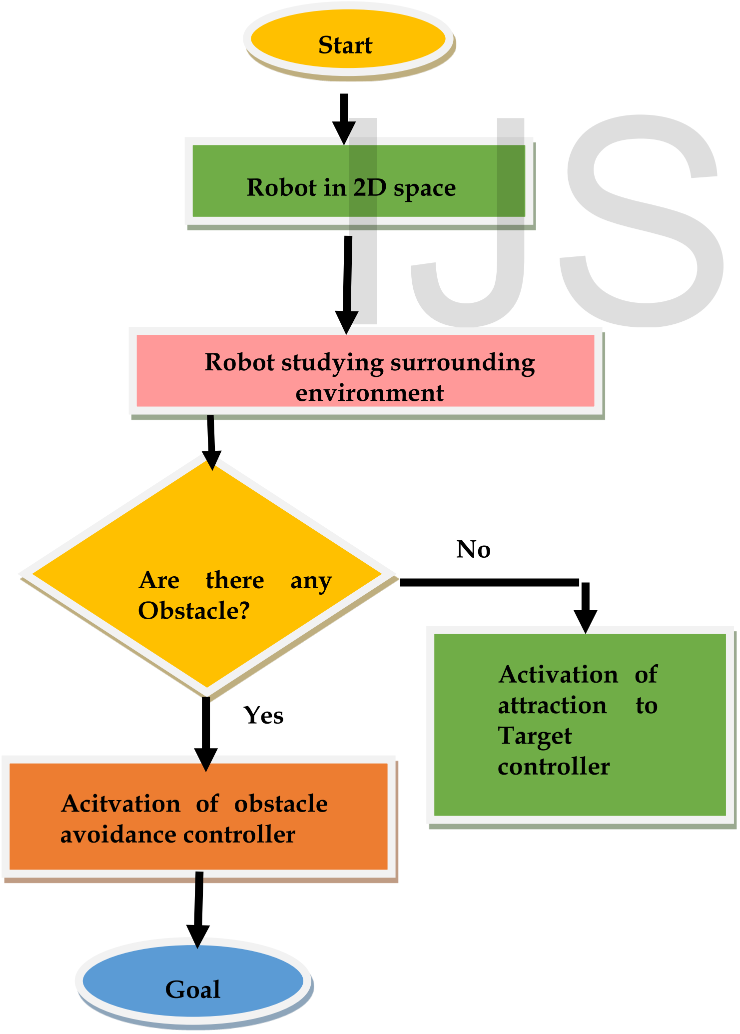

Figure 2: Flow Chart

3.2 Algorithm: The proposed algorithm comprises of various procedural logical steps as outlines herebelow:

1. From the start (x0, y0, θ0) the robot continues to move towards target until it is obstructed by the active obstacle and at a certain distance active obstacle controller is activated.

2. Treat the obstacle intersected by line DiT as 'Active obstacle' and which does not get intersected by DiT as 'Dormant obstacle'.

The ellipse cover around active obstacle is

$$\frac{(x\text{-}Oxi)^2}{\mathbf{ai}^2} + \frac{(y - Oyi)^2}{\mathbf{bi}^2}$$

3. Once the path of robot towards goal is obstructed by obstacle, perpendicular is drawn from the obstacle centre on the path meeting at Pi which extends to meet the ellipse cover of the obstacle at Di . This meeting point is Dummy Target.

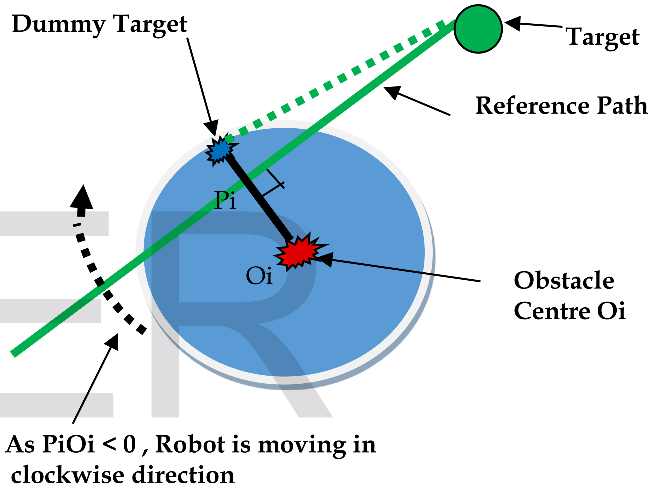

Figure 4: Robot path

4. After dummy target is fixed, the robot will move in direction (clock wise or counter clockwise) of new reference line DiT. Consequently, next dummy target is located until robot gets path free to reach goal.

5. Number of dummy targets will be equal in number as the number of active obstacles unless problem like local minima arises. Under local minima conditions, the robot follows tangential path.

6. If there are 'n' cluttered or overlapping active obstacles with co ordinates (Oxi, Oyi), i = 1, 2……. n then, summing up 'n' obstacles to give total centered position (xOi, yOi) where

$$xOi = 1/n \sum Oxi\ n\ i \text{ and } yOi = 1/n \sum Oyi\ n\ i$$

The process will continue as stated in (3) and (4) above.

7. Orientation errors are defined as follows:

ex = xT − xr …………. (4)
ey = yT – yr …………. (5)
eθ = θT – θr …………. (6)

3.3. Graphical Representation: The above outlined steps are illustrated in Figure. 3

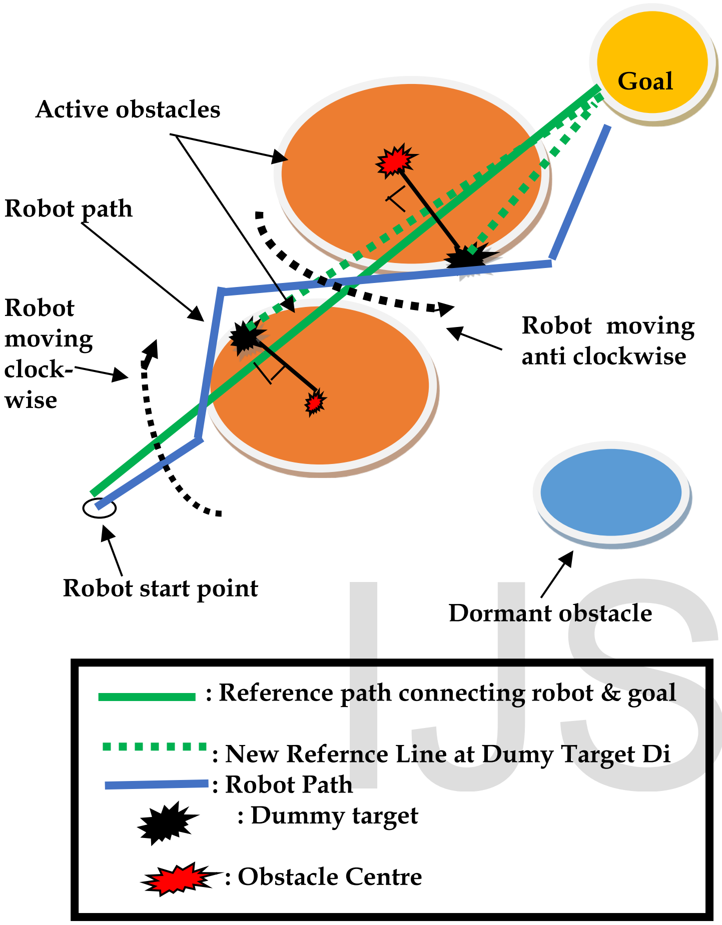

Figure 3: Robot navigation towards the target

3.4 Wall Following Method: Situations involving cluster of obstacles implying wall following method

Wall following method is used in case the obstacles found in close proximity. (figure 5). In wall following method, at a position where robot detects cluster of obstacles after sensing it from motor sensors, wall following controller is activated. When robot encounters first obstacle of the clusters, it finds the two meeting points of perpendicular drawn from obstacle centre to the ellipse covering. The meeting point which makes smaller angle beta with the direction of robot axis is chosen as **dummy target.** This dummy target will move robot in the direction of next dummy target after making minimum angle with previous dummy target. In order to avoid closeness with obstacle, safety angle 'δs' is added in the orientation. This method of selection of dummy targets is iterated till robot comes out of the wall and finds free path to the target.

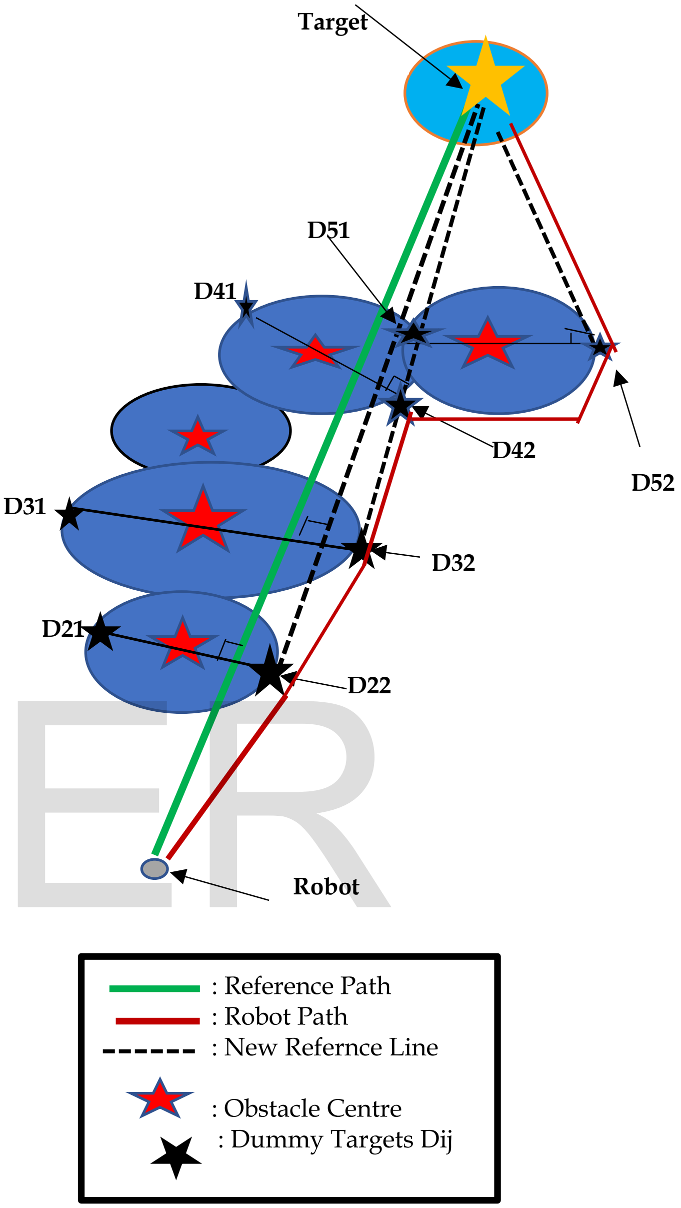

Figure 5: wall following method

4 The Control Strategy of the Robot During the entire navigation of robot to the target, the world frame W is under the influence of Controllers and ***Virtual Force Field*** (VFF). The controller is governed by simple control laws U=[u ω] and robot follows navigation given by system of ODE equations given by equations (1), (2) & (3), which in Matrix form are:

$[\dot{x}r\ \dot{y}r\ \theta.r]$ T = cos. θr 0 sin θr 0 0 1 u ω = P U T (7) 4.1 The force of attraction exerted by Target on robot:

$$[\, xr\ yr\ \ \theta r\ ]^{T} = \begin{pmatrix} \cos\theta r & 0 \\ \sin\theta r & 0 \\ 0 & 1 \end{pmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} \qquad \ldots\ldots\ (7)$$

$$= P\,U^{T}$$

4.1 The force of attraction exerted by Target on robot:

Let r = $\sqrt{(xr - xT)^2} + (yr - yT)^2$ isdistance between current position of robot (xr,yr) & Target (xT, yT) so that force of attraction component in x direction towards target is

$fctx = fa * \cos(\theta r) = fa * (xT - xr)/r$ …….. (8)

Force of attraction component in y direction towards target is

$fcty = fa * \sin(\theta r) = f\,a(yT - yr)/r$ ………. (9)

Where ' fa 'is attraction constant.

4.2. The force of repulsion exerted by obstacle on the robot

It is assumed that there are 'n' active obstacles with centres (Oxi, Oyi) for i = 1 …. n.

Two components of forces of repulsion are exerted by obstacle on robot (xr, yr) at Dummy target Di(xti, yti) are

Force of repulsion in 'x' direction is fcrxi = Pi* (xti-xr) and Force of repulsion in 'y' direction is fcryi = Pi* (yti-yr)

where Pi = fri/ (Dti)^3/2 , 'fri' is repulsion constant and

Dti = √ $(xti - xr)$^ 2+ $(yti - yr)$^ 2 is the distance between Robot position and the next dummy target

. 4.3 Obstacle Avoidance Controller

Resultant Force Acting on the Robot when it encounters obstacles are:

Resultant force in 'x' direction Fx = fctx − ($\sum fcrxi\ n\ i$) ….. (10)

Resultant force in 'y' direction F y = fcty – ($\sum fcryi\ n\ i$) … (11)

Resultant Force Acting on the Robot in obstacle free environment: Resultant force in 'x' direction Fx = fctx

as ($\sum fcrxi\ n\ i = 0$)

Resultant force in 'y' direction F y = fcty

as ($\sum fcryi\ n\ i = 0$)

Where fctx & fcty are given as equations (8) & (9) Resultant orientation angle (beta) $\beta = \tan - 1\ (F\,y/F\,x)$ …………. (12)

The new controls 'u '& 'ω 'defined for robot navigation are as follows: $u = -k1*(\sqrt{Fx}.\ 2 + Fy.\ 2) * \cos(e\theta)$ …………… (13)

$\omega = -k2 * (e\theta)$ …………. (14)

Where k1 > 0, k2 > 0 are control constants. The robot path is demonstrated by simulations in Matlab using ODE45

5. Testing the stability of model using Lyapunov Criteria: Stability of controller defined in above section 4 can be tested using Lyapunov function.

5.1: Construction of Lyapunov function: For the model above, Lyapunov function V be defined as

V = 1/2 (ex.^ 2 +ey^ 2+eθ^ 2) ……………..(15)

where ex, ey, eθ are orientation errors given by equations (4), (5) and (6)

$ex = xT - xr, \quad ey = yT - yr, \quad e\theta = \theta T - \theta r$

As the target is invariable,

$e\dot{x} = x\dot{T} - x\dot{r} \rightarrow e\dot{x} = -\,x\dot{r}\ (x\dot{T} = 0),$

$e\dot{y} = y\dot{T} - y\dot{r} \rightarrow e\dot{y} = -\,y\dot{r}\ (y\dot{T} = 0)$

$\dot{e}\theta = \theta\dot{T} - \theta\dot{r} \rightarrow \dot{e}\theta = -\,\theta\dot{r}\ (\theta\dot{T} = 0)$

Taking time derivative of V in equation (15)

$\boldsymbol{dV\ dt} = 1/2\ (2\ ex\ e\dot{x} + 2\ ey\ e\dot{y} + 2\ e\theta\ \dot{e}\theta)$

$\boldsymbol{dV\ dt} = ex\ e\dot{x} + ey\ e\dot{y} + e\theta\ \dot{e}\theta$

$= ex - x\dot{r} + ey - y\dot{r} + e\theta - \theta\dot{r}$

$= ex\ (-\ u \cos\theta r) + ey\ (-\ u \sin\theta r) + e\theta\ (-\omega)$

$= ex\ (-u \cos(\theta T - e\theta)) + ey\ (-u \sin(\theta T - e\theta)) + e\theta\ (-\omega)$

$= ex\ (-u)\ (\cos\theta T \cos e\theta + \sin\theta T \sin e\theta) +$
$ey\ (-u)\ (\sin\theta T \cos e\theta - \cos\theta T \sin e\theta) + e\theta * (-\omega)$

Let D= $\sqrt{Fx}$^2+ Fy^ 2

Substituting u= −k1 cos(eθ) and ω = −k2 eθ,

cos(eθ) = $Fx\ D$ and sin(eθ) = $Fy\ D$ so that

$\boldsymbol{dV\ dt} = ex\ (k1 \cos(e\theta)\ D)\ [\cos(e\theta)\ Fx\ D + \sin(e\theta)\ Fy\ D] +$
$ey\ (k1 \cos(e\theta)\ D)\ [\sin(e\theta)\ Fx\ D - \cos(e\theta)\ Fy\ D] + k2\ e\theta$^ 2

$= k1 \cos 2\ (e\theta)\ (Fx\ ex + Fy\ ey) +$
$k1 \sin(e\theta) \cos(e\theta)\ (Fy\ ex - Fx\ ey) + k2\ e\theta$ ^2 …………(16)

As t → ∞ , xti→ xT and (xti – xr) → xT –xr = ex yti→ yT and (yti –yr) → yT –yr = ey & distance DiT = r

therefore

$$Fx = \frac{ex}{r^{3/2}}\ [fa.\ r^{1/2} - fr\ ]$$

$$Fy = \frac{ey}{r^{3/2}}\ [fa\ r^{1/2} - fr\ ] \quad \text{and}$$

Fxey =Fyex

so equation (16) is

$\boldsymbol{dV\ dt} = k1 \cos^2(e\theta)(\ fa.\ r^{1/2} - fr)\ (ex^2 + ey^2) + k2\ e\theta^2$

The given system of equations 1),2),3) is asymptotically stable if

$dV\ dt < 0$ i.e.

if $k1 \cos^2(e\theta)(\ fa.\ r^{1/2} - fr)\ (ex^2 + ey^2) + k2\ e\theta^2 < 0$

If $k2\ e\theta^2 < k1 \cos^2(e\theta)(\ fr - fa.\ r^{1/2})\ (ex^2 + ey^2)$

Stability is achieved by making repulsion force constant 'fr' relatively larger than attraction force Constant fa .

Also k2 is made relatively very small than k1 .

all the parameters are maintained within the range to achieve stability as per Table VI

6. Testing controllability of Model: The matrix form of system of equations (1),(2) & (3) is

$$[\,x_r\ y_r\ \ \theta_r\,]^T = P\,U^T \quad \text{where}$$

$$P = \begin{pmatrix} \cos(\theta r) & 0 \\ \sin(\theta r) & 0 \\ 0 & 1 \end{pmatrix} \quad \& \quad U = [\,u\ \omega\,]$$

$$[\,x_r\ y_r\ \ \theta_r\,]^T = \begin{pmatrix} \cos(\theta r) & 0 \\ \sin(\theta r) & 0 \\ 0 & 1 \end{pmatrix} * u + \omega \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Let g1 = $[\cos(\theta r)\ \ \sin(\theta r)\ \ 0\,]^T$, g2= $[0\ 0\ 1]^T$ and

Let z= $[\,xr\ yr\ \theta r\,]$

Where vector field g1 generates forward backward motion and vector field g2 generates clockwise and counter clockwise motion.

Lie bracket [g1 g2] generates motion in the direction perpendicular to the orientation of WMR.

The lie bracket [g1 g2] is

$$[g1\ g2] = \frac{\partial g2}{\partial z} * g1 - \frac{\partial g1}{\partial z} * g2$$

$$[g1\ g2] = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \cos(\theta r) \\ \sin(\theta r) \\ 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 & -\sin(\theta r) \\ 0 & 0 & \cos(\theta r) \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} \sin(\theta r) \\ -\cos(\theta r) \\ 0 \end{pmatrix}$$

Now rank of g1,g2 &[g1 g2] = rank of $\begin{pmatrix} \cos(\theta r) & 0 & \sin(\theta r) \\ \sin(\theta r) & 0 & -\cos(\theta r) \\ 0 & 1 & 0 \end{pmatrix}$

The system of equations 1) ,2) ,3) is controllable
If rank of{ g1 ,g2 [g1 g2]}= 3

As determinant of [g1 ,g2 [g1 g2]] = 1 (≠ 0) ,
Rank of [g1 g2 [g1 g2]] =3,
it proves that the system is controllable

7. Result through simulations: For the purpose of evaluating the efficiency of the proposed path formation methodology, the simulation trials are conducted using MATLAB. Four different situations are tried as models 1 to 4. Figure 5, 6, 7 and 8 show the graphical output of the simulation trials. Tables I, II, III and IV show the coordinates of various salient intermittent positions and the segmental and total path lengths traced by the robot.

7.1: Model 1

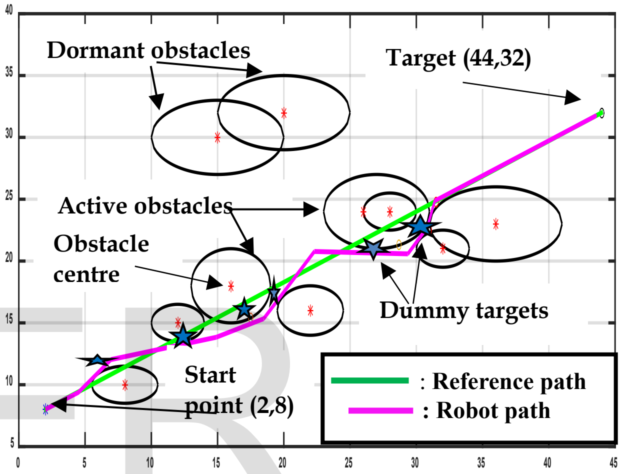

Figure 5: Model 1-Robot navigation towards goal using dummy targets

Sr No.	Initial conditions	Dummy targets	Path length
1	[2,8,pi/2]	(2,8)(start oint)	2.8528
2	[4.4974, 9.3790, 1.5708]	(7.18,11.42)	0.0023
3	[4.4961, 9.3771, 1.9635]	(12.77,13.61)	3.6202
4	[6.9567, 12.0325 ,1.9733]	(17.53,15.41)	8.1434
5	[14.8978, 13.8365,1.9743]	(20.87,17.28)	3.8871
6	[18.4826,15.3395 ,1.9842]	(28.68,21.28)	6.6440
7	[22.3061, 20.7731,1.9941]	(32,22.5)	7.0578
8	[29.3613 ,20.5818, 2.0041]	(31, 24.8)	2.5923
9	[30.8278 ,22.7194, 2.0060]	---	2.2134
			14.4907
	Total Path length		**51.5040**

Table I: Model 1 simulation result

7.2: Model 2

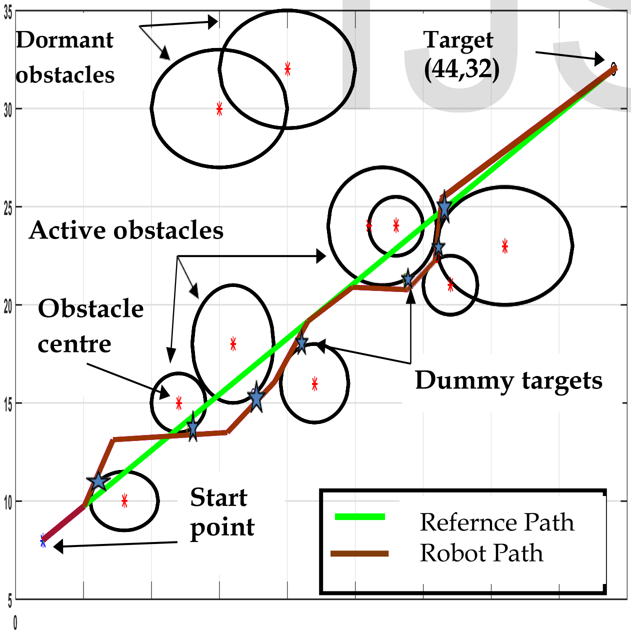

Figure 6: Model 2-Robot navigation towards goal using dummy targets

Sr	Initial conditions for	Dummy tar-	Path lengths
1	[2,8, pi/2]	(2,8) (start	3.50
2	[5.0389, 9.7365,	(7.18,11.42)	0.0099
3	[5.0488, 9.7367,	(12.77,13.61)	4.0049
4	[7.1768, 13.1295,	(17.53,15.41)	8.4186
5	[15.5873, 13.498,	(20.87,17.28)	4.3240
6	[19.0613, 16.072,	(28.68,21.28)	3.9460
7	[21.5102, 19.1670,	(31, 24.8)	4.0758
8	[24.7867, 20.9003,	(32,22.5)	3.7067
9	[28.8605, 20.7737,	----	2.4453
10	[30.8247, 22.2302,	----	3.2298
11	[31.3252, 25.4210,	----	14.2805
Total path length			**51.9415**

Table II: Model2 simulation result

7.3: Model 3

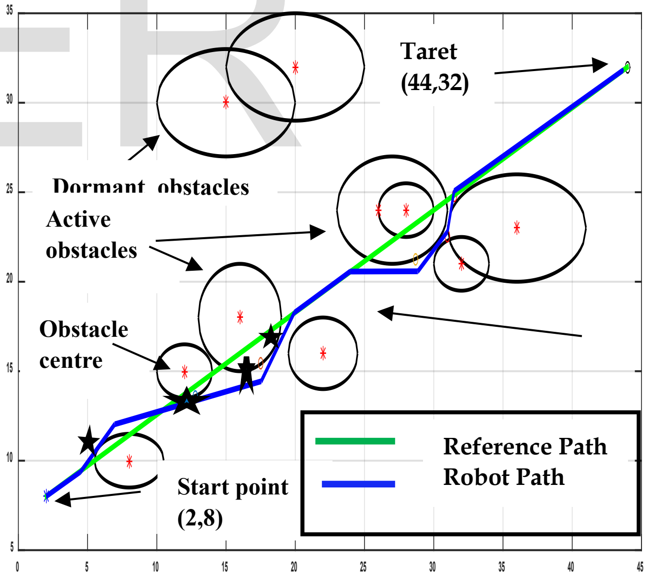

Figure 7: Model 3-Robot navigation towards goal using dummy targets

Sr N	Initial conditions for ODE45 for model 3	Dummy targets	Path length
1	[2,8, pi/2]	(2,8) (start point)	2.8528
2	[4.4974 9.3790 1.5708]	(7.18,11.42)	0.0023
3	[4.4961 9.3771 1.9635]	(12.77,13.61)	3.6202
4	[6.9567 12.0325 1.9733]	(17.53,15.41)	10.8551
5	[17.5422 14.4366 1.9743];	(20.87,17.28)	4.5108
6	[19.9225 18.2682 1.9842]	(28.68,21.28)	4.6442
7	[23.9567 20.5690 1.9922]	(32,22.5)	4.8934
8*	[28.8501 20.5814 2.0022]	(31, 24.8)	3.0945
9	[30.9968 22.8102 2.0043]	---	2.3030
10	[31.4679 25.0645 2.0143]	-----	14.3232
		Total path length	**51.0995**

TableIII: Model 3 Robot Navigation simulation result

7.4: Model 4

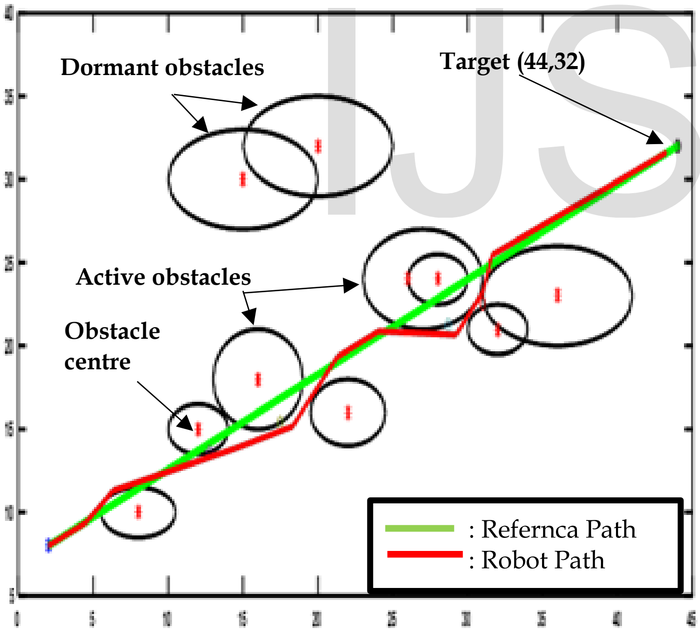

Figure 8: Model 4-Robot navigation towards goal using dummy targets

Sr N	Initial conditions for ODE45	Dummy targets	Path length
1	[2, 8, pi/2]	(2,8)	2.8354
2	[4.4749, 9.3837, 1.5694]	(7.18,11.42)	0.0055
3	[4.4787, 9.3877, 1.5792]	(12.77,13.61)	2.9954
4	[6.5403, 11.5608, 1.5871]	(17.53,15.41)	12.4009
5	[18.4715, 14.9415, 1.5887]	(20.87,17.28)	3.7649
6	[20.3750, 18.1898, 1.5966]	(28.68,21.28)	4.6442
7	[24.3844, 20.5335, 1.6046]	(32,22.5)	4.7340
8	[29.0812, 21.1259, 1.6126]	(31, 24.8)	1.9962
9	[30.6112, 22.4080, 1.6128]	---	3.4063
10	[31.8342, 25.5872, 1.6209]	-----	13.7525
		Total path length	**50.5353**

Table IV: Model 4 robot navigation simulation result

7.5 Wall following method environment.

Table V gives the numerical values of the parameters involved.

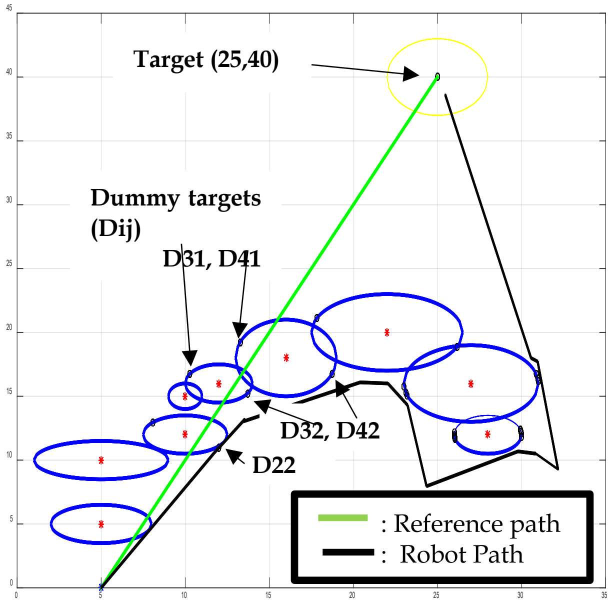

Figure 9: wall following method

Sr.No	Initial conditions for ODE45	Selected dummy target
1	[5; 0; pi/6]; (start point)	**(xt_1, yt_1)** = (5,0)
2	[11.7204, 10.5607, 0.2678]	**(xt_2, yt_{22})** = (12,11)
3	[13.5330,13.1841, -1.9946];	**(xt_{32}, yt_{32})** = (13.716,
4	[13.4539, 13.0278, -1.9157]	**(xt_{42}, yt_{42})** = (18.73,
5	[20.3976, 16.088, 18.0979];	**(xt_{52}, yt_{52})** = (26.1714,
6	[22.0516, 16.005, 19.7540]	**(xt62, yt62)** = (23.01,
7	[23.2227,14.3042, 0.8721];	**(xt71, yt71)** =
8	[24.3542, 7.9553, 0.8446];	**(xt81, yt81)** = (29.94,
9	[29.7896, 10.6827, -1.5909];	**(xt92, yt92)** = (26.05,
10	[30.852, 10.5065, -1.3178];	**(xt101, yt101)** =
11	[32.1726, 9.2786, -1.3749];	**(xt112, yt112)** =
12	[30.9599, 17.7174, 0.7114];	**(xt122, yt122)** =
13	[30.5668, 17.9965, -1.3094];	**(xT, yT)** = (25,40)

Table V: simulation result (wall following method)

8. Validation of the proposed methodology:

In order to ascertain the performance of the proposed methodology, a performance index is defined as ratio of length of path determined by the proposed methodology to the reference path length, i.e. path without encountering any obstacles.

Performance Index (PI)

$$= \frac{\text{length of path obtained by simulation}}{\text{Reference path length}}$$

Ideally, the PI value as close as possible to 1 is desirable.

For the purpose of determining the performance measure of the ***proposed methodology***, simulation trials are conducted.

In the typical situation analyzed, the Start point coordinates are (2, 8) and the Target point coordinates are (44, 32). Consequently, the ***Reference Path length*** is calculated to be **48.3735** units.

The length of the path obtained for **four different sets** of robotic environments, the paths are devised using MATLAB ODE45 utility. The parameters K1, K2, Fa and Fr are varied.

The results are as shown in the following table:

Figure No.	Parameters range				Path length	Performance index	Time (s)
	K1	**K2**	**Fa**	**Fr**			
Model1	68.5	0.49	6.999	48	51.504	1.06	3.90
Model2	87	0.49	6.999	58	51.941	1.07	3.640
Model 3	88.5	0.4999	7.999	47	51.099	1.05	4.6235
Model 4	90	0.49	6.99	78	50.5353	**1.04**	3.90
						(optimum)	

Table VI: parameter range, path lengths and performance index

For the purpose of validation, the PI arrived at by the proposed methodology, is compared with the similar Index obtained by **MPSO method,** employed in [13], where Start point coordinates are (1,1), Target point (22,30) and Reference Path length is 35.805 units. PI value of 1.04 in case of the proposed method is certainly found to be superior to the PI value in case of MPSO method.

Figure No	Path length	Performance index
6	39.4558	**1.10**
7	39.4558	
8	39.4558	
9	39.4558	

Table VII: path lengths and performance index by MPSO method [13].

Further, the PI arrived at by **ACO method,** as in [14] is also reviewed. Here, the PI values of both the methods are comparable and, in a way, validates the proposed methodology.

Algorithm	Shortest Path length	Mean path length	Performance Index
Original ACO	49.66	51.42	1.04
Simplified A* ACO	43.91	45.82	1.04

Table VIII: path lengths and performance index in [14]

9. Conclusion The robotic application environment dealt with in this research work, basically comprises of 2D static obstacles of varying sizes, assumed to be encompassed in elliptical envelope. As compared to the limit-cycle method of path formation, the proposed methodology of introducing intermittent dummy targets for formation of straight-line segments of the shortest path to be traced by the robot is found to be efficient. However, in case of wall formation type of problems, despite the proposed strategy of angular preferential guiding factor for clockwise or anticlockwise directional progression of robot around the obstacles, the local minima effect plays a dominant role and the methodology does not guarantee shortest path length.

Simulation trials carried out using MATLAB, by way of solving system of three non-linear ordinary differential equations representing the path, using ODE45, have adequately validated the proposed methodology. Parameters k1, k2, fct, fcr appearing in the equations are randomly selected and a suitable Lyapunov function is devised to check the stability, which works well for present system. Also, controllability of the system is tested through lie bracket derivatives.

There lies a tremendous scope for further investigation of the methodology for the applications in the dynamic robotic environment involving obstacles moving with different velocities.

10.Declarations :

Conflict of Interests The authors declare that they have no conflict of interest for this work.

References 1. Borenstein, J. and Koren, Y., Optimal Path Algorithms for Autonomous Vehicles Proceedings of the 18th CIRP Manufacturing Systems Seminar, (1996)

2. Khatib, O., Real-Time Obstacle Avoidance for Manipulators and Mobile Robots IEEE International Conference on Robotics and Automation, 1985, 500-505.

3. J. Borenstein, and Y. Koren, Real-time Obstacle Avoidance for Fast Mobile Robots IEEE Transactions on systems, Man, and Cybernetics, 1989, 1179-1187.

4. Mastellone, S., Stipanovi, D.M., Graunke, C.R., Intlekofer, K.A. and Spong, M.W., Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments The International Journal of Robotics Research, 2008, 107-126.

5. Olunloyo, V.O.S. and Ayomoh, M.K.O., Autonomous mobile robot navigation using hybrid virtual force field concept European Journal of Scientific Research, 31(2), 2009, 204-228.

6. Kazem, Bahaa I and Hamad, Ali H and Mozael, Mustafa M, Modified Vector Field Histogram with a Neural Network Learning Model for Mobile Robot Path Planning and Obstacle Avoidance., Int. J. Adv. Comp. Techn.,2(5), 2010, 166-173.

7. Ferreira, Andr´e and Pereira, Fl´avio Garcia and Vassallo, Raquel Frizera and Bastos Filho, Teodiano Freire and Sarcinelli Filho, M´ario, an approach to avoid obstacles in mobile robot navigation: the tangential escape, Sba: Controle & Automa¸c˜ao Sociedade Brasileira de Automatica, 19(4), 2008, 395-405.

8. Ni, Jianjun and Wu, Wenbo and Shen, Jinrong and Fan, Xinnan, An improved VFF approach for robot path planning in unknown and dynamic environments, Mathematical Problems in Engineering,2014, Hindawi.

9. Ramdane Hedjar and Messaoud Bounkhel, Real-time obstacle avoidance for a swarm of autonomous mobile robots, International Journal of Advanced Robotic Systems, 11(4), 2014.

10. Shijie Zhang and Yi Ning, Parametric approach to trajectory tracking control of robot manipulators Journal of Applied Mathematics, 2013, 2013.

11. Liu Chengqing, Marcelo H Ang, Hariharan Krishnan, and Lim Ser Yong, Virtual obstacle concept for local-minimum-recovery in potential eld based navigation, In Robotics and Automation proceedings, ICRA'00 IEEE International Conference, volume 2, pages 983-988, 2000.

12. Seung-Hwan Lee, GyuhoEoh, and Beom H Lee, Robust robot navigation using polar coordinates in dynamic environments, Journal of Industrial and Intelligent Information Vol, 2(1), 2014.

13. Optimal Path Planning for Intelligent Mobile Robot Navigation Using Modified Particle Swarm Optimization
Nadia Adnan Shiltagh, Lana Dalawr Jalal, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-2, Issue-4, April 2013.

14. Design and Simulation of a Collision-free Path Planning Algorithm for Mobile Robots Based on Improved Ant Colony Optimization Lu Bai*, Chenglie Du School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China.