Behavior Based Co-ordination of a Troop of Vehicles Targeted to Different Goals in an Unknown Environment

Submitted 26th January 2012; accepted 16th August 2012

Sourish Sanyal, Ranjit Kumar Barai, Rupendranath Chakrabarti, Pranab Kumar Chattopadhyay

Abstract:
The issue of coordinated operation of multi-vehicle for a variety of tasks is getting increasing attention day by day and standing as a major research field due to their increased capacity and flexibility they can offer as a team. This paper presents a novel algorithm for multi-vehicle navigation, based on exhaustive search to avoid a set of randomly generated obstacles, predict the approximate position of other vehicles and thus keeping a safe distance to avoid collision and to maintain a formation amongst them while targeted towards the assigned goals. The proposed algorithm uses two optimizing functions in deriving drive commands, direction and turning, for a troop of vehicles. This particular algorithm is similar to the artificial potential field (APF) method which is widely used for autonomous mobile robot path planning due to its simplicity and mathematical elegance. In this work we have taken a behavior based reactive scheme together with artificially generated perturbation as the vehicles are running in a real time environment. Simulations have been carried out for a group of four vehicles, paired in two groups, approaching two different targets avoiding eight randomly generated obstacles, and keeping proper coordination between the members of intra and inter groups. The effectiveness of the proposed approach has been shown by some simulation results.

Keywords: *behavior-based collision avoidance, randomized obstacles, multi-vehicle coordination, particle swarm optimization.*

1. Introduction

The challenge that a troop of multiple uninhabited autonomous vehicles (UAVs) would be able to adaptively react to their environment, whether known, unknown or uncertain, and learn about their surroundings while following either an individual or a communal agenda is an intriguing field of research. Achieving such a degree of control and producing such sophisticated behavior remains an elusive goal that demands considerable attention and this is inherently a complex task. The problem of multi-vehicle coordination and control has been receiving an exquisite amount of attention during the past few years due to critical importance of the field in wide-ranging applications [8].

In many practical applications of autonomous vehicles multiple teams are to be used. Such teams have many potential benefits, including faster completion through parallelism and increased robustness through redundancy. Further, teams of vehicles can increase the application domain of autonomous vehicles by providing solutions to tasks that are inherently distributed, either in time, or in space, or in functionality. Since the 1980s, researchers have addressed many issues in multi-vehicle, or multi-robot teams or automated guided vehicles (AGVs) [12], such as control architectures, communication, task allocation, swarm robots, learning [25]. A critical issue in these mobile robot teams is coordinating the motions of multiple vehicles interacting in the same workspace. Regardless of the mission of the vehicles, they must be able to effectively share the workspace to prevent interference between the team members. Solutions to the motion coordination problem are approached in a variety of ways, depending upon the underlying objectives of the vehicle team. In some cases, the paths of the robots are explicitly planned and coordinated in advance, as might be needed in a busy warehouse management application. In other cases, planning is relaxed and emphasis is placed on mechanisms to avoid collision, applicable for tasks such as automated hospital meal deliveries. In yet other situations, the robots could have mechanisms with little pre-planning that focus on coordinating vehicle motions in real-time using reactive, behavior-based, or control-theoretic approaches, such as would be used in a convoying or formation-keeping application.

Existing work on multi-vehicle control focuses receding-horizon planning (an optimization method) and hierarchical structures. The receding-horizon trajectory planner based on Mixed Integer-Linear Programming (MILP) is capable of planning planner-based trajectories directed to a goal [14,15,16]. The goal is constrained by no-fly areas, or obstacles, and is free from leader-follower architecture which is adopted by model predictive control (MPC) [17]. Game-theoretic approach is also adopted by different co-ordination schemes for decision making of the multi-vehicle problem [18,19,20]. A disjoint path algorithm for a reconfiguration of multi-vehicle was also proposed [21]. A class of triangulated graphs for algebraic representation of formations have been introduced to specify a mission cost for a group of vehicles [22]. The present work focuses on simultaneous movement of a troop of vehicles from their initial locations towards different targets in such an environment where obstacles are generating stochastically based on the Artificial Potential Field (APF) approach. The basic idea of the APF approach is to fill the robot's workspace with an artificial potential field in which the robot is attracted to its target position and is repulsed away from the obstacles [4]. This method is particularly attractive because of its elegant mathematical analysis and simplicity. The application of

APF for obstacle avoidance was first developed by Khatib [3]. In the past decade this method has been studied extensively for autonomous mobile robot path planning by many researchers [5-7]. This is a new approach where the troops are divided into two groups and set out for their own targets, maintaining a formation amongst them. This work is an extension of the work done by Kevin Passino [2] on obstacle avoidance of a single vehicle in presence of a number of fixed obstacles.

2. Problem description

A. Cooperation of multi-vehicles

The word cooperation means interaction or integration of multiple vehicles [11]. In a cooperative team the vehicles have to communicate, exchange information or interact in some way to achieve an overall mission. The term cooperation has been widely discussed in different scientific community and different definitions have been proposed.

B. Multi-vehicle path planning problem

It is defined as follows: given a set of **m** vehicles in **k**-dimensional workspace, each specified with an initial starting configuration (e.g., position and orientation) and a desired goal configuration, determine the path each vehicle should take to reach its goal, while avoiding collisions with obstacles and other vehicles in the workspace. More formally, let A be a rigid vehicle in a static workspace W=$\mathbb{R}^k$[18,19], where k= 2 or k = 3. The workspace is populated with obstacles. A *configuration* q is a complete specification of the location of every point on the robot geometry. The *configuration space* C represents the set of all the possible configurations of A with respect to W. Let O $\subset$ W represent the region within the workspace populated by obstacles. Let the close set A(q) $\subset$W denote the set of points occupied by the vehicle when it is in the configuration q$\in$C. Then, the *C-space obstacle region*, C_{obs}, is defined as [1]:

$$C_{obs} = \{q \in C \mid A(q) \cap o \neq \Phi\ \} \qquad (1)$$

The set of configurations that avoid collision (called the *free space*) is:

$$C_{free} = C \setminus C_{obs}\,. \qquad (2)$$

A *free path* between two obstacle-free configurations C_{init} and C_{goal} is a continuous map:

$$\tau[0,1] \to C_{free} \qquad (3)$$

such that $\tau(0) = c_{init}$ and $\tau(1) = c_{goal}$.

For a team of m vehicles, define a state space that considers the configurations of all the robots simultaneously:

$$X = C^1 \times C^2 \times \ldots \times C^m\,. \qquad (4)$$

Note that the dimension of X is N, where $N = \sum_{i=1}^{m} \dim(C^i)$The C-space obstacle region must now be redefined as a combination of the configurations leading to a robot-obstacle collision, together with the configurations leading to vehicle to vehicle collision. The subset of X corresponding to robot A^i with the obstacle region O, is

$$X^i_{obs} = \{x \in X \mid A^i(q^i) \neq \Phi\} \qquad (5)$$

The subset of X corresponding to robot Ai in collision with robot A^j is

$$X^{ij}_{obs} = \{x \in X \mid A^i(q^i) \cap A^j(q^i) \neq \Phi\} \qquad (6)$$

The obstacle region in X is then defined as the combination of Equations (5) and (6), resulting in

$$X_{obs} = (\bigcup_{i=1}^{m} X^i_{obs}) \cup (\bigcup_{ij,i \neq j} X^i_{obs}) \qquad (7)$$

With these definitions, the planning process for multi-vehicle system treats X the same as C, and X_{obs} the same as C_{obs} ,where C_{init} represents the starting configuration of all the robots, and C_{goal} represents the desired goal configurations of all the vehicles.

The APF uses two types of potential field, namely a repulsive potential field to force a robot away from obstacles or forbidden regions and an attractive potential field to drive the robot to its goal. The robot moves under the action of a force that is equal to the negative gradient of that potential, and it is driven towards the positions with the lower potential.

In this paper, we consider the robot as one particle that moves under the action of the composition of forces $\vec{A}_r$, which is the summation of goal's attractive force $\vec{F}_{rg}$ and the obstacle's repulsive force $\vec{F}_{or}$ as shown in Fig. 1.

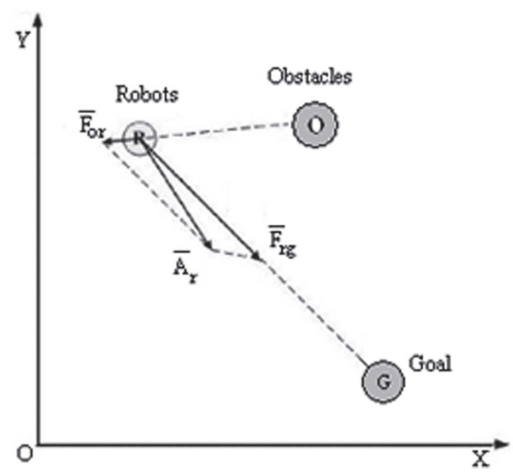

Fig. 1. Virtual attractive force of robot in APF

Typically, optimization criteria guide the choice of a particular solution from an infinite number of possible solutions. Examples are: minimal path lengths (local or global), minimal time to reach targets, and minimal energy consumption to reach the goal. Presence of constraints brings forth more complexity. Such constraints arise from navigational restrictions e.g. limitation on the maximum angle of rotation, restrictions on maximum slope, inability to traverse rocky terrain, etc. or the need for a vehicles to move in tandem. Since general optimal solution for multiple moving objects is computationally difficult, sometimes intractable [24], local optima is sought for instead of global optima in the path planning problem.

3. Problem formulation and the proposed scheme

For convenience and for reducing complexity, the scope of the present work has been limited to the motions in only 2-D space. It is assumed that the obstacles are not dynamic but they are randomly generated in the workspace. No vehicle is stationary with respect to another. So it is obligatory to keep a safe distance between mobile vehicles to avoid collision and maintain a formation. It is assumed that there are **m** no of vehicles and the $\mathbf{i_{th}}$ one follows a discrete time kinematic model given as:

$$x_{v1}^{i}(k+1) = x_{v1}^{i}(k) + d\cos(\theta_v^i(k)) \quad (8)$$

$$x_{v2}^{i}(k+1) = x_{v2}^{i}(k) + d\sin(\theta_v^i(k)) \quad (9)$$

$$\theta_v^i(k+1) = \theta_v^i(k) + f\theta_v(u^i(k)) \quad (10)$$

where **k** is the discrete time index taking values of non-negative integers {0,1,2,3,….}(in the present problem k denotes the number of search steps); θ_v^i is the orientation of the i_{th} vehicle; $f\theta_v$ can be a nonlinear function encoding kinematic restrictions on the vehicles; u^i is the local controller corresponding to i_{th} vehicle. For convenience, let

$$x_{v_p}^i = [x_{v_1}^i, x_{v_2}^i]^T \text{, and } x_v^i = [(x_{v_p}^i)^T, \theta_v^i]^T . \quad (11)$$

It has been assumed that the controller has prior access to the information on randomly generated obstacles but not to the vehicles. The vehicles are to communicate with the controller (distributed controllers, dedicated one for each vehicle like an embedded system) to update the information on their positions at every iteration before taking the next move. The environment is modeled as a 2D, plane, having four quadrants (upper right and left and lower right and left) of a Cartesian coordinate system with axes (x_1,x_2). A Gaussian profile map has been set up which is accessible to all the vehicles through their controllers. It encodes the possible obstacle locations $x_s^i = [x_{s1}^i, x_{s2}^i]^T$, i =1,2,…n obtained from sensory data which act as centers of the Gaussian peaks. It is assumed that the number of the obstacles is n (n=8 for this case).

Considering initial position to be $[x_1,x_2]$, the mathematical description is as given below:

$$M_p(x_1,x_2,k=0) = \sum_{i=1}^{n} c_i \exp - \left[\frac{(x_1 - x_{s_1}^i)^2 + (x_2 - x_{s_2}^i)}{v_i^2} \right] \quad (12)$$

There may be some uncertainty in the data for the distances measured by the sensors. The uncertainty can be encoded with variation in v_i . Then uncertainty of prior information having a peak width of v_i and the distances of the real obstacles from the centre of the peak in terms of v_i may be clubbed together. Furthermore, a specific priority can be assigned intentionally to a particular task by assigning different values as weights to c_i . In this approach, all the vehicles share the common map M_p (x_1,x_2) at every iteration. The vehicle (controller) sensor samples the Cartesian plane to get information on updated positions of obstacles and other vehicles and derive the drive command. The output is in the form of binary i.e. an output of 0 means no obstacle or and an output of 1 means an obstacle in near proximity.

$$M_t(x_1,x_2) = 1, \ [x_1,x_2]^T \in \left\{ \left[x_{t_1}^i, x_{t_2}^i \right]^T \right\}$$
$$= 0 \text{ otherwise} \quad (13)$$

Emphasis has been given on moving the vehicles in discrete steps as if moving from cell to cell rather than moving along a smooth curve. Random velocities have been assigned to the vehicles. No restriction has been imposed on maximum angle of rotation in one step but in reality a sharp turn may adversely affect the stability of the vehicles. This problem can be redressed by slowing down the vehicles.

For most of the practical situations, a vehicle located somewhere in the terrain is unable to locate all the obstacles and other vehicles at a time. In order to account for this inability, an artificial perturbation has been added to the output vector [9, 10]. The problem can now be formulated by slightly modifying the above-mentioned kinematic problem:

$$\begin{bmatrix} x_{v1}^i(k+1) \\ x_{v2}^i(k+1) \\ \theta_v^i(k+1) \end{bmatrix} = \begin{bmatrix} x_{v1}^i(k) + \lambda\cos(\theta_v^i(k) + T\beta_k^i) \\ x_{v2}^i(k) + \lambda\sin(\theta_v^i(k) + T\beta_k^i) \\ \theta_v^i(k) + T\beta_k^i \end{bmatrix} \neq \quad (14)$$

The nonlinear function f of the previous model has been reduced to a linear incremental function having a step increment in sample time T and λ is the minimum incremental distance that any vehicle traverses before scanning its world map for the next time slot, β_k^i is the steering angle and $\theta_v^i(k)$ is the orientation with respect to X-axis for the last update.

So the concise form of the model for positional and angular updating is given as:

$$\left[x_{v1}^i(k+1), x_{v2}^i(k+1) \right] = \Gamma\left\{ \left[x_{v1}^i(k+1), x_{v2}^i(k+1) \right], u_k^i \right\} \quad (15)$$

where x_{v1}^i (k+1) and x_{v2}^i (k+1) are the updated states of X-Y coordinates in the Cartesian coordinate system at time t for the of the i_{th} vehicle and u_k^i is the drive commands generated by the controller – the minimal positional and steering rate update at T_{th} discrete sample time, where Γ is the mapping function. The goal of vehicle coordination is to derive a sequence of controls for each vehicle i.e.

$$u^i = \left\{ u_0^i, u_1^i, \dots, u_k^i \right\} \quad (16a)$$

such that the trajectories are:

$$x^i = \left\{ x_0^i, x_1^i, \dots x_n^i \right\} \quad (16b)$$

4. The real-time problem taken for path optimization

The specific scenario and the considerations behind experimentations and finding out the simulation results are worth mentioning at this point. Four vehicles paired in two different groups have been considered – they have set out for two different targets of same preference.

- There are eight obstacles randomly generated in the workspace whose locations can be traced out by the sensors of the distributed controlling mechanism of each individual vehicle.

- Each vehicle can sense their present location in Cartesian X-Y coordinate system while- they also have prior information about their starting locations.
- The controller of each vehicle can communicate with that of other vehicles and can distinguish between a moving object and a static obstacle.

Against these considerations in the backdrop, the real time algorithm used in this experimentation has been framed. Three functions have been taken: one for obstacle generation, a Gaussian profile function to get estimate of the obstacle positions and a goal function to find out the best possible position and orientation for gradually getting nearer to the goal.

4.1. The proposed algorithm

Three functions have been used in this multi-vehicle path planning viz. 'obstaclegeneration', 'obstaclefunc', and 'goalfunc' along with the main program.

Main Algorithm

Step 1: Segregate the 360^o contour of the robots world map into N no. of segments.

Step 2: Do while count<preset value

Step 3: Loop for K=1 to N (incrementing step angle), calculate the Gaussian profile obstacle distance from each point of the circular trajectory by calling the function 'obstaclefunction' and find the furthest obstacle distances from each point and follow the same for finding the Euclidian distance to goal from all of those points by calling the function 'goalfunction'.

Step 4: Add the return array functions so as to treat this function as a composite one.

Step 5: Minimize the composite function so as to get the best angle to move.

Step 6: Orient the robot towards that best found direction and move minimum incremental distance i.e. proceeding cell by cell (as if repulsed by the random obstacles and attracted towards the goal).

Step 7: Repeat step 2 to step 5 for all subsequent robots.

Reactive Behavior Scheme:

Step 8: Loop for i = 1 to 3

Step 9: While $Robot_{i_{x,y}}$ or $Robot_{i+1_{x,y}} \neq Goal_{x,y}$ do

Step 10: If $Robot_{1_{x,y}} - Robot_{2_{x,y}} \leq$ *predefined thresold* give either an X-axis shift or Y-axis shift accordingly.

Step 11: Endwhile

Step 12: Endfor

Add Artificial Perturbation

Step 13: generate delta-increment and delta-angle by random function generation

Step 14: Add them with goal function and obstacle function array

Step 15: Go back to step 2

Step 16: End-while

Analysis:

As it can now be seen that the above algorithm shares similarities with the approach of Artificial Potential Field algorithm, first proposed by O. Khatib classically for stationary obstacles and goals. In the present problem the obstacles are generated randomly but after that they are stationary for the entire run. There is also a reactive behavior amongst the motions of the robots as each robot considers the others like obstacles and keep safe distance as well as a specific formation. The reactive behavior is also exhibited while the robots are repulsed from the obstacles. The troop is also attracted towards the goal more aggressively than they are being repulsed from the obstacles. This weighted approach is taken to find nearer space to global optimal solution while optimizing the composition of the goal function and maximum distance Gaussian profile obstacle function. The higher aggression to reach the goal reduces the probability of being confined to local minima and forces it to follow a much straighter path as can be seen from the traced out paths of the robots through the resulting diagrams (viz. Fig 6 and Fig 9).

Three functions have been used in this multi-vehicle path planning viz. 'obstaclegeneration', 'obstaclefunc', and 'goalfunc' along with the main program.

The main program executes the simulation loop of the constrained optimization problem and derives drive commands for the troop. The pseudo-code of the main program is given below:

loop for i=1 to size(sampled contour) ***% Starting of the Main Program***
theta(i,1) ← theta(i-1,1) + angular increment
end of loop
Set xgoal $_{1,2}$← assign goal coordinates $_{1,2}$ ***% Assign two goals and four initial locations***
Set initial$_{1,2,3,4}$← assign initial locations coordinate $_{1,2,3,4}$ ***% of the vehicles***
Call obstaclegeneration function ***% to generate eight random obstacles and to display***
Loop for k=1 to size(iterations)
Set x$_{1,2,3,4(min,max)}$ ← Workspace$_{(min,max)}$ ***% to keep vehicles within the workspace***
Loop for m=1 to size(sampled contour)
Xs$_{1,2,3,4}$(:,m) ← x$_{1,2,3,4}$ (rows,k)+increment(rad,theta$_m$)
Go(m,1) $_{1,2,3,4}$← call obstaclefunc(Xs$_{1,2,3,4}$(:,m),w1)
Gg(m,1) $_{1,2,3,4}$← call goalfunc(Xs$_{1,2,3,4}$ (:,m),xgoal$_{1,2,3,4}$,w2)
Ggo(m,1) $_{1,2,3,4}$← Go(m,1) $_{1,2,3,4}$+ Gg(m,1) $_{1,2,3,4}$
End of inner loop
minvalue$_{1,2,3,4}$← min(Ggo(m,1)$_{1,2,3,4}$ ***% minimum value and its sequence***
minvalueseq$_{1,2,3,4}$← sequence(min(Ggo(m,1) $_{1,2,3,4}$)
x$_{1,2,3,4}$(rows,k+1)← x$_{1,2,3,4}$(rows,k)+ increment(incr, theta(minvalueseq$_{1,2,3,4}$)
*deltaincr ← 0.1*incr*random generaration* ***% To generate artificial pertubation***
*deltaangle ← 2*pi* random generaration*
x$_{1,2,3,4}$ (rows,k+1)← x$_{1,2,3,4}$(rows,k+1)+increment(delta incr,theta(deltaangle))
Loop i=1 to (number of vehicles-1) ***% Formation & Coordination amongst the vehicles***
delta$_{1,2,3}$← x$_{i+1(x,y)}$-x$_{i(x,y)}$
if (delta<mags(safedistance) then x$_{i(x,y)}$ ← x$_{i(x,y)}$+shift
Plot online path tracing
End of last inner loop
End of the main simulation loop
Plot some results of the troop movement
End of the Main Program

Three functions were called from the main program of which the first one is obstacle generation. This function has not taken any input from the main program and not also returned any value but generates eight random obstacles in the world map of the vehicles.

5. Simulation results

The results obtained from the simulations are given in this section. Fig. 2 shows four vehicles at their starting points. Fig. 3 shows the input obstacle functions of the workspace. Fig. 4 shows the upper left and Fig. 5 the upper right goal functions estimated from the sampled workspace Fig. 6 shows the online tracing of the traversed path Fig. 7 shows the output vector from goal optimizing function Fig. 8 shows the output vector from weighted combination of Gaussian profile (obstacles) optimization and Fig. 9 shows the final paths traced out by the vehicles

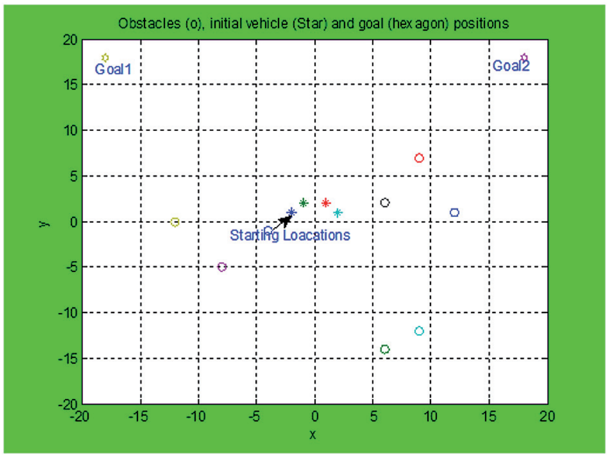

Fig. 2. Four vehicles just about to start in two different groups in the given workspace

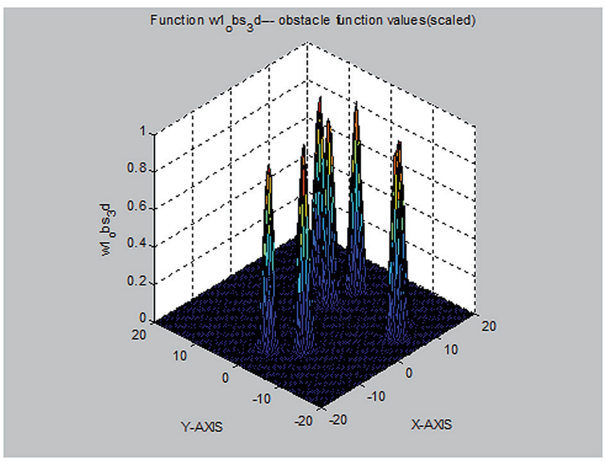

Fig. 3. Input obstacle functions estimated from the sampled workspace

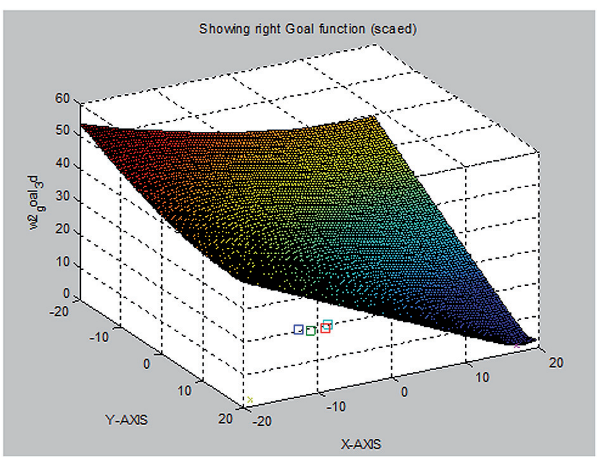

Fig. 4. Upper left goal function estimated from the sampled workspace

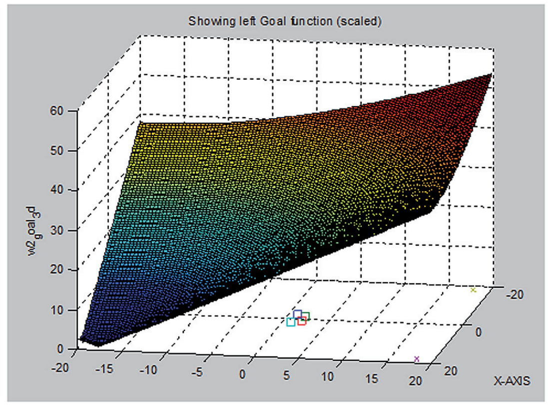

Fig. 5. Upper right goal function estimated from the sampled workspace

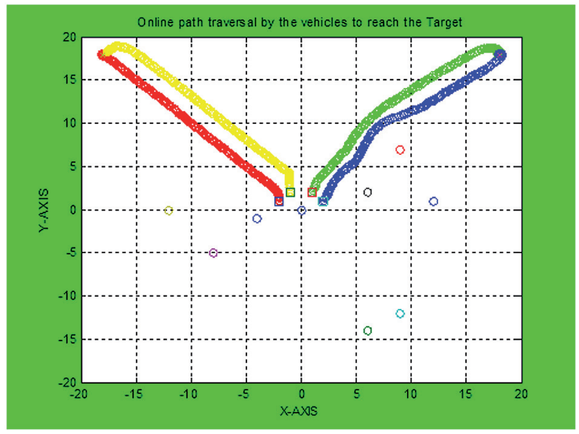

Fig. 6. Online tracing of the traversed path of four vehicles to two goals in a group of two

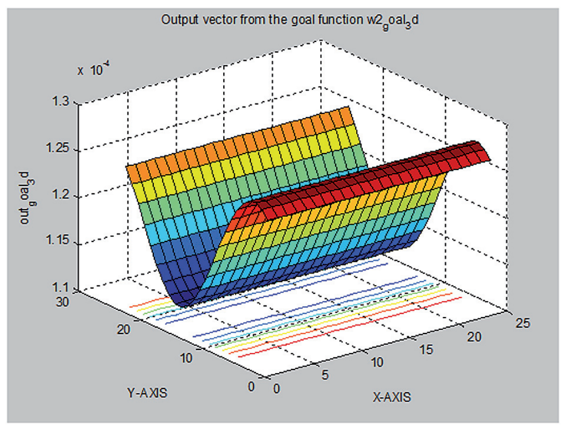

Fig. 7. Output vector from goal optimizing function for every iteration

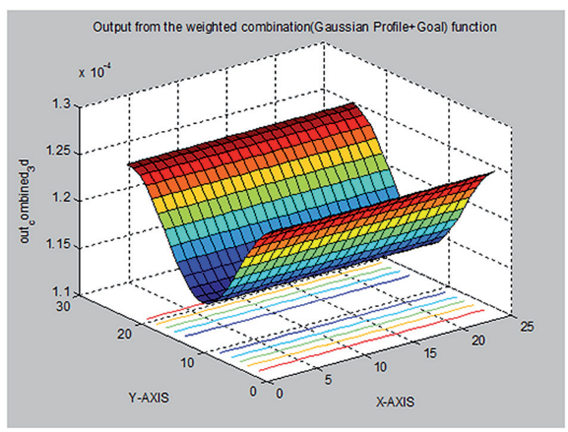

Fig. 8. Output vector from weighted combination of Gaussian profile (obstacles) optimizing and goal optimizing function

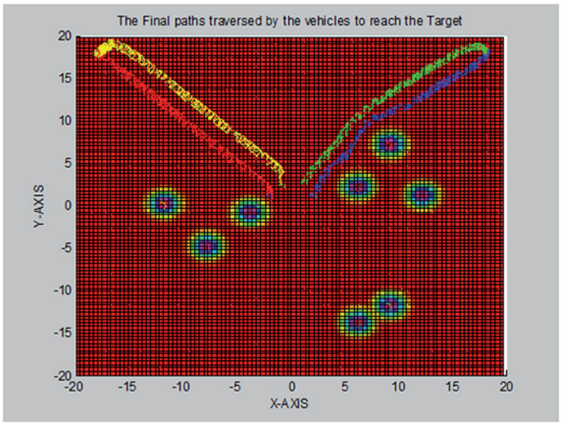

Fig. 9. Final paths traced out by the vehicles for a given random obstacle map and goals

6. Conclusions

There are many open issues in multi-vehicle path planning and coordination which are yet to be addressed. Currently used techniques are not suitable for very large number of vehicles and for 3-D trajectories (aerial vehicles). Another difficulty is faced in practical implementation of the real time mobile vehicles. It requires incorporating practical motion and sensing constraints of physical vehicles in 2-D space. As already mentioned this technique is an application of Artificial Potential Field Approach in static environment. This approach could be extended to control and coordination of mobile vehicles in highly stochastic and dynamic environment but that would be a slight deviation from classical APF and its complexity is higher. It may require online path planning and coordination strategies.

Motion coordination of multiple vehicles in a shared workspace has large scale practical values. Example applications include container management in ports, extra-planetary explorations, search and rescue, mineral mining, transportation, industrial and household maintenance, construction, hazardous waste cleanup, security, agriculture, and warehouse management. Due to complexity and cost, relatively few real-world implementations of these systems have been accomplished till date. It is expected that such systems will have wide-spread use in near future as the technology continues to mature.

Because of the need for motion coordination of multi-vehicle systems, the work described in this paper is of critical importance. As multi-robot systems can operate in stochastic and unpredictable settings, the study of the interaction dynamics of these settings may have broader impact in a wide range of applications. One possible solution of multi-vehicle problem has been presented in this paper. The task is to find out the optimal path towards goals avoiding obstacles by learning through random search in an unknown environment. A Gaussian Profile Map function optimally directs the vehicles away from the obstacles as if the Robots are repulsed from the obstacles. The vehicles are more aggressive towards the goals rather than to the obstacle avoidance phenomenon in this project. The time taken by the troop to reach two different goals in two pairs is less than a minute as being observed. This algorithm, if compared with others, the time efficiency could be found to be better to some extent. The obstacles are not dynamic in the present work but they may be stochastically generated at any location in the workspace. Moreover the troop is to maintain formation and coordination amongst themselves. A comprehensive result has been achieved by simulating the algorithm in MATLAB® environment. In highly stochastic environment, a more robust and adaptive algorithm may be required. Application of Neuro-Fuzzy or Neuro-GA system could be very useful in this context.

A simple algorithm based on random search has been used which is very easy to implement. It is based on APF approach and also shares some similarities with evolutionary computation techniques. The system is initialized with a population of random solutions and searches for optima by updating generations. The potential solutions, in this algorithm fly through the problem space following current optimal output. They are taken as the optimally best possible directives for movement of each individual vehicle. The controller updates the parameters in accordance with the optimal directives generated by the algorithm. Still then, for very complicated, dynamic and stochastic environments an expert system with leader-follower architecture may be even a better alternative.

Acknowledgement

This work was inspired by the prior research work of Kevin Passino [2] and O. Khatib [3]. The authors wish to acknowledge their contributions. The authors are also indebted to the Head of the Dept. of Electrical Engg, Jadavpur University, for allowing this research work to be conducted in its Mechatronics Laboratory.

AUTHORS

Sourish Sanyal* – Electronics & Communication Department, College of Engineering & Management, West Bengal University of Technology, West Bengal, India. E-mail: sourish2007_may@yahoo.co.in

Ranjit Kumar Barai, Rupendranath Chakrabarti, Pranab Kumar Chattopadhyay – Electrical Engineering Department, Jadavpur University, West Bengal, India. E-mails: ranjit.k.barai@gmail.com; rupen_chakrabarti@yahoo.co.in; pkchattopadhyay47@hotmail.com

*Corresponding author

References

[1] L. E. Parker, "Path Planning and Motion Coordination in Multiple Mobile Robot Teams", *Encyclopedia of Complexity and System Science*, Robert A. Meyers, Editor-in-Chief, Springer, 2009.

[2] K. Passino, *Biomimicry for Optimization, Control, and Automation*, Springer, 2005.

[3] 0. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", *International Journal of Robotics Research*, vol. 5, no. 1, 1986, pp. 90–98.

[4] M. Gerke, “Genetic path planning for mobile robots”. In: *Proceedings of the American Control Conference 1999*, vol. 4, 1999, pp. 2424–2429.
[5] J. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, Boston, 1991.
[6] 0. Khatib, “Real-Time Obstacle Avoidance for Manipulators and Mobile Robots”, *International Journal of Robotics Research*, vol. 5, no. 1, 1986, pp. 90–98.
[7] Zhang Pei-Yan, Lu Tian-Sheng, Song, Li-Bo Source, “Soccer robot path planning based on the artificial potential field approacch with simulated annealing”, *Robotica*, vol. 22, no. 5, September/October, 2004, pp. 563–566.
[8] C. Zhang, R. Ordonez, C. Schumacher, “Multi-Vehicle Cooperative Searchwith Uncertain Prior Information”. In:*Proceedings of the 2004 American Control Conference*.
[9] N.M. Kwok, Q.P. Ha, G. Fang, “Motion Coordination for Construction Vehicles using Swarm Intelligence”, *International Journal of Advanced Robotic Systems*, 2012. DOI: 10.5772/5672
[10] N.M. Kwok, Q.P. Ha, V.T.Ngo, S.M. Hong, “Particle Swarm Optimization of a Group of Construction Vehicles”. In: *ISARC 2006*.
[11] F. Arrichiello, *Coordination Control of Multiple Mobile Robots*. Dissertation at Universit`a Degli Studi Di Cassino, Dipartimento Di Automazione, Elettromagnetismo, Ingegneria Dell’informazione E Matematica Industriale.
[12] Ping Ping Khaw, W.S. Wijesoma, Eam Khwang Teoh, “Intelligent Control And Navigation of an Outdoor AGV”. School of Electrical and Electronics Engineering, Intelligent Machines Research Lab. Nanyang Technological University, Singapore. Available at: http://www.araa.asn.au/acra/acra1999/papers/paper43.pdf
[13] D. Rathbum, S.n Kragelund, A. Pongpunwattana, B. Capozzi, *Metron Aviation, Inc., Herndon,VA.*
[14] J.S. Bellingham, A. Richards, J.P.How, “Receding horizon control of autonomous aerial vehicles”. In: *Proc. American Control Conference*, Anchorage, Alaska, May 2002.
[15] J.S. Bellingham, M. Tillerson, M. Alighanbari, J.P. How, “Cooperative path planning for multiple UAVs in dynamic and uncertain environments”. In: *Proc. of 41th conf. Decision Contr.*, Las Vegas,Nevada, USA, 2002.
[16] Y.K.M. Alighanbari, J.P.How, “Coordination and Control of multiple UAVs with timing constraints and loitering”. In: *Proc. American Control. Conf.*, Denver, CO, 2003.
[17] W.B. Dunbar, R.M. Murray, “Model predictive control of coordinated multi-vehicle formations”. In: *Proc. of 41th Conf. Decision Contr.,* Las Vegas, NV, 2002.
[18] S. Ganapathy, K.M. Passino, “Agreement strategies for cooperative control of uninhabited autonomous vehicles”. In: *Proc. American Control conf.*, Denver, Colorado, June 2003.
[19] Y. Liu, M.A. Simaan, J.J.B. Cruz, “Game theoretic approach to cooperative teaming tasking in the presence of an adversary”. In: *Proc. American Control Conf.*, Denver, Colorado, June 2003.
[20] Q. Li, S. Payandeh, “Planning for dynamic multi-agent planar manipulation with uncertainty: A game theoretic approach”. In: *Proc. American Control Conf.,* Denver, Colorado, June 2003.
[21] M.E. Broucke, “Disjoint path algorithms for planar reconfiguration identical vehicles”. In: *Proc. American Control Conf.,* Denver, Colorado, June 2003.
[22] R.O. Saber, W.B. Dunbar, R.M. Murray, “Cooperative control of multi-vehicle systems using cost graphs and optimization”. In: *Proc. American Control Conf.*, Denver, Colorado, June 2003.
[23] J-C. Latombe, “*Robot Motion Planning”*, Kluwer Academic Publishers, 1991.
[24] S. M. LaValle, “*Planning Algorithms”*, Cambridge University Press, 2006.
[25] J. E. Hopcroft, J. T. Schwartz, M Sharir, “On the complexity of motion planning for multiple independent objects; PSPACE-Hardness of the ”Warehouseman’s Problem”, *International Journal of Robotics Research*, vol. 3, 1984, no. 4, pp. 76–88.