Multi-Agent Mobile Robot Navigation through Genetic-Fuzzy Algorithm

Nirmal Baran Hui
Department of Mechanical Engineering
National Institute of Technology, Durgapur

Taimika Biswas
Department of Computer Science & Engineering
St. Thomas College of Engineering & Technology, Kolkata

Susmita Samanta
Department of Computer Science & Engineering
St. Thomas College of Engineering & Technology, Kolkata

Saurabh Pratap
Department of Industrial & Systems Engineering
Indian Institute of Technology, Kharagpur

Abstract

The present paper deals with the navigation of multiple wheeled robots working in a common dynamic environment in a decentralized manner. Two different motion planning approaches have been proposed to solve the said problem. In Approach 1, automatic design of a Mamdani-type Fuzzy Logic Controller (FLC) using a binary-coded Genetic Algorithm (GA) has been considered. On the other hand, a potential field- based motion planner has been developed in the next approach. Since, the robots are agents and all the time competition is not good, a strategic approach has been proposed to solve the conflicts related to coordination among agents. Performance of the developed approaches have been tested through computer simulations. Computational complexity of both the approaches are also compared to see the feasibility of their on-line implementations. It has been observed that proposed coordination strategy along with the developed motion planners are found to generate movement of the robots in a human-like manner.

Keywords- Mobile Robot Navigation; Multi-agent System; Coordination; Potential Field Method; Fuzzy Logic Control; Genetic Algorithm

I. INTRODUCTION

Multiple mobile robots working in a common work-space will have to negotiate their motion. Main aim here is to find collision-free paths of all the robots while they are moving from their respective starting points to the destinations. The path for each robot is constrained by its current position, the goal point and the movements of the other robots. Therefore, it is a complicated task and there must be an intelligent and adaptive motion planner to solve the same. Motion planner can be designed in two ways. Firstly through centralized manner, in which there will be a master robot who will dictate the motion plan to other robots and other robots obey the master. However, it suffers from following drawbacks.

- Malfunctioning of the master robot leads to the failure of the whole system,
- Skill of each robot is restricted due to less flexibility and autonomy in the system,
- It breaks down completely in non-homogenous environments.

Therefore, most of the researchers are preferring the other option, which is known as decentralized motion planning. In case of decentralized system, each robot carries out tasks cooperatively. This sort of system offers more freedom to the robots and allows each robot to take the decision independently/selfishly. However, to build a full-proof decentralized system, it should have three design characteristics Kim et al. (2004): Coordination, Communication and Cooperation. For better understanding of those characteristics, interested readers may go through Kim et al. (2004).

Quite a few researchers Mackworth (1993); Asada et al. (1999); Alami et al. (1998); Kim et al. (2004); Ota (2006) have considered soccer playing could be one example of decentralized motion planning of multiple agents. They have set some long term goal. Most of the researchers are presently trying to fulfill the long term goal set by Mackworth (1993) and Alami et al. (1998). Navigation of multiple robots includes generation of local plans of each agent maintaining a proper communication among them and coordination of these local plans to achieve the desired aim. Peng and Akella (2003) have tried to solve this problem considering it to be a mixed integer nonlinear programming problem, in which path of each agent is decomposed into collision-prone and collision-free segments. However, this approach is not at all suitable for real-time systems, as the environment is highly dynamic in nature. Reinforcement learning Nakamura (2006); Uchibe et al. (1998) has been widely used to develop some combined task forming of a Multi Agent System (MAS). In reinforcement learning, the main aim is to maximize the reward and minimize the penalty. However, the performance of this approach widely depends on the chosen reward and/or penalty function, selection of which demands a-priori knowledge of the problem.

Park et al. (2001) developed a method based on Modular Q-learning with uni-vector field method. In this method, they used overhead vision system and collected all important information and then transferred to host computer. They have considered two situations, blocked and non-blocked situations. In case of non-blocked situation 96 and for blocked situations 104 different

states were considered (depending on robot location) and then for each state, action was defined (like whether the robot will kick/defend/attack). For both the situations, different expressions of rewards have been assumed and uni-vector field function helped in navigation. How- ever, this process is computationally expensive and setting of actions corresponds to a state is very difficult for a person who does not have proper knowledge about the circumstances. A Case Based Reasoning (CBR) technique have been developed by Ros et al. (2009) for selection of actions in robot soccer. They divided the entire terrain into some controllable (team mates positions) and non-controllable features (balls & opponents position). For each and every feature, they have used separate rules to set the actions. This becomes cumber- some with the increase in the number of players. Springer et al. (1997) proposed some strategies for avoiding collisions in a soccer robot system. Later on Jung and Zelinsky (1999) proposed a behavior-based architecture scheme for cooperative motion planning of multiple robots. However, their approaches are found to be very simple and restricted. Moreover, none of them take into account the coordination issues among the robots. Game theory has been adopted by Skrzypezyk (2004) and Lucaterio et al. (2001). Both of them used maximin and minimax concepts to find solution of the problem. The maximal Nash equilibrium concept was utilized to find the favorable strategies for each robot. This approach is fast enough to cope with the coordinated situations of real robots. Strategies by the agents were finally adopted based on their interaction history. For the purpose of which learning plays an important role in this context. Geometry Data Based (GDB) technique have been implemented by Kasinski and Skrzypczynski (2001). They used multi-mode sensors to collect information of the environment, and form the perception network. Wu et al. (2013) described the strategy of robot soccer game and proposed two models (a) ranking model to indicate the weak node of strategy (b) probability based model to evaluate the strategy followed by the robot. Qu et al. (2013) introduced a global path planning for robot navigation. In this, model used to solve by Co-evolutionary Improved Genetic Algorithm (CIGA) andGA. Rocha et al. (2005) developed a vision based system for a cooperative multi-robot system. They used SLAM Architecture to build 3D map of any unknown environment. Then updating the map upon new sensory information based on Gaussian sensor model through some stereo vision sensor was carried out. However, this process involves handling of huge data, which is often cumbersome and computationally expensive.

Sevstka and Overmars (1995) used probabilistic road maps to solve the multi-robot path planning problems. In their approach, the motion of an agent is restricted in a zone/grid. Quite a few researchers Vail and Veloso (2003); Lefebvre et al. (2004); Warren (1990) have tried to use potential field method, which is very popular for motion planning of single agent. Some of them also considered separate potential functions for different agents and different task achievement. However, the performance of potential field method depends on the selection of potential functions. Moreover, in potential field method, there is no in-built optimization module, thus it may give some feasible solution which may not be an optimal one. Bennewitz and Burgard (2000) suggested a probabilistic method for planning collision-free trajectories of multiple mobile robots. Maa (2014) proposed a model for multi robots in dynamic warehouse. In this, PSO-based Con-Per-PSO and SA-PSO algorithms are proposed to solve path planning in dynamic warehouse. Wang and Xin (2012) introduced a model for multiple autonomous robots cooperative tracking considering obstacle avoidance. In this, the control design model focused on cooperative tracking, obstacle avoidance and control effort minimization. Buck et al. (2001) developed a hybrid scheme of different planning methods, in the context of robot soccer. Xi et al. (1993) developed an event-based method. Yao and Tomizuka (1993) proposed an adaptive control method and Wang and Pu (1993) used cell-to-cell mapping for multiple coordinating robots. Some more attempts were made by various researchers to solve coordination problems of both manipulators as well as mobile robots. Latombe (1991) provides an extensive survey on the algorithmic approaches of robot motion planning. Soft computing-based motion planning schemes have also be proposed by some researchers. Fuzzy logic Marapane et al. (1996); Pradahan et al. (2009) and neural network Nelson et al. (2004); Jolly et al. (2007) have also been used by some researchers. However, all such methods have some common drawbacks, which are listed below.

1) Most of the researchers have neglected the competition among the robots.
2) Role of an agent is kept fixed, thus the robustness and adaptability of a agent is very low.
3) A particular agent is allowed to navigate in a fixed zone.

Therefore, the coordination among the agents is still a challenging research issue in robotics. In the present study, an attempt was made to solve the motion planning problem of multiple mobile robots moving in a com- mon dynamic environment. For the motion planning purposes, two different approaches have been developed and some strategies are proposed to evolve the coordination's among the agents. The rest of the paper is structured as follows: in Section 2, coordination of multiple mobile robots have been studied. Developed navigation schemes and coordination strategies are discussed in Section 3. Results are presented and discussed in Section 4. Finally, some concluding remarks are made and the scopes for future work are indicated in Section 5.

II. COORDINATION OF MULTIPLE MOBILE ROBOTS

Multiple mobile robot working in a common dynamic environment constitute a Multi Agent System (MAS). In such a system, all the robots will have to find their collision-free paths during navigation among themselves. Depending on the postures occupied by the other robots, a particular robot may find a number of collision- free paths. However, our aim is to determine that particular path, which is not only the collision-free but also time-optimal. The performances of the developed motion planners have been tested on an experimental set-up consisting of some car-like robots, one of which is shown in Figure 1. Kinematic and dynamic constraints of the robot may impose some restrictions on its motion. Therefore, a particular collision-free path (may be time-optimal also) of the robot may not be possible to achieve until or unless the constraints are satisfied. Figure 2 shows a typical problem scenario, in which four two-wheeled differential drive robots are navigating in a common dynamic environment. Each

robot will have to find its time-optimal and collision-free path. Starting and goal positions of any planning robot (say, R1) are denoted by S and G. Now, to reduce the complexity of the problem, only one robot has been treated as the most critical one and the motion of the planning robot is planned based on that particular robot. Moreover, the wheels of the robot are allowed to move due to pure rolling action only and Coriolis component of the force is neglected, in the present study. Thereafter, the total path of the robot is assumed to be a collection of a number of small segments, each of which is traveled during a fixed time.

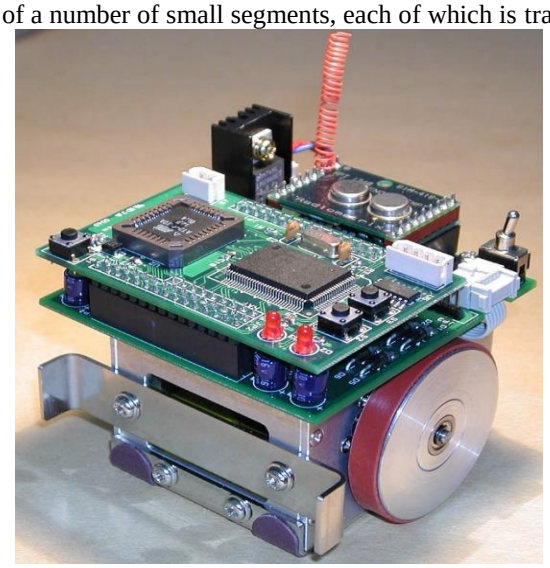

Fig. 1: Photograph of a robot used in the experiment.

ΔT the robot negotiates its motion during those time steps, in order to avoid collision with the most critical robot. The critical robot has been identified depending on the relative position of the robot and the obstacles. The robot physically closest to the planning robot, may not be treated as the most critical one always. If any robot lies within an angle of 120◦ (within +/- 60◦ from the robot's main axis and inside the imaginary extended bounding circle of the planning robot) and is directed towards the planning robot, then it might be considered as the critical one. Among all such robots lying within the angle of search, the physically closest one is taken as the most critical one. Thus, although the robot R4 is the physically closest to the robot, it is not being treated as the most critical one. Rather, the robot R3 is considered to be the critical one, because it lies within the angle of search and is directed towards the robot also. The radius of the imaginary extended boundary circle of the planning robot is taken to be equal to the distance that the robot can travel in one time step. The angle of search is decided based on the fundamentals of human vision. Now, the motion of the planning robot is planned based on the two inputs – distance, angle as shown in Figure 2. The motion planner will determine acceleration and the angle through which the planning robot (say, R1) should deviate with respect to the reference line, so that it can avoid collision with the other robots moving in the same environment.

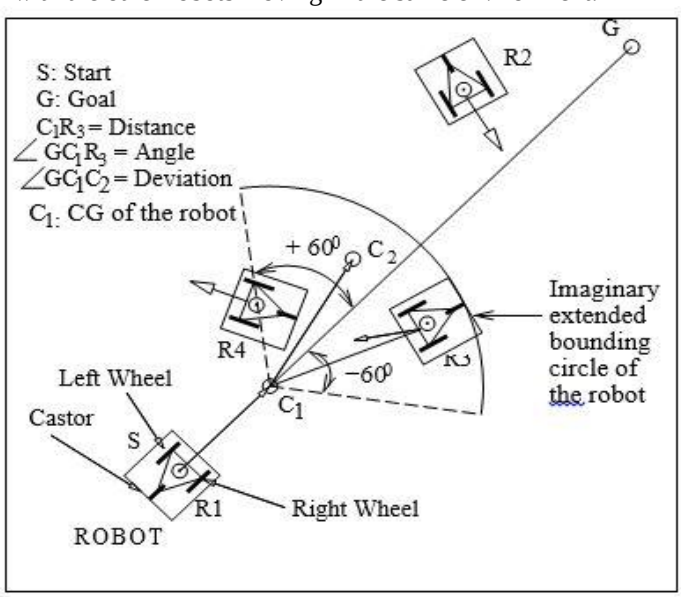

Fig. 2: Robots navigation in a dynamic environment.

A. Mathematical Formulation of the Problem

The developed motion planning scheme of the robot is explained with the help of Figure 3. In a 2-D space, multiple robots are moving starting from an initial position with different speed and in different direction. Starting and final positions of all the robots are defined a-priori and those of one robot are different from the other. The total path (starting from a pre-defined position to a fixed goal) of any robot is assumed to be a collection of some small segments (either a straight one or a combination of straight and curved paths), each of which is traversed during a fixed time ΔT. If a robot finds any other robot to be critical robot (which may collide with the planning robot if it moves along the previous direction and by maintain the same speed), the motion planner is activated. Otherwise, the robot moves toward the goal in a straight path with a maximum possible velocity. The task of the motion planner is to determine the acceleration (a) and deviation (θ1) of the robot based on the distance and angle inputs, to avoid collision with it. Since distance is one of the major factor based on which critical robot is identified. Thus, there is a chance that a robot critical to the planning robot may also consider the planning robot to be critical during its own motion planning. As a result of which, both will get deviated from their previous direction of motion and their speed will also be hampered. It will then increase the traveling time to be taken to reach the goal by the robots. In order to avoid the same a strategic decision tool is adopted to predict which robot will cooperate with the other in a particular time step. This process of motion planning will continue, until all the robot reaches their individual destination and total traveling time for each robot (T) is then calculated by adding all intermediate time steps taken by the robot to reach it. It is important to mention.

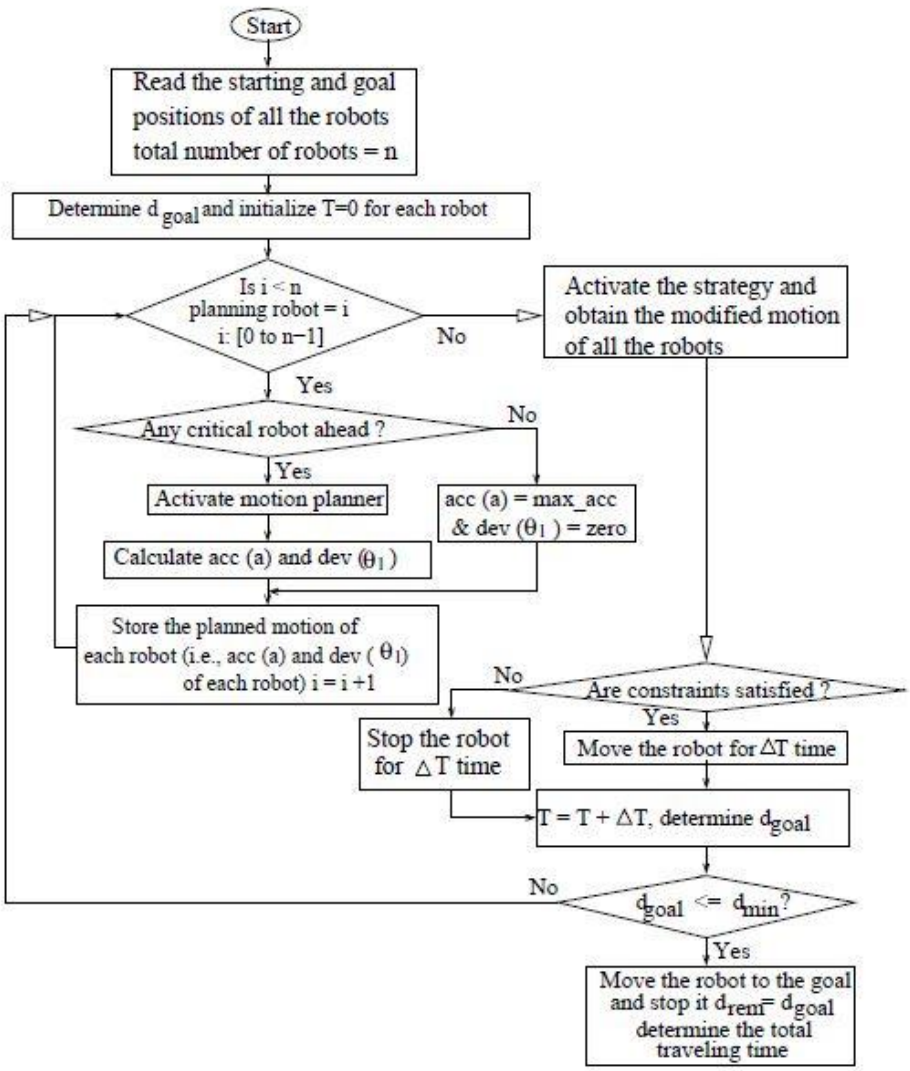

Fig. 3: A schematic diagram showing the working of the developed motion planning scheme.

That the last time step (T_{rem}) may not be a complete one and it depends on the distance left uncovered (d_{goal}) by the robot. If it (i.e., the goal distance d_{goal}) comes out to be less than or equal to a predefined minimum distance (d_{min}), it starts decelerating and stops at the goal. Again, sometimes the robot's motion as provided by the motion planner may violate its kinematic and/or dynamic constraints. In such a situation, the robot is stopped at the present position itself. Our aim is to design a suitable adaptive and cooperative motion planner, so that all the robots will be able to reach their destination with the lowest possible traveling time

by avoiding collision among themselves. Therefore, the present problem can be treated as a constrained traveling time (T) optimization problem as indicated below.

$$Minimize\, T = \sum_{i}^{n} (U^i \times \Delta T + T_{rem}^i), \qquad (2.1)$$

Where U i indicates the number of complete time steps for robot i and n denotes the total number of robots present in the environment.
Subject to

– The path is collision-free,
– Following constraints of the robots are satisfied Bemporad et al. (1996); Fraichard and Scheuer (1996); Hui (2007).

1) Kinematic Constraints

$$(i)\ -\dot{X}\cos\theta + Y\sin\dot{\theta} = 0, \qquad (2.2)$$

$$(ii)\ (\dot{X})^2 + (\dot{Y})^2 - (\rho_{\min}\dot{\varphi})^2 \geq 0 \qquad (2.3)$$

Where $\dot{X}$ and $\dot{Y}$ are the component of tangential velocity along +ve X-axis and +ve Y-axis, respectively and θ is the angle between the X-axis and the main axis of the robot. Minimum radius of curvature is represented by ρ_{min} and $\dot{\phi}$ denotes the rate of steering angle during turning.

2) Dynamic Constraints

$$(i)\ \sqrt{(\mu_{fg})^2 - (\upsilon\dot{\phi})^2} \leq \alpha \leq \sqrt{(\mu_{fg})^2 + (\upsilon\dot{\phi})^2} \qquad (2.4)$$

$$(ii)\ \alpha \geq \frac{60P}{2\Pi r \times GR \times M \times N_m}, \qquad (2.5)$$

$$(iii)\ \upsilon \geq \rho_{\min}\dot{\varphi} \qquad (2.6)$$

Where, *v* and α denotes the tangential velocity and acceleration of the CG of the robot, respectively and power required by the motor to create maximum angular speed Nm is expressed by the term P. Moreover, GR represents the gear ratio of the wheels, r is the radius of the wheels of the robot and the mass of the robot is denoted by M. Again, μ_f indicates the coefficient of friction between the wheels and the surface of the terrain, and acceleration due to gravity is represented by g.

III. DEVELOPED NAVIGATION SCHEMES & COORDINATION STRATEGY

Several methods had been tried by various investigators to solve similar kind of problems. The authors have developed two motion planning approaches along with a novel coordination strategy, all these are discussed below one after another.

A. Approach 1: Genetic-Fuzzy System

In this approach, a fuzzy logic-based motion planner is developed for solving the navigation problems of a real car-like robot. Two inputs, such as distance of the robot from the most critical obstacle and angle between the line joining the robot and the most critical obstacle and the reference line (joining the robot and its goal) have been considered for the motion planner and it generates two outputs, namely deviation and acceleration required by the robot to avoid collision with the obstacle.

In the present study, the range of distance is divided into four linguistic terms: Very near (VN), Near (NR), Far (FR), and Very Far (VF). Five linguistic terms have been considered for both the angle as well as deviation: Left (LT), Ahead Left (AL), Ahead (AH), Ahead Right (AR) and Right (RT) and acceleration is considered to have four terms: Very Low (VL), Low (L), High (H), and Very High (VH). As there exists a maximum of twenty (i.e., 4 × 5) input combinations and for each input combination, a maximum of twenty output (i.e., 5 × 4) combinations are possible, a maximum of 400 (i.e., 20 × 20) rules are possible. One such rule may look like the following:
IF distance is VF AND angle is LT, THEN deviation is AH and acceleration is VH.

For ease of implementations, membership function distributions of both the input as well as output variables are assumed to be symmetric triangles (refer to Figure 4). Thus, the data base of the FLC may be modified using four continuous variables representing the half base-widths (i.e., V1, V2, W1, W2) of the triangular membership function distributions. The performance of an FLC depends on both its data base as well as rule base, which are to be optimized simultaneously. It is observed that the performance of an FLC is mainly dependent on its rule base and optimizing the data base is nothing but a fine-tuning process Pratihar et al. (1999). Thus, a proper care is to be taken to optimize the rule base of an FLC.

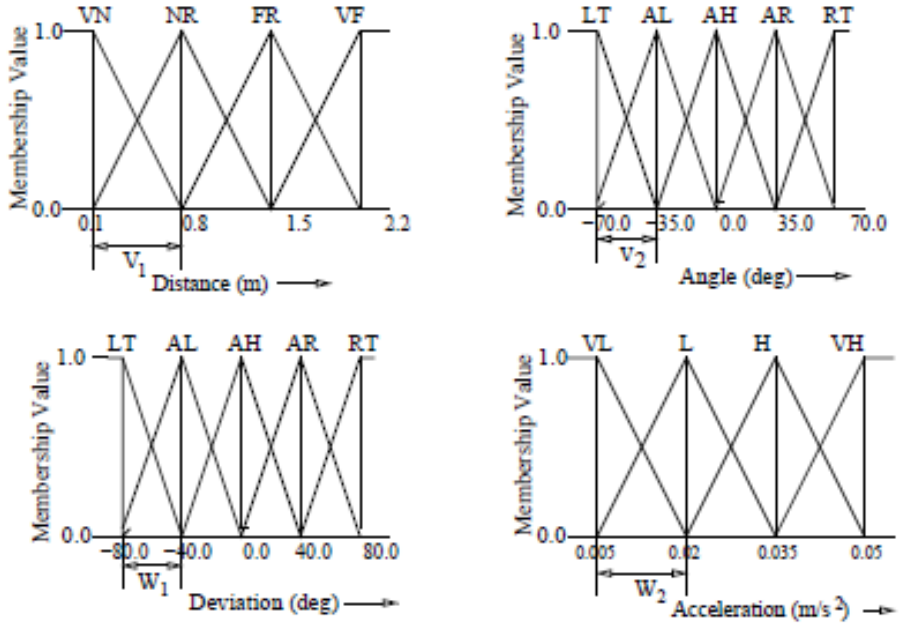

Fig. 4: Membership function distributions for input and output variables of the FLC.

Several methods have been proposed by various investigators to develop an optimal knowledge base (consisting of both rule base as well as data base) of an FLC. Quite a variety of techniques, such as, least square method Pham and Valliappan (1994), gradient descent technique Nomura et al. (1992), back-propagation algorithm of neural network Takagi and Sugeno (1983); Wang and Mendel (1992), reinforcement learning Berenji (1992), Tabu search algorithm Denna et al. (1999), ant colony optimization Casillas et al. (2000), genetic algorithm Wang and Yen (1999); Yupu et al. (1998); Pratihar (2000) and others, have been used for the said purpose. However, all of these methods have some limitations as indicated below.

1) Not all the methods are able to generate globally optimal rules, as they may suffer from the local minima problem,
2) When the number of rule increases in the rule base, not all the methods can provide with a feasible solution,
3) Computational complexity of some of the above techniques is quite high and thus, they may not be suitable for on-line application,
4) Most of the methods may generate some redundant rules, which will have less importance to the problem to be solved.

Thus, it is still a research issue to search for a method, which will design a globally optimal knowledge base (Consisting of more number of important rules) of an FLC within a reasonable computational time.

To improve the performance of FL-based motion planner, an automatic design procedure of FL using a binary-coded GA is adopted in the present study. A GA-string consisting of 440-bits is considered to represent the KB of the FLC as shown below.

$$\underbrace{1..1\ 0..1\ 1..0\ 0..1}_{\textit{Data base}} \quad \underbrace{10\cdots01}_{\textit{Input combinations}} \quad \underbrace{101\ldots101\ldots100}_{\textit{Consequent of the rules}}$$

First 40-bits are utilized to represent the data base i.e., V1, V2, W1, W2, (10 bits for each variable) of the FLC and the next 20-bits are used to indicate the presence or absence of the input combinations of the RB (1 for presence and 0 for absence). Out of remaining the 380-bits every 19-bits gives the output combination for a particular input combination. The total number of 1*s* present in each 19-bits long sub-string is counted and if it comes out to be equal to zero, it represents the first output combination, i.e., deviation is LT and acceleration is VL, and so on.

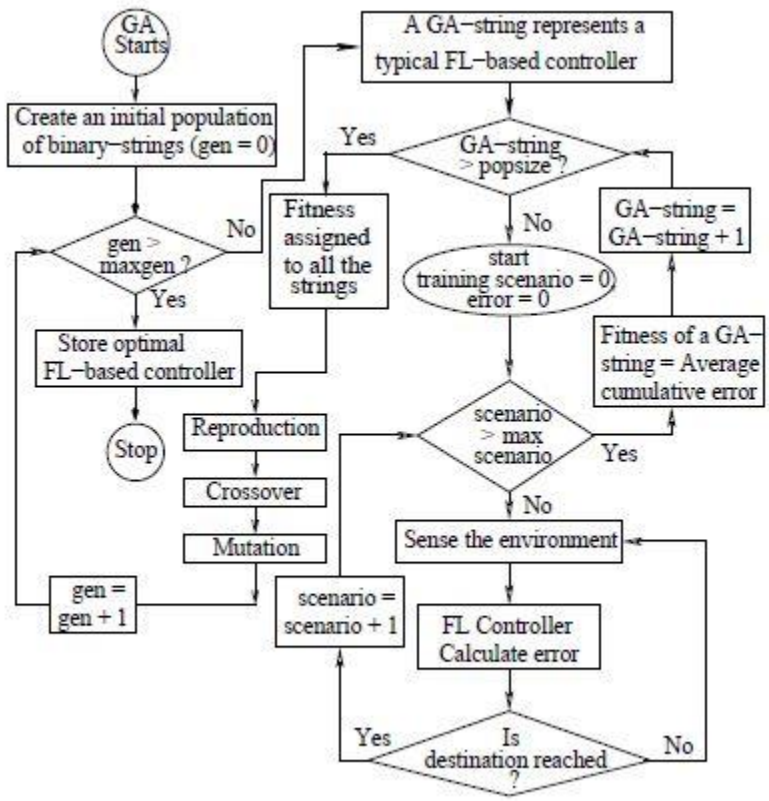

Fig. 5: shows the working principle of the combined GA-FLC approach.

The GA begins, its search by randomly creating a number of solutions (equals to the population size) represented by the binary strings, each of which indicates a typical FL-based motion planner. Each solution in the population is then evaluated, to assign a fitness value. The fitness of a GA-string is calculated using the equation given below.

$$Fitness = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{U}\sum_{s=1}^{U}\sum_{\upsilon=1}^{2}(T_{nsv} - O_{nsv}) + \text{Penalty-Reward},$$

where U denotes the total number of time steps in a planned path and the total number of training scenarios is indicated by N. O_{nsv} and T_{nsv} are representing the values of actual and target outputs, respectively, of an output variable (say,v). The target outputs for deviation and acceleration are taken to be equal to zero and maximum permissible acceleration of the robot, respectively. A fixed penalty equals to 2000 is added to the fitness of a string, if the motion planner represented by it is unable to generate a feasible motion of the robot. Moreover, a fixed reward equals to 100 is given to a string that allows at least some robots to coordinate others. Once the fitness is assigned to each solution in the population, they are modified using three operators – reproduction, crossover and bit-wise mutation. One iteration involving these three operators followed by the fitness evaluation is called a generation. Generations proceed until a termination criterion is satisfied. In this approach, the GA is allowed to run for a pre-specified number of generations. It is to be noted that during optimization, V1, V2, W1, W2, are varied in the ranges of (0.4, 0.7), (20, 40), (20, 40) and (0.005, 0.015), respectively.

B. Approach 2: Potential Field Method as proposed in Ge and Cui (2000)

Potential field method was first proposed by Khatib Khatib (1986) in the year 1986. According to this approach, any robot moves due to the combined action of attractive potential generated by its goal and repulsive potential created by opponent robots moving in the same environment. Different potential functions have been proposed in the literature. The most commonly used attractive potential takes the form Latombe (1991); Borenstein and Koren (1989); Koren and Borenstein (1991).

$$U_{att(q)} = \frac{1}{2}\xi\rho^2(q, q_{goal})$$

where ξ is a positive scaling factor and ρ(q, qgoal) is the distance between the robot q and the goal qgoal . One commonly used repulsive potential function takes the following form Latombe (1991):

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\eta(\frac{1}{q(q,q_{obs})} - \frac{1}{\rho_o})^2, & if\ \rho(q,q_{obs}) \le \rho_o \\ 0 & if\ \rho(q,q_{obs}) > \rho_o \end{cases}$$

Where η is a positive scaling factor, ρ(q, qobs) denotes the minimal distance from the robot q to the obstacle, qobs denotes the point on the obstacle and ρ0 is a positive constant denoting the distance of influence of the obstacle.

There are several drawbacks associated with the traditional PFM. Interested readers are asked to go through the literature Koren and Borenstein (1991) for more details. However, looking onto the eqn. 3.3, it is clear that the magnitude of attractive potential varies with the distance between the robot and its goal. As a result of which, as the robot reaches to the goal, it loses its momentum leading to overall increase in traveling time of the robot. The situation becomes more critical if any planning robot faces any critical robot in front of it, while the same is closer to the goal. This problem is known as Goal Non-Reachability with Obstacles nearby (GNRON). The GNRON problem arises because the global minimum of the total potential field is not at the goal position when the goal is within the influence distance of the obstacle. It is found that if the repulsive potential approaches zero as the robot approaches the goal, the total potential will take the global minimum at the goal. This motivated Ge and Cui Ge and Cui (2000) to construct a new repulsive potential function which takes the relative distance between the robot and the goal into consideration as

$$U_{rep}(q) = \begin{cases} U_{rep}, & if\ \ \rho(q,\rho_{obs}) \le \rho_o \\ 0 & if\ \ \rho(q,\rho_{obs}) > \rho_o \end{cases} \qquad (3.5)$$

Where $U_{rep} = \frac{1}{2}\eta(\frac{1}{\rho(q,q_{obs})} - \frac{1}{\rho_o})^2 \rho^2(q,q_{goal})$

In comparison with (3.4), the introduction of $\rho(q, q_{goal})$ ensures that the total potential U_{total} (q) = U_{att} (q)+ U_{rep} (q) arrives at its global minimum, 0, if and only if q = q_{goal} .

The robot is then allowed to move due to the combined action of attractive and repulsive forces derived by differentiating the corresponding potentials. Total force is computed by summing the above two force vectors. In the present study, acceleration of the robot is made proportional to the resultant force and future direction of movement is made along the direction of the resultant forces.

C. Proposed Coordination Strategy

In the presently developed navigation scheme, there are two inputs of the motion planner, distance and angle. Distance is nothing but the Euclidean distance between the planning robot and the most critical robot, which try to obstruct the movement of the planning robot. Therefore, there is a chance that the planning robot might also turn out to be critical robot to the previously critical robot. Say, in a multi agent system (refer to Figure 1), there are four robots and robot R3 is most critical to R1 in a particular time step. In the same time step, R1 might also become most critical to R3. In such a scenario, motion planner will plan the motion of both the robots individually. As a result of which, there could be two incidences. In the first incidence, they might avoid the collisions between them, but it would have been better that deviation required by one robot was not necessary. If it happens to be so then overall outcome of the game will be deteriorated. The present problem might be solved from general cooperative attitude and there could be at least three possible solutions.

- Zero Coordination: The robot will not sacrifice its planned motion, i.e., it will follow the deviation and acceleration as suggested by the motion planner.
- Full Coordination: The robot will sacrifice its planned motion, i.e., it is ready to follow the outputs which is different as suggested by the motion planner.
- Partial Coordination: Since there are two outputs in the present study, the robot might sacrifice one of them. The robot may follow the deviation as suggested by the motion planner and readjust the acceleration as necessary to avoid the collision with the opponent or vice-versa.

In the present study, two different strategies have been used to resolve the conflicting situations, i.e., when two robots in a time step mutually treat each other to be critical. Results of both the approaches are presented here for two different strategies as mentioned below.

1) Strategy 1: Both the robot will adopt zero coordination strategy, i.e., they will move along the direction and with acceleration as planned by the motion planner.
2) Strategy 2: Here one robot will adopt the zero coordination and the other one will adopt partial coordination, i.e., it will move with the acceleration as suggested by the motion planner and will altercate its future direction of movement using the collision avoidance scheme as suggested by Hui et al. Hui et al. (2006). The robot whose planned deviation is less will be allowed to follow zero coordination and the other will follow the partial coordination.

IV. RESULTS AND DISCUSSIONS

The performances of the developed approaches have been compared among themselves as explained below. The cycle time (ΔT) is assumed to be equal to 16 seconds for carrying out the computer simulations. The limits of acceleration (a), velocity (v), steering rate ($\dot{\phi}$) and minimum radius of curvature (ρmin) of the robot are given below.

$$0.005m/s2 \leq a \leq 0.05m/s2$$
$$0.007m/s \leq v \leq 1.0m/s \qquad (4.6)$$
$$-30^{\circ} \leq \dot{\phi} \leq 30^{\circ}$$
$$\rho min \geq 0.063m$$

In the developed soft computing-based approach, the FLC is trained off-line, with the help of a GA, as explained earlier. The computer simulation is carried out for three different cases. In the first case, eight moving robots are considered, whereas ten and twelve moving robots are taken into consideration in the 2nd, and 3rd cases, respectively. A field of size of 20m × 20m is considered in computer simulations. Considering the physical dimensions of the robot, a hypothetical field of size of 19.95m x 19.95m is used to prevent hitting of the robot with the boundary of the field. For tuning of the FLC, a set of 200 training data is created at random, in which initial position, final position and direction of movement of the robots have been varied. With all such randomly-generated training data, the robots start moving towards their goal. It is to be noted that a batch mode of training has been adopted in this thesis. After the tuning of the FLC is over, their performances have been compared among the Approaches 1 and 2, in terms of traveling time and their CPU times, for a set of 20 randomly-generated test scenarios. In the next sub-sections, results of both the training as well as test scenarios are explained in detail, for all the three cases.

A. Case 1: Motion Planning of Ten Robots

Ten robots are cooperating among themselves in a grid of 19.95 × 19.95m2 in a 2-D space. The initial and final points are specified for all the robots. Each of these robots will try to find a collision-free, time-optimal path to reach its destination. Performance of FLC-based approach depends on the selection of a good Knowledge Base (KB) consisting of Rule Base (RB) and Data Base (DB). A binary-coded GA has been used for this purpose. To get best result from GA, a systematic study has been carried out in the similar manner as mentioned in Section 4.1. Best results have been observed with the following GA-parameters: crossover probability (p_c) = 0.84, mutation probability (p_m) = 0.0036, population size (popsize) =120, maximum number of generation (Max_{gen}) = 190.

Through this process of optimization GA has selected twelve good rules as presented in Table 3. Optimized DB of the FLC is show in Figure 8.

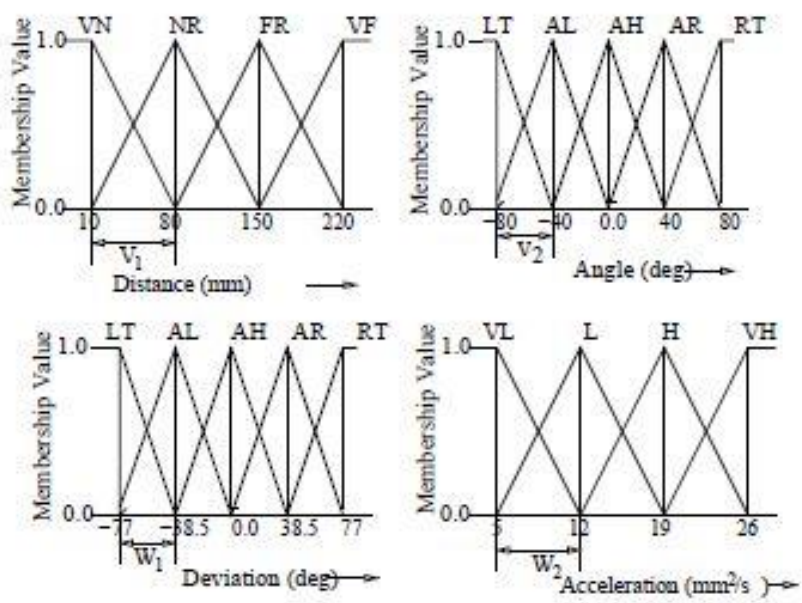

Fig. 8: Optimized membership function distributions of the FLC – Case 1.

Table 3: Optimized rule base of the FLC for ten robots case.

Distance	Angle	Deviation	Acceleration
VN	AL	AH	VL
VN	AH	AH	VL
VN	AR	AH	L
VN	RT	AH	VL
NR	AH	AH	VL
NR	RT	AH	VH
FR	LT	AR	VL
FR	RT	AL	H
VF	LT	AH	VL
VF	AL	AH	L
VF	AH	AH	VL
VF	RT	AH	H

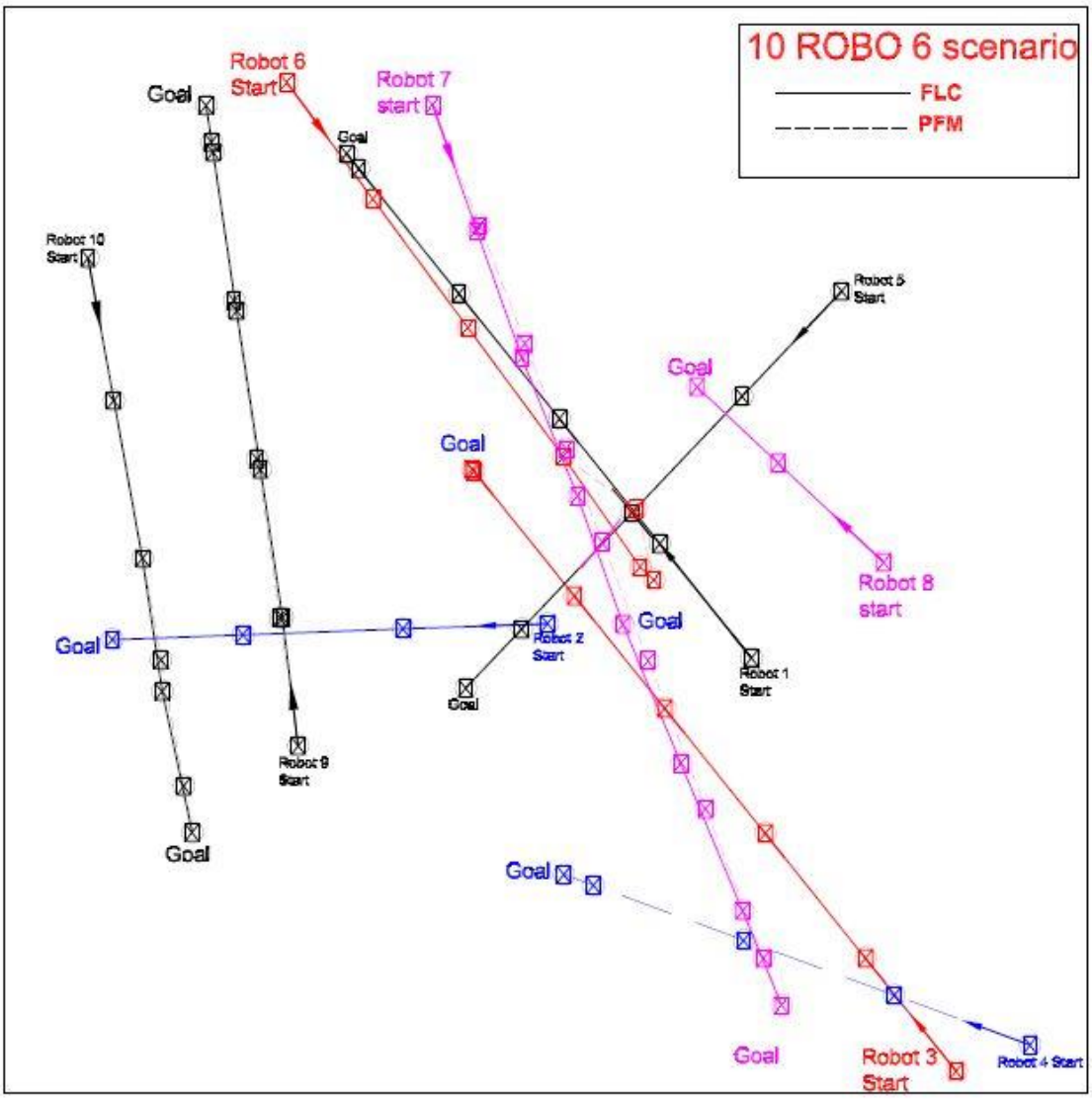

Fig. 10: Movements of the robots in 2D space – sixth scenario of Table 4.

After the tuning is over, effectiveness of optimized FLC (Approach 1) is compared with that of PFM (Approach 2) for solving twenty different scenarios. Traveling time taken by both the approaches following Strategy 2 are tabulated in Table 4. In most of the scenarios, Approach 1 has outperformed (refer to Table 4), Approach 2. It may be due to the fact that there is no in-built

optimization module inside the PFM. The paths planned by all ten cooperating robots for a test scenario (say, sixth of Table 4) are shown in Figure 10. CPU times of both the approaches are observed to be low, making them suitable for on-line implementations.

B. Case 2: Motion Planning of Twelve Robots

Motion planning of twelve robots have been considered in this case. During optimization of FLC, best performance of GA has been observed with the following parameters: crossover probability (p_c) = 0.84, mutation probability (p_m) = 0.0054, population size (pop_{size}) = 120, maximum number of generations (Max_{gen}) = 160. Optimized rule base is presented in Table 5 and optimized data base is shown in Figure 11.

Performances of both the approaches are compared among themselves for solving twenty different scenarios. Performance of Approach 1 is found to be better in almost all the scenarios (refer to Table 6). It may be due to the fact that the FLC-based approach is more adaptive. Movement of all the robots for a test scenario (8th of Table 6) is shown in Figures 12. CPU times of the Approaches 1 and 2 are seen to be equal to 0.033 and 0.015, respectively.

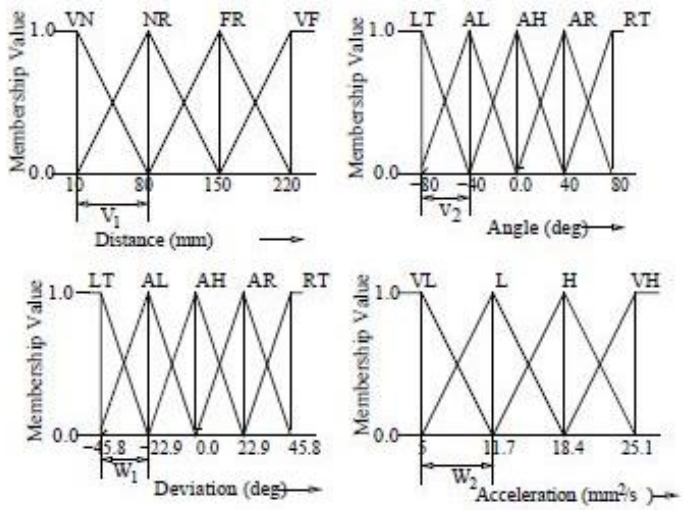

Fig. 11: Optimized membership function distributions of the FLC – Case 2

Table 5: Optimized rule base of the FLC for twelve robots case.

Distance	Angle	Deviation	Acceleration
VN	AL	AH	H
VN	AH	AH	VL
VN	AR	AH	VL
NR	AR	AH	L
FR	AL	AH	VL
FR	AH	AH	VL
FR	RT	AL	H
VF	LT	AH	VL
VF	AH	AH	VL
VF	AR	AH	VL

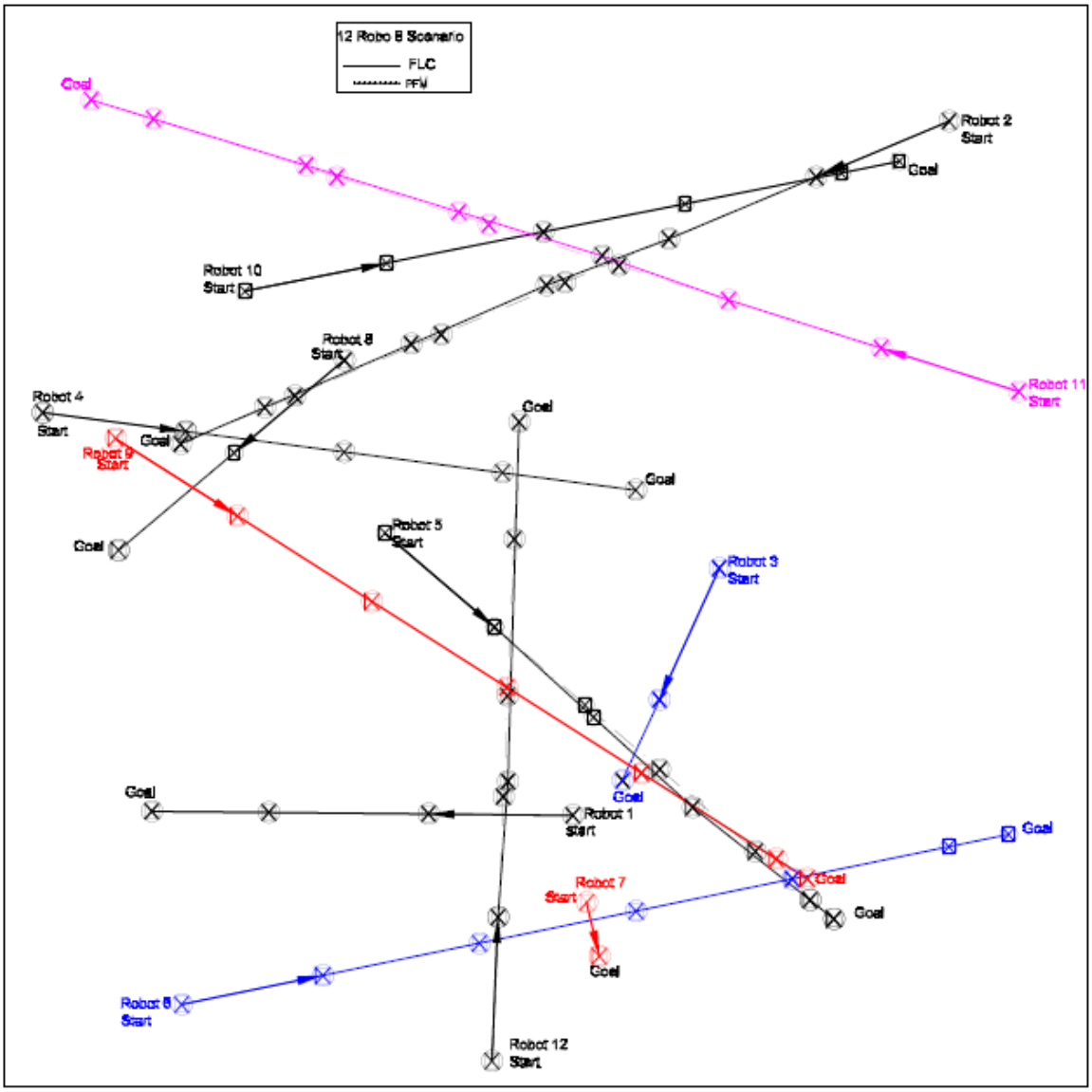

Fig. 12: Movements of the 12 robots in 2D space – 8th Scenario of Table 6.

V. Concluding Remarks

Motion planning problem of multiple robots is solved, in the present work. Two different approaches have been developed for this purpose. In Approach 1, a GA-tuned Mamdani-type FLC has been considered. On the other hand, a modified potential field method as proposed by Ge and Cui Ge and Cui (2000) has been used. The effectiveness of both the approaches are tested through computer simulations for three different cases. In first case eight robots are moving, whereas, ten and twelve robots are negotiating their motion in the next two cases. Performances are compared for twenty different scenarios (randomly generated and are different from the training scenarios). Approach 1 is found to outperform for most of the scenarios in compared to the other. CPU times of both the approaches are found to be low, thus making them suitable for on-line implementations.

Therefore, both FLC as well as PFM-based motion planner might be useful in designing a controller for each agent of a multi-agent system.

Note: This paper is the extended version of "Coordinated Motion Planning of Multiple Mobile Robots Using Potential Field Method" published in 2010 International Conference on Industrial Electronics, Control and Robotics.

References

[1] Alami, R., Chatila, R., Fleury, S. and Ingrand, F. (1998). An architecture for autonomy, International Journal of Robotics Research. 17(4), 315–337.
[2] Asada, M., Kitano, H., Noda, I. and Veloso, M. (1999). Robocup: today and tomorrow– what we have and learned, Artificial Intelligence. 110, 193–214.
[3] Bemporad, A., Luca, A. D. and Orilo, G. (1996). Local incremental planning for a car-like robot navigating among obstacles, Proc. of IEEE Conference on Robotics and Automation, Minneapolis, Minnesoia, pp. 1205–1211.

[4] Bennewitz, M. and Burgard, W. (2000). A probabilistic method for planning collision-free trajectories of multiple mobile robots, Proc. of the Workshop Service Robotics Applications and Safety Issues in an Emerging Market at the 14th ECAI.
[5] Berenji, H. R. (1992). Learning and tuning fuzzy logic controllers through reinforcements, IEEE Transactions on Neural Networks. 3, 724–740.
[6] Borenstein, J. and Koren, Y. (1989). Real time obstacle avoidance for fast mobile robots, IEEE Trans. On Systems Man and Cybernetics. 19(5), 1179–1187.
[7] Buck, S., Weber, U., Beetz, M. and Schmitt, T. (2001). Multi robot path planning for dynamic environments: A case study, Proc. of the IEEE/RSJIROS Conference, Wailea, USA.
[8] Casillas, J., Cordon, O. and Herrera, F. (2000). Learning fuzzy rules using ant colony optimization algorithms, Proceedings of 2-nd International Workshop on Ant Algorithms, Brussels, Belgium, pp. 13–21.
[9] Denna, M., Mauri, G. and Zanaboni, A. M. (1999). Learning fuzzy rules with tabu search an application to control, IEEE Transactions on Fuzzy Systems. 7(2), 295–318.
[10] Fraichard, T. and Scheuer, A. (1996). Car-like robots and moving obstacles, Proceedings of IEEE Conference on Robotics and Automation, pp. 64–69.
[11] Ge, S. S. and Cui, Y. J. (2000). New potential functions for mobile robot path planning, IEEE Transactions on Robotics and Automation. 16(5), 615–620.
[12] Hui, N. B. (2007). Design and Development of Adaptive Robot Motion Planners for Wheeled Robots, PhD thesis, Indian Institute of Technology, Kharagpur.
[13] Hui, N. B., Mahendar, V. and Pratihar, D. K. (2006). Time-optimal, collision-free navigation of a car-like mobile robot using neuro-fuzzy approaches, Fuzzy Sets and Systems. 157(16), 2171–2204.
[14] Jolly, K. G., Ravindran, K. P., Kumar, R. V. and Kumar, R. S. (2007). Intelligent decision making in multi agent robot soccer system through compounded artificial neural networks, Robotics and Autonomous Systems. 55, 589–596.
[15] Jung, D. and Zelinsky, A. (1999). An architecture for distributed co-operative planning in a behavior based multi-robot system, Robotics and Autonomous System. 26, 149–174.
[16] Kasinski, A. and Skrzypczynski, P. (2001). Perception network foe team of indoor mobile robots: concept, architecture, implementation, Artificial intelligence. 14, 125–137.
[17] Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots, The International Journal of Robotics Research. 5(1), 90–98.
[18] Kim, J.-H., Kim, D.-H., Kim, Y.-J. and Seow, K.-T. (2004). Soccer robotics, Springer, Amsterdam.
[19] Koren, Y. and Borenstein, J. (1991). Potential field methods and their inherent limitations for mobile robot navigation, IEEE Conf. Robotics and Automation, Sacramento, CA, p. 13981404.
[20] Latombe, J. C. (1991). Robot motion planning, Kluwer Academic Publishers, Boston, MA, USA.
[21] Lefebvre, O., Lamiraux, F., Pradalier, C. and Fraichard, T. (2004). Obstacle avoidance for car-like robots integration and experimentation on two robots, Proc. of IEEE Intl. Conference on Robotics and Automation,
[22] Lucaterio, C. R., Bueno, A. D. A. and Lozano, R. (2001). A game theory approach to robot tracking problem, WSEAS Transactions on Computers. 3, 862–869.
[23] Maa, Y., W. H. X. Y. G. M. (2014). Path planning for multiple mobile robots under the double - warehouse, Informations Sciences. 278, 357–379.
[24] Mackworth, A. (1993). On seeing robots, computer vision systems, theory and applications. pp. 1–13.
[25] Marapane, S. B., Trivedi, M. M., Lassiter, N. and Holder, M. B. (1996). Motion control of cooperative robotic teams through visual observation and fuzzy logic control, Proc. of IEEE Intl. Conference on Robotics and Automation, pp. 1738–1743.
[26] Nakamura, T. (2006). Development of self-learning vision-based mobile robots for acquiring soccer robot behaviors, Lecture Notes in Computer Science. 1395/1998, 257–276.
[27] Nelson, A. L., Grant, E. and Henderson, T. C. (2004). Evolution of neural controllers for competitive game playing with team of mobile robots, Robotics and Autonomous Systems. 46, 135–150.
[28] Nomura, H., Hayashi, I. and Wakami, W. (1992). A learning method of fuzzy inference rules by descent method, Proceedings of IEEE International Conference Fuzzy Systems, San Diego, CA.
[29] Ota, J. (2006). Multi-agent robot systems as distributed autonomous systems, Advanced Engineering Informatics. 20, 59–70.
[30] Park, K. H., Kim, Y.-J. and Kim, J.-H. (2001). Modular q learning based multi-agent co-operation for robot soccer, Robotics and Autonomous systems. 35, 109–122.
[31] Peng, J. and Akella, S. (2003). Coordinating the motion of multiple robots with kinodynamic constraints, Proc. of IEEE Intl. Conference on Robotics and Automation, Taipei, Taiwan, pp. 4066–4073.
[32] Pham, T. D. and Valliappan, S. (1994). A least squares model for fuzzy rules of inference, Fuzzy Sets and Systems. 64, 207–212.
[33] Pradahan, S. K., Parhi, D. R. and Panda, A. K. (2009). Fuzzy logic techniques for navigation of several mobile robots, Applied Soft Computing. pp. 290–304.
[34] Pratihar, D. K. (2000). Path and Gait generation of legged robots using GA-Fuzzy Approach, PhD thesis, Indian Institute of Technology, Kanpur.
[35] Pratihar, D. K., Deb, K. and Ghosh, A. (1999). A genetic-fuzzy approach for mobile robot navigation among moving obstacles, International Journal of Approximate Reasoning. 20, 145–172.
[36] Qu, H., Xing, K. and Alexander, T. (2013). An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots, Neurocomputing. 120, 509–517.
[37] Rocha, R., Dias, J. and carvalho, A. (2005). Co-operative multi-robot system: A study vision based system 3d mapping using information theory, Robotics and Autonomous Systems. 53, 282–311.
[38] Ros, R., Arcos, J. L., de Mantras, R. L. and Velsco, M. (2009). A case based approach for co-ordinated action selection in robot soccer, Artificial intelligence. 173, 1014–1039.
[39] Sevstka, P. and Overmars, M. H. (1995). Coordinated motion planning for multiple car-like robots using probabilistic road maps, Proc. of IEEE Intl. Conference on Robotics and Automation, pp. 1631–1636.
[40] Skrzypezyk, K. (2004). Multi-robot co-ordination based on co-operative game, Lecture Notes in Computer Science, Artificial Intelligence and Soft Computing - ICAISC 2004. 3070, 798–803.
[41] Springer, G., Hannah, P., Stonier, R., Smith, S. and Wolfs, P. (1997). Simple strategies for collision avoidance in robot soccer, Robotics and Autonomous System. 21, 191–196.
[42] Takagi, T. and Sugeno, M. (1983). Derivation of fuzzy control rules from human operator's control action, Proceedings of IFAC Symposium On Fuzzy Information, Knowledge Representation and Decision Analysis,
[43] Marseilles, France, pp. 55–60.
[44] Uchibe, E., Asada, M. and Hosoda, K. (1998). Cooperative behavior acquisition in multi mobile robots by reinforcement learning based on state vector estimation, Proc. of IEEE Intl. Conference on Robotics and Automation, Leuven, Belgium, pp. 1558–1563.
[45] Vail, D. and Veloso, M. (2003). Multi-robot dynamic role assignment and coordination through shared potential fields, Multi-Robot Systems. pp. 87–98.
[46] Wang, F. Y. and Pu, B. (1993). Planning time-optimal trajectory for coordinated robot arms, Proc. of IEEE Intl. Conf. on Robotics and Automation, Vol. 1, pp. 245–250.
[47] Wang, J. and Xin, M. (2012). Distributed optimal cooperative tracking control of multiple autonomous robots, Robotics and Autonomous Systems. 60(4), 572–583.
[48] Wang, L. X. and Mendel, J. M. (1992). Generating fuzzy rules by learning from examples, IEEE Transactions on Systems, Man and Cybernetics. 22(6), 1414–1422.

[49] Wang, L. and Yen, J. (1999). Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and kalman filtering, Fuzzy Sets and Systems. 101, 353–362.
[50] Warren, C. (1990). Multiple robot path coordination using artificial potential fields, Proc. of the IEEE International Conference on Robotics and Automation (ICRA), pp. 500–505.
[51] Wu, J., Snaˇsel, V., Ochodkova, E., Martinoviˇc, J., Svatoˇn, V. and Abraham, A. (2013). Analysis of strategy in robot soccer game, Neurocomputing. 109, 66–75.
[52] Xi, N., Tarn, T. J. and Bejczy, A. K. (1993). Event-based planning and control for multi-robot coordination, Proc. of IEEE Intl. Conf. on Robotics and Automation, Vol. 1, pp. 251–258.
[53] Yao, B. and Tomizuka, M. (1993). Adaptive coordinated control of multiple manipulators handling a constrained object, Proc. of IEEE Intl. Conf. on Robotics and Automation, Vol. 1, pp. 624–629.
[54] Yupu, Y., Xiaoming, X. and Wengyuan, Z. (1998). Real-time stable self-learning fnn controller using genetic algorithm, Fuzzy Sets and Systems. 100, 173–178.