PATH PLANNING OF MOBILE ROBOT USING REINFORCEMENT BASED ARTIFICIAL NEURAL NETWORK

A. D. Dubey[1], R. B. Mishra[2] and A. K. Jha[3]
[1,2]Dept. of Computer Engineering,
Indian Institute of Technology Banaras Hindu University, Varanasi, India
[3]Dept. of Mechanical Engineering,
Indian Institute of Technology Banaras Hindu University, Varanasi, India

ABSTRACT

In this paper, we have applied a cognitive based artificial neural network which is used to determine a collision free shortest path of a mobile robot from the initial point to the destination in an unknown environment. In this paper we have created an Artificial Neural Network (ANN) which is used to a path by its nonlinear functional approximation. The training samples of this artificial neural network have been collected by a reinforcement learning method known as Q learning. This path planning algorithm has been devised using five state action mapping relationship. The algorithm was applied on our designed mobile robot manipulator and the results show that the path planning done by this method was better than the path devised out by using ANN and Q learning method separately.

KEYWORDS: *path planning, cognitive, artificial neural network, q learning, mobile robot*.

I. INTRODUCTION

In order to train the mobile robot about its environment, machine learning has been quite frequently used. Supervised learning is one of the major learning techniques that train the mobile robot in order to decide the next move in a given environment. However, the supervised learning is often rendered useless once the environment witnesses even a small change. To overcome this learning policy, reinforcement learning is used. Since reinforcement learning rests on the principle reward and punishment and no prior training instances are required, it proves to be quiet efficient in dynamic and static environment.

The problem of motion planning has two parts: path planning and trajectory planning. The path planning involves generation of a collision free path in an environment with obstacles and optimizes it with some criteria [1] [2]. In many cases, the path planning of the environment depends on the sensory inputs of the robot manipulator to deal with the dynamic changes in the environment [3]. Trajectory planning takes into consideration, the time and velocity of the robots while planning its path to the goal.

The path planning of the mobile robot is mainly of two types. If the environment is static and the path is generated in advance, it is said to be off-line algorithm while if the environment is dynamic and the path planning is done along with the motion of the robot to avoid collision with obstacles occurring due to sudden environment changes, it is known as on line path planning. The works in [4] involve mapping, navigation, and planning tasks for Khepera II mobile robot. In these works, the computationally intensive tasks, such as the planning andmapping tasks were not performed onboard. They were performed on a separate computer. The sensor readings andthe motor commands are communicated between the robotand the computer via serial connection.

There are many studies on robot-path planning using variousapproaches, such as the grid-based A*(A-star) algorithm [5], [6], road maps (Voronoi diagrams and visibility graphs) [6], [7], cell decomposition [8], [9], [10] and artificial potential field [11], [12]. Also, various leaning methods have been proposed so far for themobile robot, such as fuzzy logic [14], evolutionary algorithm [15], ANN [16] and reinforcement learning algorithm [17]. Since ANN has the good generalization performance and can approximate any functions in any accuracy, they are considered to be one of the most effective method for solving the problem of nonlinearmapping. However, reinforcement learning (RL) is different from ANNsupervised learning. In reinforcement learning, agent perceives states of environment and learns the optimal mapping from the state to action. Reinforcement learning is able to evaluate the action throughthe reinforcement signal provided by environment. Therefore,the RL is suitable for the application of the control of the intelligent robot. Since reinforcement learning also suffers some problems such as the problems of the delayed reward and temporal credit assignment, with the integration of ANN and RL, the systemmay achieve the advantage of both without many of the limitations of either.

In our paper, we have proposed a cognitive based artificial neural network that is used to plan a collision free path of the mobile robot to the destination. This multi-layer feed forward artificial neural network used the Q learning method to collect the training samples. This ANN has seven outputs which were as follows: move forward, turn left 45º, turn right 45º, turn left 90º, turn right 90º, rotate 180 º and stop. Since the stop action need not to be learned since this action suggests that the task has been performed, only the remaining four actions were need to be learned by the Q learning. The Q table was developed in order to collect the training data for ANN. When the Q learning method finally converges, the training of the artificial neural network is initialized. Since the action performed is always going to be one at a single situation, only one of the five outputs will be true at any given step. This method was applied on the mobile robot manipulator with three links designed by us. The path planning done by this method proved to be better than the ANN and Q learning methods applied separately.

The structure of the paper is as follows: section 2 discusses the Q learning method and the Artificial Neural Network, section 3 describes the design of our mobile robot manipulator. Section 4 describes the implementation of the Q learning and Artificial Neural network in our proposed algorithm, section 5shows the results of the simulation and path planning of the mobile robot manipulator. Section 6 concludes the paper.

II. MECHANISM

2.1. Artificial Neural Networks (ANN)

An artificial neural network (ANN) is a mathematical model or computational model that is inspired by the structure and functional aspects of biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Modern neural networks are non-linear statistical data modeling tools. They are usually used to model complex relationships between inputs and outputs or to find patterns in data.

Neural network based methods have foundconsiderable attention for generating real-time collision free robottrajectories. A probabilistic learning approach to mobile robot path planning was proposed by Svestka and Overmars[18]. Zalama et al presented a neural network model for mobile robotnavigation [19, 20]. With the help of this model, collision free dynamic paths can be generated through unsupervised learning. Quoyet al. reported a dynamic neural network based method forrobot path planning and control [21].In order to avoid time-consuming learning process, a Hopfield-type neural network model forreal-time trajectory generation in anon-stationary environment was proposed by Glasius et al [22]. Similarly, Yang and Meng proposed a neural network approach to dynamic collision-free trajectory generation for robots in dynamic environments [23, 24] which was extended tovarious robot systems [25, 26]. A further distance propagating system was developed for robot path planning [27].

2.2. Q- learning

Reinforcement learning has been tested in many simulated environments [28]-[30] but on a limited basis in real-world scenarios. A real-world environment poses more challenges than a simulated

environment, such as enlarged state spaces [2] increased computational complexity, significant safety issues (a real robot can cause real damage), and longer turnaround times for results.
Q- Learning uses learned action value functions to approximate the optimal action value function independent of the current policy. This simplifies policy evaluation and updating while only making the assumption that all state-action pairs continue to be updated. Unlike sarsa, Q-learning does not need to assume state-action pairs are visited an infinite number of times, making it more likely to reach the optimal policy thansarsa given a finite number of episodes [31].
Algorithms, such as Q-learning [32], can be used to map sensory input states to actions using a goal [33]. Although Q-learning is not biologically plausible, the use of goal driven learning is. For example, in the superior calculus (SC), activity in topographic maps for coincident visual, auditory or somatosensory signals is enhanced by cortical feedback, strengthening the connection between spatially and temporally located events [34].

III. EXPERIMENTAL PLATFORM

Our mobile robotic manipulator has been designed for the purpose of the pick and place task [35] and has total three links. All the actuators used in this project, are servo motors. Servo motors are very light weight, accurate and provide large standing torque even at low voltage ratings. They require a specific PWM signal for operation. By varying the duty cycle of the PWM signal the shaft of the motors can be rotated accordingly.The motor used in the robotic arm are the four mega torque quarter scale servo which provide extreme torque of 20kg.cm at 4.8V and 25kg.cm at 6V. The robotic arm has been made of aluminium sheets which are light weight and provide optimum support to the motors.
The base of the robot manipulator has been made by using fiber glasses which has the dimensions of 30x22 cms and have been separated by a distance of 4.6 cms. There are four wheels which are connected to the base to provide the mobility to the manipulator have a diameter of 3 cms. The distance between the two wheels on one side is approximately 10 cms. These wheels are connected to High Torque DC Geared Motor 300RPM, massive torque of 30Kgcm to make the wheels rotate providing the mobility to the manipulator [36].
The robotic manipulator has a general purpose robotic gripper which has the ability to be used with 2 servo motors for gripper open/close and wrist rotate. This robotic arm can grip and lift up to 200 gm of load in form of small objects. All the servo motors attached to the robotic arm and the dc motors attached to the wheels are driven by the Rhino control board controller which contains ATMega16 microcontroller, used to control any servo based robot.
The current position (x_r, y_r) of the robot is detected by camera sensor. Given the target position (x_g,y_g), the distance between the target and the robot is

$$\sqrt{\left(x_r - x_g\right)^2 + \left(y_r - y_g\right)^2} \quad (1),$$

and the relative angle between them is

$$\theta_{rg} = \arg\ tan\left(\frac{(y_r - y_g)}{(x_r - x_g)}\right) \quad (2).$$

Using an action space to express the seven executable discrete actions, labeled as M={m_f, m_{l45}, m_{r45}, m_{l90}, m_{r90}, m_0, m_s}. The task of path planning can be changed to find a directly mapping relationship, namely {d_{roi}, d_{rg}, θ_{rg}}→M. This relationship can also be seen as a classified problem. Divide the whole state-action mapping relationship into five categories. This can deduce the learning space evidently and increase the convergent speed. Conditions needed for the optimal path and other restrictions are realized by the reinforcement function.

IV. IMPLEMENTATION

Q-learning procedure:

Reinforcement learning is a learning technique based on trial and error. It can be used to learn the unknown desired outputs by providing autonomous mobile robots with suitable evaluation of their performances and also can be used to realize the robots online learning. Online learning and adaptation are desirable traits for any robot learning algorithm operating in changing and unstructured environments where the robot explores its environment to collect sufficient samples of the necessary experience. In a complex environment, it is required perform online learning through interaction with the real environment. In our work, we will focus on learning the path planning behavior. Q-learning, one of the reinforcement learning algorithms has been widely used due to its simplicity and theory maturity.

In this paper the process starts with the initial Q-learning phase and then followed by the training of the artificial neural network. The Q-learning phase consists of the following steps:

(i) Capture Initial State of each Q-Learning Training Cycle:

Capture current state of the mobile robot with respect tothe environment. Each state corresponds to a matrix of28 parameters. State at any given time t, s_t, is represented mathematically as below:

$$s_t = [I_o\ I_1 \ldots\ldots I_{26}\ I_{27} d_{rg} \theta_{rg}] \tag{3}$$

The first two variables I_0 and I_1 denoted the rotations needed for the robot so that it could move in the direction of the destination from the direction it was initially located. The rotation allowed in the algorithm can only be -2 or +2(180 degrees left or 180 degrees right), -1(90 degrees left), +1 (90 degrees right). The combination of I_0 and I_1 was devoted as following:

(I_0,I_1)= (0,0) represents +2 or -2; (I_0,I_1) =(0,1) represents +1; (I_0,I_1)=(1,1) represents 0; (I_0,I_1)=(1, 0) represents -1. (4)

The other bits (I_2 to I_{25}) represent the condition of the map. Considering the whole map as a grid of (MXN), we take a small portion of it (5X5), with the robot at the center of the map. Each coordinate of this map is marked as 1 if the robot can move to this point in the next step, or 0 if the point does not exist or the robot cannot move to this point because of some obstacle. So we have taken a small part of the graph. If (x,y) be the coordinate of the robot, we take (x-2,y-2) ……(x+2,y+2) as the input. There are 24 points, excluding the center of the graph (where our robot stands) are fed into the neural network d_{rg} is the relative distance from the mobile robot's current centerof mass position to the goal position and θ_{rg} is the relative angle from the mobile robot's current forward moving axis to the goal position.

(ii) Quantify the Captured State

Quantification of the captured state reduces the total number of states that the mobile robot is required to travel. The quantification degree, on the other hand, limits the size of the learning space. Excessively largelearning space will take extremely long time for thetraining to be completed, while excessively small learning space will not be sufficient to represent the entire working environment. Thus, the quantification degree depends greatly on the size and complexity of the working environment.

In this paper, the camera sensor readings (I_0-I_{25}) are quantified to two degrees, near (0) or far (1) to themobile robot, depending on whether the target is in the given 5x5 grid or not. The remaining two parameters, the relative distance (d_{rg}) and relative angle (θ_{rg}), are quantified to 6 degrees (0-5). Here, '0' representing smallest relative distance or relative angle from the robot's current position to the goal and '5' the greatest relative distance or relative angle from the robot'scurrent position to the goal.

(iii) Random Action that satisfied Supplied Control Policy (SCP)

At each time step, the robot is required to choose one ofthe seven available actions according to the SCP (O_0,O_1, O_2, O_3, O_4, O_5, O_6 and O_7). These outputs represent namely running forward, turning left, turning right, rotating 180^o and stopping respectively. SCP was developed to reduce the possibility of agent colliding with obstacles during the training process. The detailed rules in SCPare:

1. If any obstacle in front of the robot is close, perform only rotation of the robotic manipulator,
2. If all the obstacles in front of the robot are far and thecurrent angle between the robot's heading and thegoal is small (between -45^o to 45^o), the robot movesforward, and

3. If none of the situation is as described in 1 or 2, theagent is free to randomly choose between the 3actions.

(iv) Capture the Latest State and Quantify the Latest State

Capture the latest state, S_{t+1}, of the mobile robot withrespect to the environment. It is quantified according tothe quantifying scheme as in (ii).

(v) Calculate Numerical Reward/Penalty and Q-Value

The numerical reward or penalty for each action iscalculated and given immediately soon after it isperformed. Followed by the calculation of Q-Value,Q(s, a), is calculated through Bellman's equation. The Value Function, R_t, associated with each stateinformation is calculated using the following equation:

$$R_t = -\sum_{i=0}^{27} w_i d_{roi} + w_{28} d_{rg} + w2_9 d_{rg} \tag{5}$$

Where w_i (i = 0 to 27), w_{28} and w_g are weigh parameters. The values of weighs should reflect the influence degree withregard to the corresponding modular.

$$r_t = \Delta R_t = R_{t+1} - R_t \tag{6}$$

Q-Values could be calculated using the Bellman'sequation. Bellmen's equation is also known as the greedy policy as it tries to evaluate each action not only depending on its effect in the previous time step, butalso on its implications in later time steps. The mathematical form of Bellman's equation is as below:

$$\widehat{Q}_n(s_t, m_i) = (1-\alpha_n)\widehat{Q}_{n-1}(s_t, m_i) + \alpha_n \left[r_{t+1} + \gamma \max_{m_i \in M}(s_{t+1}, m_i) \right] \tag{7}$$

Where α_n is the learning rate, α is the discount rate between 0 and 1.
(vi) Repeat steps (ii) – (v) until the Mobile Robot reaches the goal. The training cycle will terminate if mobile robot collides with any of the obstacle.
(vii) Repeat steps (i) – (vi) until the Q-Value, Q(s, a), converges. The process is recursive until the Q-Values in the Q-Tableremains unchanged.
(viii) Final Optimal State-Action Table is constructed for each state, $a_t{*}(s_t)$, by locating the maximum Q-Value for the particular state.
Because the planning task has been partitioned into seven action space, only six mapping relationship needs to be learned namely $\{d_{roi}, d_{rg}, 0_{rg}\} \rightarrow m_f$, $\{d_{roi}, d_{rg}, 0_{rg}\} \rightarrow m_{l45}$, $\{d_{roi}, d_{rg}, 0_{rg}\} \rightarrow m_{r45}$, $\{d_{roi}, d_{rg}, 0_{rg}\} \rightarrow m_{r90}$, $\{d_{roi}, d_{rg}, 0_{rg}\} \rightarrow m_{l90}$, $\{d_{roi}, d_{rg}, 0_{rg}\} \rightarrow m_o$, while the stopping action mapping $\{d_{roi}, d_{rg}, 0_{rg}\} \rightarrow m_{s.}$ can be simply realized by one reason rule. The size of the learning space is determined by the number of the input variables and their quantified degree.

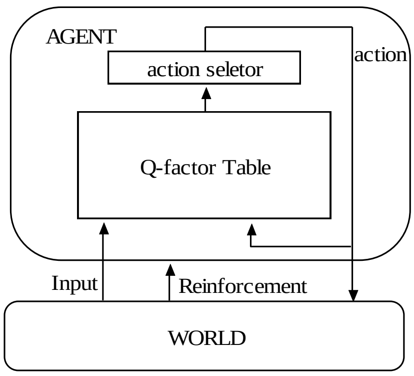

Fig 1. Structure of the learning agent

Learning of the Artificial Neural Network

After the Q look up table has been converged, the state action pairs of the Q lookup table are fed to the ANN and the connection weights are then adjusted. The artificial neural network is trained in order to establish a correspondence between the sensor inputs and the action patterns and decide the output accordingly.

Though there is no effective rule to determine the number of hidden neurons in an artificial neural network, the general experience states that the number can be determined by the formula

$$n = \sqrt{n_i + n_o} + A \tag{8}$$

Where n is the number of hidden neurons in the ANN, $n_i$is the number of input neurons and n_o is the output number. A is a constant between one and ten.

The number of input neurons is 28, the output neurons are 7, therefore according to the equation, and the numbers of hidden neurons were fixed to 10 as shown in the diagram.

The activation function used in the hidden layer is sigmoid function. Steepest Descent Back Propagation has been adopted as the learning algorithm. The robot can take only one action at a given time hence only one output neuron can be 1 and others will be 0. For this purpose, competitive function as the activation function has been selected for the output layer. The connection weights between input layer and hidden layer (W_{ij}) will be updated as:

$$W_{ij}(k+1) = W_{ij}(k) - a\frac{\partial E(k)}{\partial W_{ij}(k)} \tag{9}$$

a is a learning rate, suppose the value is 0.01. E(k) is the total error performance function, selected as the mean square error.

The learning algorithm of the connection weighs between the hidden layer and the output are updated according to the winner takes all (WTA) rule, where weights are modified onlyfor the neuron with the highest net value. If the i_{th} neuron wins, then the i_{th} row elements of the connection weights vector are adjusted using formula (10). The weights are labeled as W_{mn}.

$$W_{mi}(k+1) = W_{mi}(k) + a[p(k) - W_{mi}(k-1)] \tag{10}$$

Where a is learning rate. p(k) is the input vector of the output layer. By learning, those weight values which are nearest to the input vector are updated and make them to near the input vector much more. The result is that in the next time when input the similar vectors the victorious neuron have more chances to win. But to the weights which are far away from the input vector will have little and little probability to win. After more and more training, input modes which have the similar characters will make the corresponding neuron outputs 1, others output 0.

V. RESULTS AND FUTURE SCOPE

The path planning algorithm was checked using MATLAB simulation. Since the SCP (supplied control policy) assured that the path planned for a mobile robot is collision free and efficient in terms of time taken, this method of path planning is supposed to be quite efficient for the path planning problem of the mobile robot. The application of the artificial neural network in this method increases the flexibility of the controller without which the Q learning method may been a failure in an environment where the position of the obstacles changes frequently. To simulate the given method of path planning, the initial point and the destination point were defined randomly while the camera sensor mounted in the environment was used by this method to detect the points of collision and the destination. The Q table stored all the Q data and state-action pairs. The training of the artificial neural network was done after the Q table converges i.e. there is no change in the Q table on simulation.

The supplied chain policy was helpful in exploring the environment which ensures that the robot avoids the collision. After the training is done, the Q table is updated according to the present state action pairs. The training was done for 15 more cycles until an optimized path was obtained. The

entire training cycle was repeated till we concluded that the size of the state table, Q-Table and the shortest route was same for three consecutive cycles of training.
On simulation of this method the Q values were found to converge after 83 cycles. Each training cycle stops when the robot reaches its goal without any collision with the obstacles present in the environment. In this process the robot collected 1021 states of the path planning which mapped the entire surroundings.
The training of the artificial neural network begins after the Q table has converged i.e. the Q learning table has reached to its maximum learning ability. The artificial neural network used in this method was back propagation feed forward network. On the testing of this artificial neural network, it was found that approximately 786 states out of 1021 (77%) produced by this neural network were identical to the actual optimal action obtained in the first training phase.
The proposed Artificial Neural Q-Learning training algorithm was found effective in finding a collision free path. The SCP successfully prevented the simulated robot/agent from colliding with any obstacle during the whole training process thus reducing the total training duration. As compared to the results obtained by applying only the Artificial Neural Network in fig 2, the path devised out by this method in figure 3 was shorter and collision free also.

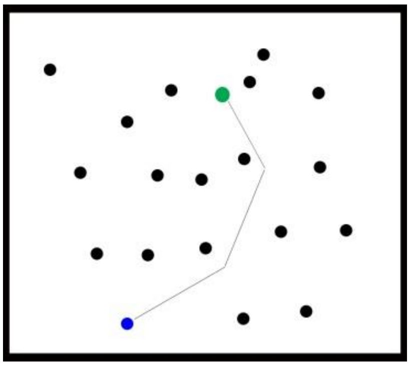

Fig 2 Results of the simulation of the path planning using only Artificial Neural Network

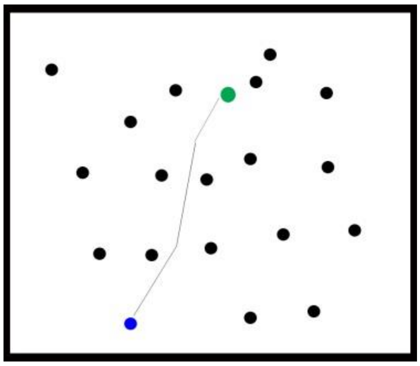

Fig. 3 Results of the simulation of the path planning using Q-learning and Artificial Neural Network

The path planning method implemented in this work can be used extended to use it for multi-robot environment which contains more than one object. The artificial neural network used can be trained using large data sets from environments which are dynamic and contain large number of obstacles which will increase the accuracy of the algorithm.

VI. CONCLUSION

In this paper we have proposed a cognitive based artificial neural network that develops a collision free path from the starting point to the destination.Combining ANN with Q learning can overcome the disadvantages of low learning speed by only using ANN method and impossible abilities to learn all of states by Q-learning. By using method, the path planning can be done without any prior knowledge of the environment while it also ensures that any minor changes in the environment will not affect effective path planning of the mobile robot.

REFERENCES

[1] Dean, T., Basye, K. and Shewchuk, J. “Reinforcement learning forplanning and Control”. In: Minton, S. (ed.) Machine LearningMethods for Planning and Scheduling. M$_{\theta rg}$an Kaufmann 1993.
[2] Bellman, R.E., “Dynamic programming “, Princeton, NJ: Princeton, University Press, p. 957.
[3] Gerke, M., and Hoyer, H., Planning of Optimal paths for autonomous agents moving in inhomogeneous environments, in: Proceedings ofthe 8th Int. Conf. on Advanced Robotics, July 1997, pp.347-352.
[4] Busoniu, L., Babushka, R., Schutter, B.De., Ernst, D., ReinforcementLearning and Dynamic Programming Using FunctionApproximators, CRC Press, Taylor & Francis group, Boca Raton, FL,2010.

[5] Hart, P.E., Nilsson, N.J., Raphael, B.: A Formal Basis for the Heuristic Determinationof Minimum Cost Paths. IEEE Trans. Syst. Sci. Cybern. 4 (1968) 100-107

[6] Warren, C.W.: Fast Path Planning using Modified A* Method. Proc. IEEE Int.Conf. Robotics and Automation, Atlanta, GA (1993) 662-667.

[7] Latombe, J.C.: Robot Motion Planning. Boston, MA: Kluwer (1991)

[8] Takahashi, O., Schilling, R.J.: Motion Planning in A Plane using GeneralizedVoronoi Diagrams. IEEE Trans. Robot. Autom. 11 (1989) 143-150

[9] Hou, E., Zheng, D.: Mobile Robot Path Planning based on Hierarching HexagonalDecomposition and Artificial Potential Fields. J. Robot. Syst. 11 (1994) 605-614.

[10] Schwartz, J.T., Sharir, M.: On the Piano Movers' Problem: I. The Case If aTwo-Dimensional Rigid Polygonal Body Moving Amidst Polygonal Barriers. IEEETrans. Robot. Autom. 36 (1983) 345-398

[11] Leven, D., Sharir, M.: An Efficient and Simple Motion Planning Algorithms for aLadder Moving in Two-Dimensional Space Amidst Polygonal Barriers. Proc. 1stACM Symp. Computational Geometry, Nice, France (1997) 1208-1213

[12] Khatib, O.: Real-Time Obstacle Avoidance for Manipulators and Mobile Robots.Int. J. Rob. Res. 5 (1986) 90-98

[13] Chuang, J., Ahuja, N.: An Analytically Tractable Potential Field Model of FreeSpace and Its Application in Obstacle Avoidance. IEEE Trans. Syst., Man, Cybern.B 28 (1998) 729-736

[14] Meng Wang, James N.K. Liu, "Fuzzy logic based robot path planning inunknown environment," Proceedings of the fourth internationalconference on machine learning and cybernetics, Guangzhou, 18-21,pp.813-818, August 2005.

[15] Lucidarme, Philippe, "An evolutionary algorithm for multi-robotunsupervised learning, " 2004 Congress on Evolutionary Computation,CEC2004, v 2, Proceedings of the 2004 Congress on EvolutionaryComputation, CEC2004, pp.2210-2215, 2004.

[16] Shou-tao Li, Yuan-Chun Li, "Neuro/Fuzzy behavior-based control of amobile robot in unknown environments, " Proceedings of the ThirdInternational Conference on Machine Learning and Cybernetics,Shanghai, 26-29, pp.806-811, August 2004.

[17] Hong Suh, Min Jo Kim, Sanghoon Lee and ByungJu Yi, "A NovelDynamic Priority-based Action-Selection-Mechanism Integrating aReinforcement Learning," Proceedings of the 2004 IEEE InternationalConference on Robotics & Automation, New Orteans, LA, pp.2639-2646, April 2004.

[18] P. Svestka and M. H. Overmars, Motion planning for carlike robotsusing a probabilistic approach, International Journal of RoboticsResearch, vol. 16, no. 2, 1997, pp119-145

[19] E. Zalama, P. Gaudiano, and J. L. Coronado, A real-time, unsupervisedneural network for the low-level control of a mobile robot in anonstationary environment, Neural Networks, vol. 8, 1995, pp103-123

[20] E. Zalama, J. Gomez, M. Paul and J. R. Peran, Adaptive behaviornavigation of a mobile robot, IEEE Transactions on Systems, Man, andCybernetics-Part A: Systems and Humans, vol. 32, no. 1, 2002, pp160-169

[21] M. Quoy, S. Moga and P. Gaussier, Dynamical Neural Networks forPlanning and Low-Level Robot Control, IEEE Transactions onSystems, Man, and Cybernectics-Part A: Systems and Humans, vol. 33,no. 4, 2003, pp523-532

[22] R. Glasius, A. Komoda, and S. C. A. M. Gielen, Neural networkdynamics for path planning and obstacle avoidance, Neural Networks,vol. 8, no. 1, 1995, pp125-133

[23] S. X. Yang and M. Meng, An efficient neural network approach todynamic robot motion planning, Neural Networks, vol. 13, no. 2, 2000,pp143-148

[24] S. X. Yang and M. Meng, Neural network approaches to dynamiccollision-free robot trajectory generation, IEEE Transactions onSystems, Man, Cybernetics-Part B: Cybernetics, vol. 31, no. 3, 2001,pp302-318

[25] S. X. Yang and M. Q.-H. Meng, Real-time collision-free 3, pp1541-1552

[26] S. X. Yang and C. Luo, A neural network approach to completecoverage path planning, IEEE Transactions Systems, Man,Cybernetics-Part B: Cybernetics, vol. 34, no. 1, 2004, pp718-725

[27] A. Willms and S. X. Yang, An Efficient Dynamic System for Real-Time Robot-Path Planning, IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol. 36, no. 4, 2006

[28] Konar, A., Computational Intelligence: Principles, Techniques and Applications. Springer-Verlag, 2005

[29] Bellman, R.E., "Dynamic programming ", Princeton, NJ: Princeton University Press, p. 957.

[30] Busoniu, L., Babushka, R., Schutter, B.De., Ernst, D., ReinforcementLearning and Dynamic Programming Using FunctionApproximators, CRC Press, Taylor & Francis group, Boca Raton, FL,2010.

[31] Smart, W. and Kaelbling, L. Effective Reinforcement Learning for Mobile Robots. Proceedings of the 2002 IEEE International Conference on Robotics &Automatons,Washington, DC, 2002.

[32] C.J.C.H.Watkins and P.Dayan, "Q-Learning," Machine Learning, vol.8, pp. 279-292, 1992.

[33] A.Burkov and B.Chaib-draa, "Adaptive Play Q-Learning with Initial Heuristic Approximation," in Proceedings of the IEEE International Conference on Robotics and Automation, 2007, pp. 1749-1754.
[34] B.E.Stein and M.A.Meredith, The Merging of the Senses. Cambridge, MA.: A Bradford Book, MIT Press, 1993.
[35] Dubey, A.D., Mishra, R.B. Jha, A.K. Design and Path Planning of a Mobile Robot. International Journal pf Computer Science and technology (IJCST), 2012, 3(2), VER 1, 0976-8491, Pg. 73-77.
[36] Tehrani, A.M.,Kamel,M.S. A methodology for specifying and designing behaviors for mobile robots; Proceedings of World Automation Congress.2004.

Akash Dutt Dubey completed his Bachelor of Technology degree in Computer Science and Engineering from Greater Noida Institute of Technology (an Institute affiliated to Uttar Pradesh Technical University) with First division (71.7%) in 2009. He has completed his Ph.D. course work with 8.25 CGPA which is Excellent Grade in Indian Institute of Technology, (BHU), Varanasi. His eight research papers in International and national journals and International and national conferences have been published (accepted). His area of interest includes artificial intelligence, robotics and image processing

R.B. Mishra is working as Professor & Head in Department of Computer Engineering, IT-BHU Varanasi, India. His areas of Research interest are Artificial Intelligence and multiagent systems: Their applications in medical computing, machine translation, e-Commerce, robotics and intelligent tutoring system.

A. K. Jha is working as Professor in Department of Mechanical Engineering, IT-BHU Varanasi, India. His area of Interest are Manufacturing process and Manufacturing System.