

Path Planning Strategy of a Mobile Robot with Bug Group of Algorithms and Obstacle Avoidance using Advance Laser Sensor

Pabitra Kumar Nandia¹ and Dr. Ajoy Kumar Dutt²

¹⁻²Research Scholar, Departments of Production Engineering, Jadavpur University, Kolkata, West Bengal
Emails: pabitra.knandi@gmail.com, ajoydutta@yahoo.co.in

Abstract— Path planning is one of the basic problems of Autonomous Mobile Robot. The mobile robot supposed to be able to work in an unfamiliar situation using an automatic plan determined by information sensed locally. In the case of real-time environments speed calculation and rescheduling of path is essential to bypass the moving obstacle and make collision free path of robot. Robot movement planning in dynamic condition requires the actions to be selected under real-time control. To avoid run-time obstacles a various approach is presented in this paper based on Bug Algorithm. To calculates the distance from obstacles mostly Ultrasonic sensor may be used. In this paper we implemented new sensors approach that is Laser sensor (LIDAR-Lite v3 s). Using this sensor, we get more accurate result that helps to smooth movement of mobile robot path planning.

Keywords— Autonomous Mobile Robot, Path Planning, Obstacle Avoidance, Bug algorithm, Advance Laser Sensors.

I. INTRODUCTION

Autonomous robots currently used in the real world have been one of the major research areas in robotics and artificial intelligence. One of the main problems in robotics, called path planning [1] of robot, is to find a free path clearing obstacles for a robot from its starting position to its destination. Obstacle avoidance [2] is the primary requirement for any autonomous mobile robot which requires integration and coordination of many sensors and actuators. The robot acquires information about its surrounding through various sensors mounted on the robot. The research in this field can be classified into two major areas: the global path planning and the local motion planning. In global path planning, coordinates of the starting point, destination point, and the obstacles are given to the robot in advance. The robot path in such applications can be calculated using this information [12][13]. Local motion planning methods require less prior knowledge about the environment. The robot is dynamically guided on the basis of information about the locally sensed obstacles. In local planning, the position of target point from its current position must be known to robot to ensure that robot can reach the destination accurately [15].

The physical measured value is converted into an analogue electrical signal by a laser sensor, a measurement value recorder that uses laser technology. In other words, the laser sensor was designed for contactless measurement. The triangulation concept underlies how the laser sensor operates. Without making touch, you may use a laser sensor to detect distances, locations, and the length of a road.

A very high resolution is used for this. Along with having different resolutions, laser sensors also have different linearities. There are laser sensors made specifically for metallic, tarnished, or dark surfaces. Because of the integrated intelligent signal analysis, the laser sensor can provide an accurate result. The laser sensor is able to take finer measurements than light-diode-based sensors because of the laser beam that is emitting that is incredibly focused. A line laser or a point laser contains these sensors. The difference between a line laser and a point laser is that the former projects one or two fixed lines. The lines are created using a fixed lens that is incorporated. Laser pointers can have one or many points. A laser sensor's tiny size and exceptional robustness make it feasible to include it into very small machines, such as industrial robots.

II. LOCAL PATH PLANNING AND BUG GROUP OF ALGORITHMS IN UNKNOWN ENVIRONMENTS

The Bug algorithms are simple algorithms which are well known mobile robot navigation methods for local path planning with minimum sensor without generating a full map of the environment. In these algorithms the robot takes an action on the basis of current percepts of sensors without taking into account the previous path and actions. The aim of the Bug algorithms is to guide a robot starting from a source (S) to a target (T) given that the robot has no knowledge of the complete environment [8]. The robot should achieve this goal with as little global information as possible. When they face an unknown obstacle, they are able to easily produce their own path contouring the object in the 2D surface if a path to the goal exists. The purpose is to generate a collision-free path by using the boundary-following and the motion-to-goal behaviors [14][16][17][18]. In addition, the Bug's family has three assumptions about the mobile robot: i) the robot is a point, ii) it has a perfect localization, and iii) its sensors are precise. The following algorithms from the Bug family have been implemented and evaluated:

A. Bug1 Algorithm

The Bug1 algorithm was the first algorithm in the bug family created by Lumelsky and Stepanov[3]. Its principle is to advance toward the target along straight line until encountering an obstacle or reaching the target. When an obstacle is encountered the robot follows its boundary until encountering the hit point again, it means after its one complete circle, it restarts its motion around obstacles until reaches to leaving point. Then, the robot goes to the nearest boundary's point to the target and continues its path towards the goal shown in Fig1.

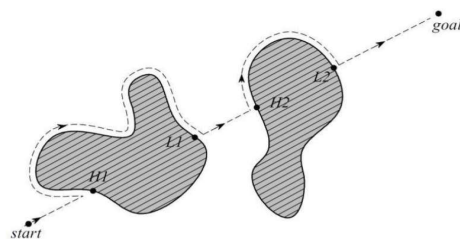


Fig. 1: Path generated by Bug1 Algorithm

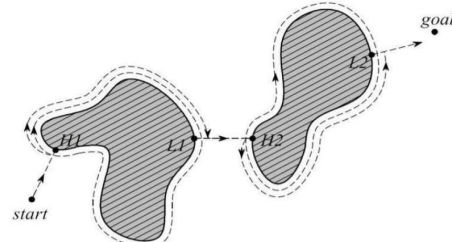


Fig. 2: Path generated by Bug2 Algorithm

B. Bug2 Algorithm

Bug2 algorithm is almost similar to Bug1 algorithm. But in this algorithm the mobile robot doesn't follow the whole boundary of obstacle. It follows the obstacle edge and calculating a new slope from every new position until the new slope matches the original slope. When this slope becomes equal to slope of initial path, the behavior of the robot is changed to move to goal. The robot follows single non-repeated path throughout its trajectory. Bug2 algorithm is more efficient than Bug1 algorithm as it allows the robot to reach the destination in less time following a short trajectory [4]. The robot heads towards the destination on the m-line shown in Fig.2 which is defined as the line joining the start location and the destination. If an obstacle is encountered, the robot follows it until the time it again encounters the m-line again closer to the goal. The robot then leaves the obstacle and again continues toward the goal on the m-line

C. Tangent Bug Algorithm

Kamon, Rivlin and Rimon [6] introduce the Tangent Bug algorithm which is a range sensor-based navigation algorithm for autonomous robots with two degrees of freedom. This algorithm uses the range sensor data to compute a locally shortest path based on the local tangent graph. Tangent Bug is one of the most frequently used algorithms for obstacle avoidance for sensor based mobile robots. In presence of static obstacles, the Tangent Bug algorithm can be used to plan a path from the start to the goal.

Tangent Bug algorithms principle is to advance towards target in straight line, and moving around encountered obstacles. The robot stops moving around obstacle and continues its path towards target once it finds a point on local tangent graph (LTG) closer to target than boundary's one. One advantage of tangent bug algorithm is that there is no need for wall following since the robot only turns on the spot and travels in straight lines. The generated path of Tangent bug algorithm as shown in Fig. 3.

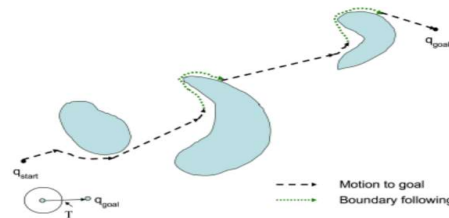


Fig. 3: Path generated by Tangent Bug Algorithm

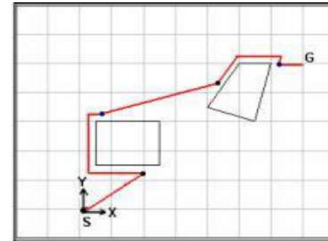


Fig. 4: Trajectory of Dist-Bug algorithm

D. Dist-Bug algorithm

This bug algorithm [5] was created by Kamon and Rivlin and originated from Bug-1 and Bug-2 algorithms. They use different data structures to store the hit and the leave points together with the followed path. In this algorithm the robot faces an obstacle in path, it starts following the edge of obstacle simultaneously and it calculates the distance from every point. The point with the minimum distance is known as leaving point. The leaving point, where it switches the behavior from obstacle avoidance to move to goal, is selected based on the condition that the distance of destination from its next position is greater than the corresponding distance from its current position i.e. ($d_{next} > d_{current}$). When it finds the leaving point during its motion around an obstacle, it generates a new path and starts following it until reaches to destination. The trajectory of Dist-Bug algorithm is shown in Fig. 4.

E. Point Bug Algorithm

The Point Bug method moves a "point" robot through an unknown environment on a 2D plane that is dotted with immovable objects of every description [7]. The term "sudden point" is a new one that this new algorithm introduces. The term "sudden point" refers to the location where a sudden shift in the sensor's range is identified. This method selects the next point to move from a starting point towards the destination. The output of the range sensor, which detects a rapid shift in the distance between the sensor and the closest obstacle, identifies the subsequent point. The abrupt shift in range sensor output is taken into account when determining whether the distance is growing or decreasing.

There are three potential modifications: It can range from (a) infinity to a specific value to (b) a specific value to a specific value to (c) a specific value to a specific value. Robot starts off staring straight ahead towards the goal point before rotating to the left or right in search of a sharp point. Following the detection of a sudden point, the robot rotates in the direction indicated by the location of the straight line between the current sudden point and the target point, also known as the d_{min} line (The line that connects the current point to the target point). The robot always rotates in the direction of the d_{min} line, i.e., it moves in the direction of the abrupt point that is at the smallest angle from the d_{min} line. In order to prevent detection of the prior sudden point, which would cause the robot to return to the previous sudden point from its present place, the robot always disregards the sensor reading during rotations of 1800 and above. The objective is deemed unavailable and the robot quits right away if no abrupt point can be discovered in a single 3600 revolution.

By continually gathering data from the range sensor and adjusting its direction to the target, this algorithm may also work in a dynamic environment. The Point Bug method works well in environments with few barriers and no spiral. The algorithm's performance is dependent on how many sudden points are found; the less sudden points found, the better the performance. A range sensor is shown scanning a pentagon-shaped barrier from A to E in Fig. 5, and a graph of the distance the sensor measured in cm is shown from A to E. The C line has the shortest observed distance to the obstruction since it is perpendicular to the impediment's surface. The distance between points C and B and C and D keeps becoming longer. The amount of distance suddenly doubles from point B to point A on the graph and increases abruptly from point D to point E from a few centimeters to infinity. Points A and E are the abrupt points and are thought to be the locations to which the robot will go to reach the following point.

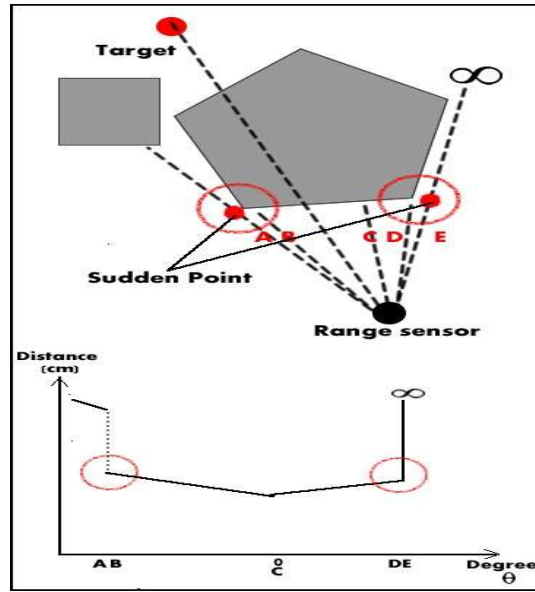


Fig. 5: Range sensor detecting and scanning an obstacle from left to right

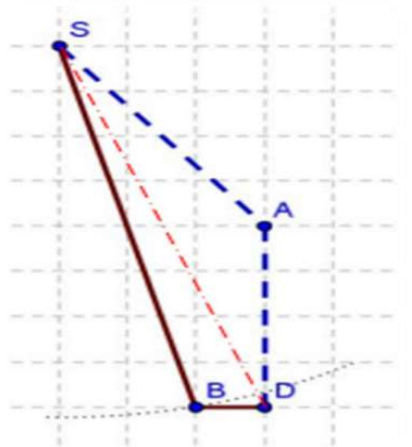


Fig. 6: Minimum angular deviations does not guarantee optimal path

The Point Bug method has a few drawbacks, some of which are listed below.

- The Point Bug algorithm does not remember visited sudden points, therefore there are no checks to see if a sudden point has already been addressed. This could cause an endless cycle to form.
- As shown in Fig. 6, choosing the location that minimizes the angular departure from the intended path does not result in the best path. Although the path length from S to D via point A is less than the path via B, the deviation angle generated by point A is bigger than that formed by point B.

F. Modified Point Bug Algorithm

As has been discussed in the previous section, and shown in Fig. 7, main limitation of Point Bug algorithm is that the selection of a 'sudden point', which minimizes the angular deviation relative to target direction (i.e., the dmin line) does not produce the optimal path in all cases.

In a modified algorithm (24), instead of choosing the sudden point (as the next point to move) which minimizes the angular deviation relative to target direction, the sudden point which is at a shortest distance from goal, is chosen as the next point to move. That is, the sudden point which is closest from the target (goal) is selected. The idea is to go as close as possible towards goal in an unknown environment where only the nearest obstacle towards the target may be detected by the range sensor.

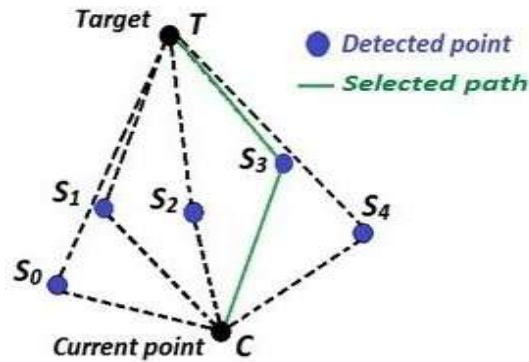


Fig. 7: Strategy of modified Point Bug algorithm

Fig. 7 shows the strategy of the modified Point Bug algorithm. In this figure, C and T are respectively current point and target point of mobile robot. And S0, S1, S2, S3, S4 are the sudden points which are detected by the range sensor. Then there are total five possible paths to reach target point. Thus the algorithm will select the point S3, since it is closest from the target. This has been shown in the figure 7. So here the selected path is C-S3-T.

G. Critical Point Bug Algorithm

This method aids in guiding a robot across a space where there are static barriers of unknowable size, shape, and placement. The robot employs a range sensor to identify sudden changes in distance and locate obstacles. The next point to go from the present location to the target is calculated based on the positions of the obstacles [9]. Consider a wheeled robot that has sensors to identify obstructions. A global frame of reference is used to determine the robot's starting coordinates. We resolve the issue in the configuration space when a point is used to represent the robot. When compared to other approaches, the Critical-Point Bug algorithms have the following advantages: (i) Few iterations are needed to find the target. (ii) Knowledge of the surroundings is not necessary. (iii) Only barriers that might potentially result in a collision will be taken into account when determining the sub objective and critical point. (iv) Calculating the coordinate points is simple. The algorithm is not meant to work in dynamic situations where obstacles shift about as the robot is moving.

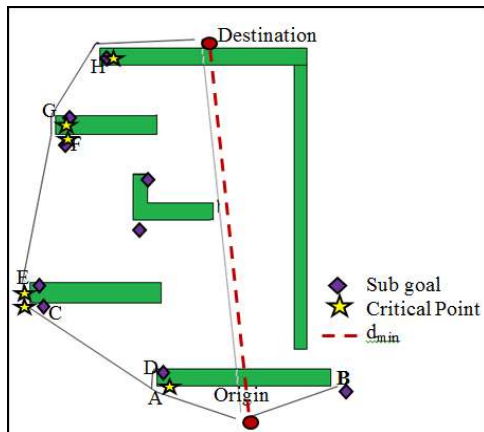


Figure 8. Trajectory of a robot using Critical-Point Bug Algorithm

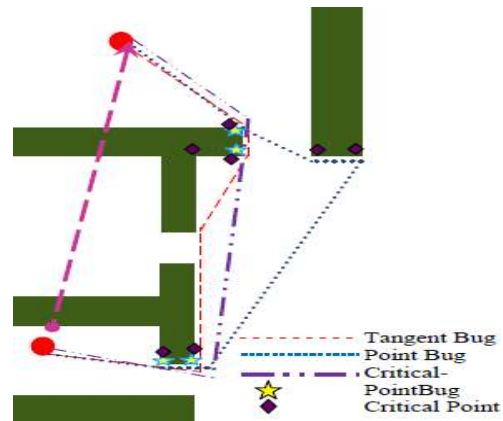


Figure 9. Trajectories formed by different Bug Algorithm

H. Co-operative Critical Point Bug Algorithm

The Bug algorithms are widely employed in local path planning and mobile robot navigation with few sensors. They are absolute, straightforward, and simple. The robot utilizes as little global information as it can to achieve its objective. It is fairly easy to understand the main logic: a) Continue moving towards the goal until a barrier is seen. b) If there is a barrier, remove it until the objective is once again attainable. c). If the objective point is visible, travel back in that direction. There are more than twelve different bug algorithms in the family, and each approach has its unique concluding procedure.

The Critical Point Bug algorithm always chooses a path that avoids obstacles by locating and going to a point that is more near the objective point than other points. Figure 1 depicts the Critical Point Bug and other bug algorithms' paths. The picture makes it evident how our approach outperforms previous bug detection techniques. Although this technique is effective, there are some drawbacks as well. Robot is taken as the point robot by the algorithm. However, a robot in the actual world must have dimension and can be either 2D or 3D. In the face of impediments that are static in nature, local path planning is used. The algorithm focuses on creating a local route while avoiding both stationary and moving impediments. Although the robot is a point robot, the method was also created from the Modified Critical Point Bug algorithm, which may be utilized to avoid dynamic obstacles. To discover the sub goal points and critical point for avoiding obstacles and moving to the next point to attain the goal, the Operative Critical Point Bus algorithm employs references from [10][11].

III. LASER SENSORS APPROACH

A. Introduction

Light Detection and Ranging, or LiDAR, is an optical distance measuring sensor with outstanding performance. In LiDAR, laser light is emitted from a transmitter and reflected by things in the environment. The time of flight (TOF) is employed to create a distance map of the scene's objects when the system receiver picks up the reflected light. Optical distance sensor with high performance: The LIDAR-Lite v3 soars when weight and space restrictions are severe. For applications involving drones, robots, or autonomous vehicles, it is the appropriate small, high-performance optical remote measurement sensor. The Function of Lidar Pulsed light waves are released into the surroundings by a standard lidar sensor. These pulses return to the sensor after being reflected by the environment's objects. The sensor makes advantage of time. The sensor determines the distance travelled by each pulse by measuring the time it takes for it to return to the sensor.



Figure 10: Image of LIDAR-Lite v3 Laser Sensor

B. Sensor Technology

This gadget determines distance by timing the interval between a Near-Infrared laser signal's broadcast and reception after reflecting off of a target. By using the established speed of light, this converts into distance. Our innovative signal processing method sends a coded signature and searches for it in the return, enabling very efficient detection at eye-safe laser intensity levels. High sensitivity, speed, and accuracy are attained in a compact, low-power, and inexpensive device thanks to the application of proprietary signal processing algorithms.

C. Procedure of measurements

- This device initially runs a receiver bias correction process before taking a measurement, compensating for shifting ambient light levels and enabling maximum sensitivity.
- A reference signal is then sent straight from the transmitter to the receiver by the device. The time delay for "zero" distance is established, the transmit signature is stored, and the time delay is regularly recalculated after many measurements.
- The instrument then starts a measurement by carrying out a string of acquisitions. The primary laser signal is sent during each acquisition, and the return signal is captured at the receiver. The outcome is kept in memory as a correlation record if there is a signal match. The following acquisition is added to the prior outcome. These repeated acquisitions result in a peak emerging from the noise at the relevant distance position in the correlation record whenever a distant object reflects the laser signal back to the instrument.

- Until the signal peak in the correlation record reaches its maximum value, the gadget integrates acquisitions. If the received signal is not strong enough for this to happen, the gadget reaches a specified maximum acquisition count before shutting off.
- Until the signal peak in the correlation record reaches its maximum value, the gadget integrates acquisitions. If the received signal is not strong enough for this to happen, the gadget reaches a specified maximum acquisition count before shutting off.

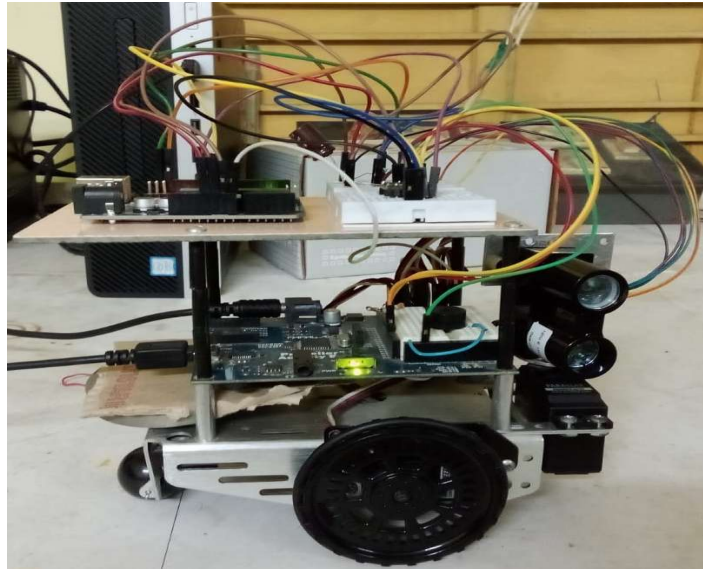


Figure 11: Image of Activity Bot Mobile Robot with connection of LIDAR-Lite v3 Laser Sensor

IV. CONCLUSION

We presented a simple sensor-based path planning algorithm to make a robot reach a specified destination from a given start location, in a region occupied by unknown obstacles. In this paper Bug Group of path planning algorithms are studied and the brief idea of laser sensors with proper technical analysis is presented. We make a model of Activity Bot Mobile Robot with hardware connection of LIDAR-Lite v3 Laser Sensor instead of Ultrasonic sensor. In future we may apply advance level of Bug group of algorithms into Activity Bot Mobile Robot.

REFERENCES

- [1] Haojie Zhang, Guangming Xiong, Bo Su, Jianwei Gong, Yan Jiang, Huiyan Chen and Wei Lan, "Anytime Path Planning in Graduated State Space", IEEE Intelligent Vehicles Symposium (IV), 2013, pp. 358 – 362
- [2] Junghee Park, Jeong S. Choi, Jimin Kin, and Beom H. Lee, "Moving Obstacle Avoidance for a Mobile Robot", IEEE International Conference on Control and Automation, 2009, 367 – 372
- [3] Lumelsky, V.J. & Stepanov, A. A. (1987). Path-planning strategies for a point mobile automation moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, 2(1-4), 403-430.
- [4] Meddah, F., & Dib, L. (2015), E-bug: New bug path-planning algorithm for autonomous robot in unknown environment. In proceedings of the international Conference on Intelligent Information Processing.
- [5] Kamon, L, & Rivlin, E. (1997). Sensory-based motion planning with global proofs. *IEEE transactions on Robotics and Automation*, 13(6), 814-822.
- [6] Banerjee, S. (2014). A comparative study of underwater robot path planning algorithms for adaptive sampling in a network of sensors, 1-159.
- [7] Devi, B. M., & Prabakar, S. (2013). Dynamic Point Bug Algorithm for Robot Navigation, *International Journal of Scientific & Engineering Research*, 4(4), 1276-1279.
- [8] Bhanu Chander V, Asokan T, Ravindran B, "A New Multi-Bug Path Planning Algorithm for Robot Navigation in Known Environments", 2016 IEEE Region 10 Conference (TENCON), 3363 – 3367
- [9] Dr. Ajoy Kumar Dutta, Mr. Subir Kumar Debnath, Mr. Subir Kumar Das, "Local Path Planning of Mobile Robot using Critical-Point Bug Algorithm Avoiding Static Obstacles", *International Journal of Robotics and Automation*, vol. 5(3), 182-187

- [10] Ajoy Kumar Dutta, Subir Kumar Debnath. Subir Kumar Das” Operative Critical Point Bug algorithm-local path planning of mobile robot avoiding obstacles”, International Journal of Electrical Engineering and Computer Science, Vol. 18, No. 3, June 2020, pp. 1646-1656.
- [11] Ajoy Kumar Dutta, Subir Kumar Debnath. Subir Kumar Das “Path Planning of Snake-Like Robot in Presence of Static Obstacles Using Bug Algorithm”, Advances in Computer, Communication and Control, Springer, 2018, 449 - 458
- [12] Muhannad Mujahed, Dirk Fischer, Bärbel Mertsching, “Admissible gap navigation: A new collision avoidance approach”, Robotics and Autonomous Systems 103, 2018, 93 – 110
- [13] Robert L. Williams II, Jianhua Wu, “Obstacle Avoidance for an Omnidirectional Mobile Robot”, Journal of Robotics, Vol. 2010, Article ID 901365
- [14] Kamilah Taylor, Steven M. LaValle, “I-Bug: An intensity-based bug algorithm”, IEEE International Conference on Robotics and Automation, 2009, 3981-3986
- [15] T. Nayl, G. Nikolakopoulos, T. Gustafsson, "Real-time bug-like dynamic path planning for an articulated vehicle" in Informatics in Control Automation and Robotics, Springer, 2015, 201-215
- [16] Nishant Sharma, Shivam Thukral, Sandip Aine, and P.B. Sujit, “A virtual bug planning technique for 2D robot path planning”, Annual American Control Conference, 2018, 5062 – 5069
- [17] Bhanu Chander V, Asokan T, Ravindran B, “A New Multi-Bug Path Planning Algorithm for Robot Navigation in Known Environments”, 2016 IEEE Region 10 Conference (TENCON), 3363 – 3367
- [18] Jinho Kim, Jangmyung Lee, “An Active Virtual Impedance Control Algorithm for Collision Free Navigation of a Mobile a Robot”, IAES International Journal of Robotics and Automation (IJRA), 2017, Vol. 6, No. 2, 101-113

BIOGRAPHIES



Mr. Pabitra Kumar Nandi is currently a PhD Research Scholar in the Department of Production Engineering, Jadavpur University, India. He received his B. E. & M. E. degrees in Electronics & Communication Engg from MAKAUT in 2008 & 2010 respectively. His Field of Specialization and Research Area are Mobile Robotics with AI, IoT based Embedded Systems, Microprocessor & Microcontroller, and Computer Architecture. He has teaching & research experience of 14 years



Dr. Ajoy Kumar Dutta is currently a Professor in the Department of Production Engineering, Jadavpur University, India. He received his B. E. & M. E. degrees in Electronics & Tele-communication Engg from Jadavpur University in 1983 & 1985 respectively, and Ph. D. (Engg) degree in the area of Robotics from Jadavpur University in 1991. His Field of Specialization and Research Area are Robotics, Sensors, Computer Vision, Microprocessor Applications, and Mechatronics. He has teaching & research experience of 39 years.