Defence Science Journal, Vol. 67, No. 5, September 2017, pp. 572-580, DOI : 10.14429/dsj.67.10237

Modelling, Simulation and Mechatronics Design of a Wireless Automatic Fire Fighting Surveillance Robot

B. Madhevan*, R. Sakkaravarthi, G. Mandeep Singh, R. Diya, and Durgesh Kumar Jha

Vellore Institute of Technology University, Chennai - 600 127, India
**E-mail: bmadhevan@iiitdm.ac.in*

ABSTRACT

The aim of this study is to design and develop an autonomous fire proof rescue robot. The robot is designed in such a way, that it can traverse through fire and hazardous situations. Further, it will sense and communicate information regarding these situations in real time with the server. The robot is fixed with multi-sensors and further, a driver circuit has been integrated for communication in these hazardous situations through Zigbee and a data acquisition system (DAQ). In mechanical design first, a 3D solid model is generated using Solid works software to understand the basic structure of robot which provides information regarding robotic platform, size and location of various components. The developed fire fighting robot is a predominately outdoor ground-based mobile robotic system with onboard subdual systems that can traverse autonomously in the hazardous environment. The robot is designed such that it can traverse into the fire and send information regarding the fire behaviour and also the images of the victim's location by using a camera. Further, a mathematical model which describes the kinematics and dynamic behaviour of robot motion are done. V-REP is used to create the simulation of the robot in a fire simulated fire environment. Finally, for the path planning, various techniques are discussed such as V-REPs inbuilt path planning module, A*, Fuzzy logic and artificial potential fields.

Keywords: Fire detecting; Mobile robot; Search and rescue security; System architecture; Virtual reality simulation; Obstacle avoidance; Robot navigation; Robot motion planning

1. INTRODUCTION

The scope of the current work is to design and develop a fireproof fire fighting surveillance robot that can traverse into fire environment and send information about the fire behaviour and the location of people trapped in these hazardous areas. When a fire breaks out in an industrial environment or a house fire environment, it takes approximately 30 min to 40 min for the fire rescue team to reach the spot. Thus, the developed robot would help the fire rescue team to better understand the fire behaviour and trapped person in these hazardous areas. The proposed model is an amateur attempt at creating an autonomous fire proof rescue machine, aiding humans in fire and hazardous situations. When the fire breaks out in a building or any closed spaces, it is quite risky for humans to traverse through this hazardous situation, as one may get trapped in such closed spaces.

For the year 2009-2013, average fire accidents in the state of Tamilnadu, India has been accounted as 23000 resulting in property loss of about 30 crores (average). In such cases, an autonomous fire proof robot, designed on-board, can be very efficiently used for fire fighting. This robot can move around the place of fire captures the conditions around and give the exact situation of the place on fire. The robot accommodates a camera protected with fire proof glass on itself to send the rescuers the live information via video streaming to take proper action. An autonomous robot used in land navigation to detect the obstacle coming in its path was discussed[1,2]. Tok[3], developed semi-autonomous mobile robots for identifying the victims trapped in hazardous and fire bounded areas. The main function of such robots was to provide information about the path traversed by the robot. Further, kinematics, heat transfer calculation, and dynamic analysis were also done. Ruangpayoongrak[4], *et al.* introduced a new algorithm that can calculate the distance of an obstacle from the sensor board based on (N*N) matrix and the transformation used was distance transform. Further, the constraint such as localisation, environmental conditions was also obtained from the algorithm. Survey on disaster prone areas was conducted extensively[5,6].

Further, these research works proposed a wireless sensor network to determine the human by an automatic system with the help of sensors and camera. Kinematic and dynamic analysis of skid steering wheeled mobile robot is presented[7] and based on these equations, software analysis was performed. The model showed a significant improves in the dead reckoning performance of the robot. A model for performance time estimation for mobile robot motion based on the Robot Time and Motion (RTM) method is proposed[8]. Thatcher[9], *et al.* developed operating and programming systems for modular robots with user defined protocols. A* algorithm to find the

Received : 04 July 2016, Revised : 03 June 2017
Accepted : 14 June 2017, Online published : 19 September 2017

optimum path for a mobile robot from a start location to goal location is demonstrated[10]. In this algorithm, the robot was assumed to be a point and obstacles were inserted to make robot find the best path towards goal. Matlab software was used to test the algorithm, which showed the proper implementation of the algorithm with different maps. The potential field algorithm is presented[11], in which an artificial field was developed. Further, the obstacles induce a repulsive force against the robot and goal develops an attractive field towards the robot in the developed artificial field. The obstacles induce a repulsive force which was inversely proportional to the distances. The resultant potential was used to calculate the next move of the robot. Fuzzy logic method, which was a reactive planning technique just as the potential field is introduced[12,13]. The term reactive means that the next step of the robot was calculated using the immediate positions and the distances, without considering the future consequences of the moves. Rama Krishna & Sowmya Bala[14], described fire fighting robot equipped with fire protection sensor to detect the human in the fire environment. The data is transferred through a wireless network from the victim to the rescue team.

Further Survey on the application of visual servoing in robot control and automatic control was elaborated[15]. General classification of the various servoing systems was explained, but the main focus was on the manipulator's arm with a single camera with installed end effectors. Trajectory Tracking and path planning algorithms for mobile robots were discussed[16,17]. System architectures for mobile robots were explicitly addressed[18-22].

The developed fire fighting robot is a predominately outdoor ground-based mobile robotic system with on-board subdual systems that can traverse autonomously in the hazardous environment. Subdual systems mounted onto the robots include anti-fire monitors and a water fog system. The new system architecture is developed and based on this, a wireless connection to transmit information from sensors on-board the robot to the operator for assisting in navigation and fire suppression. Sensors on the robots include a thermal sensor, vision camera, IR cameras and range finders to assist in avoiding obstacles.

2. ROBOTICS MODULES

2.1 Choosing a Robotics platform

There are various robotic platforms available and each has its own advantages and disadvantages. Further, a tracked robot is preferred to wheeled robots because of its distinct advantages. The main advantages of tracked robots are high performance and optimised traction system.

2.2 Material for body and Insulation

The robot has to move inside a fire environment consisting of very high temperatures, ranging from 1000 to 2000 degree Celsius. In order to make the robot approximately fire proof, it has to be shielded using a proper material as shown in Tables 1-3.

Table 1. Material properties.

Material	Density Kg/m^3	Thermal conductivity (W/m.K)
Fiber glass <250 °C	14-100	0.04 at 20 °C
Rock wool <700 °C	30-200	0.04 at 20 °C
Ceramic fiber paper	200	0.08 at 600 °C 0.10 at 800 °C 0.18 at 1000 °C

Table 2. Material for body I.

Material	Tensile strength	Compressive strength (MPa)
Aluminum	70-700	NA
Carbon steel	500	500
Iron	170	550
Copper	70-220	117-220

Table 3. Material for body II.

Material	Melting conductivity 25 °C (W/m.K)	Density point °C
Aluminum	660.3	2.7
Carbon steel	1425-1540	8.05
Iron	75	7.9
Copper	1085	8.9

3. SYSTEM DESIGN

The System architecture includes various subsystems and function of each subsystem is briefly described as follows:

- Computer system: Controls all subsystems.
- Power control system: Provides power to the system in a uniform manner.
- Navigation system: Provides a plan to traverse in any environment with obstacle and collision detection.
- Communication system: The on-board PC communicates with the robot's micro controller through serial ports.
- Obstacle avoidance system: This system consists of sensors to detect the obstacle, and to make the information available to the computer system.
- Environment detects system: This system fetches information about the environment through GPS.

4. MATHEMATICAL MODELLING

4.1 Coordinate Identification using Trilateration

From the figure illustrated in Fig. 1.

$$(r_1)^2 = (x)^2 + (y)^2 \tag{1}$$

$$(r_2)^2 = (x-d)^2 + (y)^2 \tag{2}$$

$$(r_3)^2 = (x-a)^2 + (y-b)^2 \tag{3}$$

Now condition is,

$$(d - r_1) \ll r_2 < (d + r_1) \tag{4}$$

Let us assume the values of d, r_1, r_2 as 5, 2 and 4, respectively.

- Each robot consists of three sensors and one radio frequency (RF) transceiver.
- Master robot emits RF pulse and sensor output which are received by the receiver unit of follower robot at a

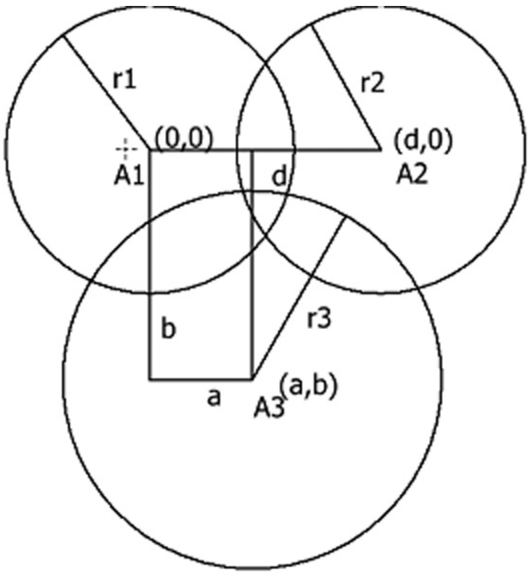


Figure 1. Trilateration technique.

particular time period.

- Three sensors are placed at positions *A, B,* and *C*.
- Let *A* be the origin (0,0).
- Fix *C* and *B* at a distance *d* units from *A* in both X-axis and Y-axis respectively. Therefore coordinates of *B* is (*d*, 0) and coordinates of *C* is (0, *d*).
- Using Trilateration technique,

$$(x_{2A} - x_{1C})^2 + (y_{2A} - y_{1C})^2 = c^2 \tag{5}$$

$$(x_{2A} - x_{1A})^2 + (y_{2A} - y_{1A})^2 = a^2 \tag{6}$$

$$(x_{2A} - x_{1B})^2 + (y_{2A} - y_{1B})^2 = b^2 \tag{7}$$

Coordinates of *A* is (0, 0) , *B* is (*d*, 0) and coordinates of *i* is (0, *d*). The new coordinate (x_{2A}; y_{2A})

$$\frac{(a^2 + d^2 - b^2)}{2d} = x_{2A} \tag{8}$$

$$\frac{(a^2 + d^2 - c^2)}{2d} = y_{2A} \tag{9}$$

and similarly for 3D, (x_{2A}; y_{2A}; z_{2A}) is given as:

$$\left(\frac{(a^2 + d^2 - b^2)}{2}, \frac{(a^2 + d^2 - c^2)}{2}, \frac{(a^2 + d^2 - e^2)}{2}\right) \tag{10}$$

4.2 Trilateration Technique in Matrix Form

$$(x_1 - x_a)^2 + (y_1 - y_a)^2 = r_1^2 \tag{11}$$

$$(x_2 - x_a)^2 + (y_2 - y_a)^2 = r_2^2 \tag{12}$$

$$(x_3 - x_a)^2 + (y_3 - y_a)^2 = r_3^2 \tag{13}$$

Subtracting Eqn (11) : Eqn (13) and Eqn (12) : Eqn (13) and rearranging,

$$\begin{bmatrix} (x_3 - x_1) & (y_3 - y_2) \\ (x_3 - x_1) & (y_3 - y_2) \end{bmatrix} \begin{bmatrix} x_a \\ y_a \end{bmatrix} = \begin{bmatrix} (r_3^2 - r_1^2) + (d_{13}^2) \\ (r_3^2 - r_2^2) + (d_{23}^2) \end{bmatrix} \tag{14}$$

Further, equation can be rearranged as,

$$\begin{bmatrix} x_a \\ y_a \\ z_a \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \left(A^T A\right)^{-1} \left(A^T B\right) \tag{15}$$

4.3 Sample Calculation for Obstacle Avoidance

Let (x_1;y_1) represent the position of the obstacle and (x_0;y_0) represent the position of the robot, ρ be the distance between the robot and the obstacle. To consider a point as an obstacle, it should satisfy both radial range and orientation angle criteria. The robot consists of three sensors in which two are mounted straight, and the third one is mounted in an inclined position.

(A) The boundary conditions for obstacle identification sensing module is given as:

1. $\rho < r$
2. $(\theta_1 - \theta_2) < \varphi < (\theta_1 - \theta_3)$ - Right
3. $(\theta_1 - \theta_3) < \varphi < (\theta_1 + \theta_3)$ - Front
4. $(\theta_1 + \theta_3) < \varphi < (\theta_1 + \theta_2)$ - Left

(B) Sample calculation, when, $(\theta_1, \theta_2, \theta_3)$ =(47, 23, 11.5) and if, φ is 65, then,

$$24 < 65 < 35 = invalid \tag{16}$$

$$35.5 < 65 < 58 = invalid \tag{17}$$

$$58.5 < 65 < 70 = left \tag{18}$$

If, φ is 50, then,

$$24 < 50 < 35 = invalid \tag{19}$$

$$35.5 < 50 < 58 = front \tag{20}$$

$$58.5 < 50 < 70 = invalid \tag{21}$$

If, φ is 30, then,

$$24 < 30 < 35 = right \tag{22}$$

$$35.5 < 30 < 58 = invalid \tag{23}$$

$$58.5 < 30 < 70 = invalid \tag{24}$$

Hence, when $(\theta_1, \theta_2, \theta_3)$ =(47, 23, 11.5) and if, φ is 65, $(\theta_1 + \theta_3) < \varphi < (\theta_1 + \theta_2)$ - Left

if, ϕ is 50, $(\theta_1 - \theta_3) < \varphi < (\theta_1 + \theta_3)$ - Front

if, ϕ is 30, $(\theta_1 - \theta_2) < \varphi < (\theta_1 - \theta_3)$ - Right

4.3.1 Kinematical and Dynamic Model

The robot's mass centre is at the body frame's geometric centre. The two adjacent wheels rotate with equal speeds. All the four wheels keep constant ground contact. Kinematical and dynamic model are as shown in Figs. 2(a) and 2(b), respectively.

- Relationship equations for robot velocities in both the frames:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ w_z \end{bmatrix} \tag{25}$$

[*X,Y*] are the coordinates globally and [*x,y*] are the robot's

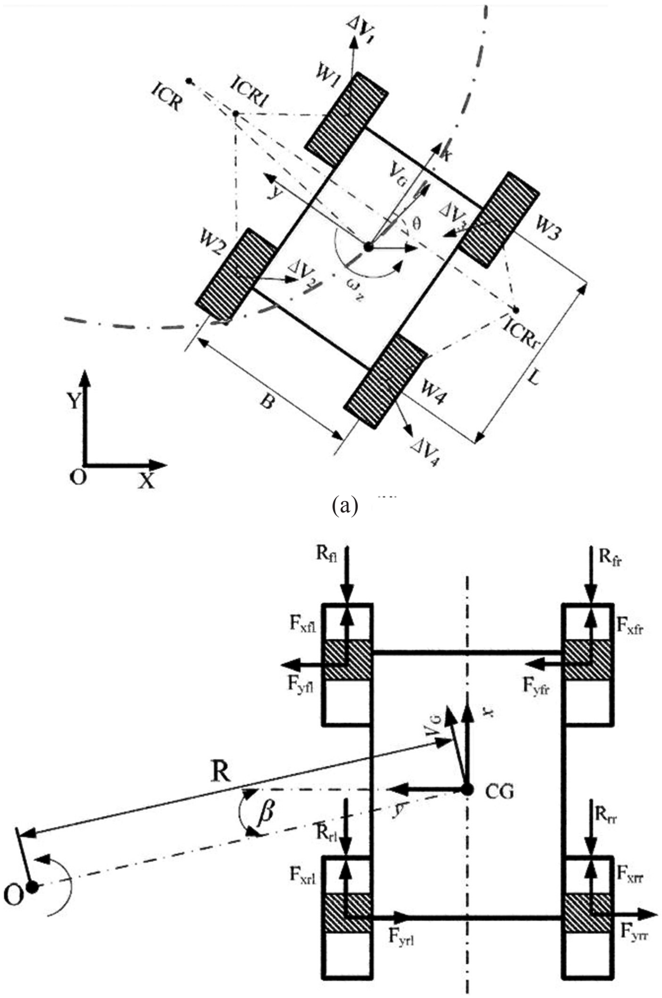


Figure 2. (a) Kinematics diagram for proposed robot and (b) Forces and moments on vehicle.

coordinates locally. v_x;v_y are linear velocities in body frame (local frame) and w_z is the angular velocity.

- Relationship equation for control inputs and speed of wheels:

$$v_x = \frac{\left[(\omega_l * r) + (\omega_r * r)\right]}{2} = \frac{v_l + v_r}{2} \tag{26}$$

$$\omega_z = \frac{\left[(-)(\omega_l * r) + (\omega_r * r)\right]}{2 * y_0} \tag{27}$$

l-left, r-right and y_0 is the instantaneous tread instantaneous centres of rotation value.

- Instantaneous radius of the path curvature:

$$R = \frac{v_G}{\omega_z} = \frac{v_l + v_r}{v_r - v_l} * y_0 \tag{28}$$

Hence, Straight line motion, $R = \infty$ and Rotational motion, $R = 0$.

- Dynamic model equations[2]:

$$F_{xfr} + F_{xrr} + F_{xfl} + F_{xrl} = R_x + \frac{m*(v_g)^2}{R} * \sin\beta \tag{29}$$

$$F_{yfr} + F_{yrr} + F_{yfl} + F_{yrl} = \frac{m*(v_g)^2}{R} * \cos\beta \tag{30}$$

$$M_d - M_r = 0 \tag{31}$$

where, F_{xfr}; F_{xrr}; F_{xfl}; F_{xrl} are the forces in longitudinal direction. F_{yfr}; F_{yrr}; F_{yfl}; F_{yrl} are the forces in lateral direction. v_g is the velocity of the vehicle, β is the angle between the velocity of the vehicle and x-axis of the local frame of reference. Rx is motion resistances on the four wheels acting externally. M_d and M_r are the drive and resistance moments, respectively.

- Ground-wheel interaction, shear stress τ_{ss} and shear displacement j can have relationship as written below :

$$\tau_{ss} = p * \mu * \left(1 - e^{\frac{-j}{K}}\right) \tag{32}$$

where p is the pressure applied normally, μ is the friction coefficient and K is the deformation shear modulus.

5. VIRTUAL REALITY SIMULATION

5.1 V-REP Path Planning Module

A virtual environment has been created using VREP software for simulating the behaviour of mobile robots and their environment. The goal position is the human sitting on a chair. Figure 3 shows the snapshots of the V-REP simulation.

After generating the environment, the simulation is made to run and the robot is made to traverse through the fire to reach the goal position. The direction of the robot is controlled by changing the left and right velocities of the caterpillar tracks. In the running simulation, the fire and smoke are also seen which are not visible when the simulation is stopped.

5.2 Path Planning Methods

5.2.1 V-REP Path Planning Module

The V-rep software has inbuilt path planning module, through which an optimum path for the starting and destination point is achieved. An example of V-REP's path planning module is as shown in Fig. 4.

After identifying the robot and a number of obstacles in the environment, calculation function module *f*(*x*) available on the left side is selected. Further, it has various calculation modules for collision detection, distance calculation, path planning, motion planning, and dynamics. Before going into the path planning module, three things are necessary, first, a start dummy, followed by an end dummy and finally a path. Start dummy is at the location of the robot and end dummy is at the end location where the robot has to be at the end. The path is the connecting curve between these two points. In the path planning module, various parameters such as an object, goal dummy, and path object are selected, which describe the path to be traversed. After completing the parameters and making appropriate parenting between start dummy, robot, path, and the optimum path are calculated by clicking on computing path.

5.2.2 A Algorithm*

This is another method of using a mathematical A* star algorithm for generating an optimum path from start to end location. The snapshots of A* star algorithm: sequence 1 and

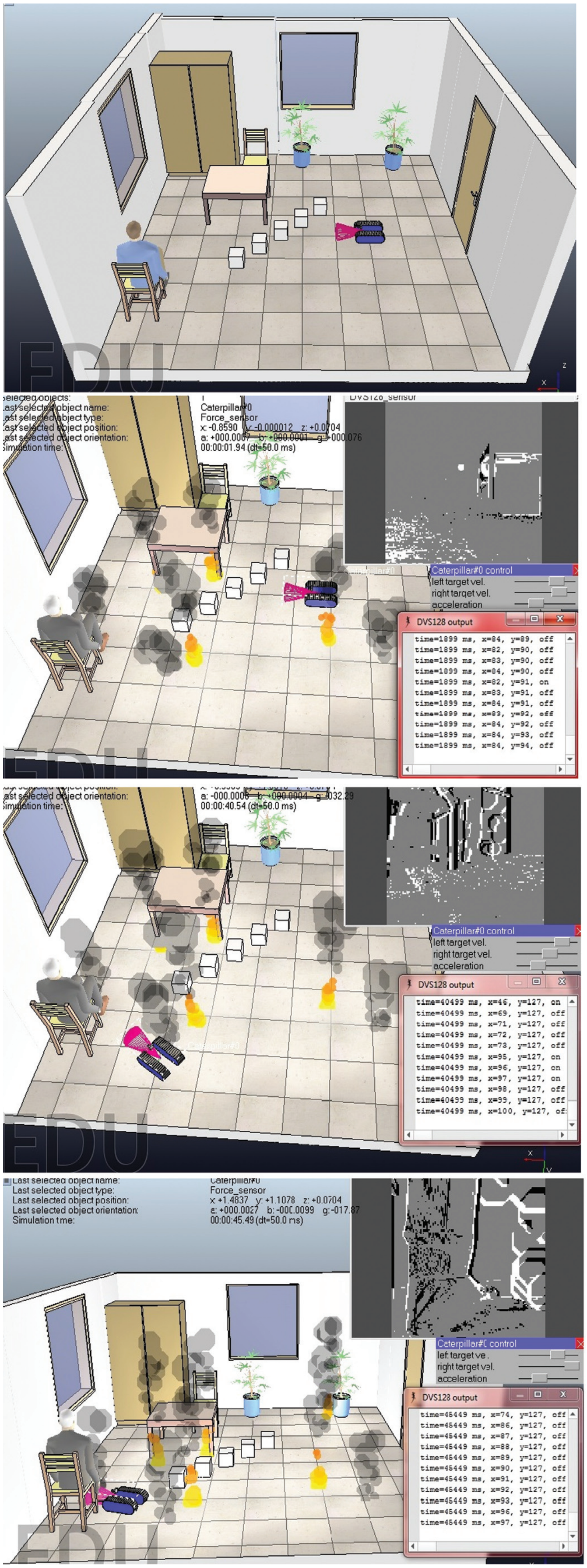


Figure 3. The snapshots of the V-REP simulation.

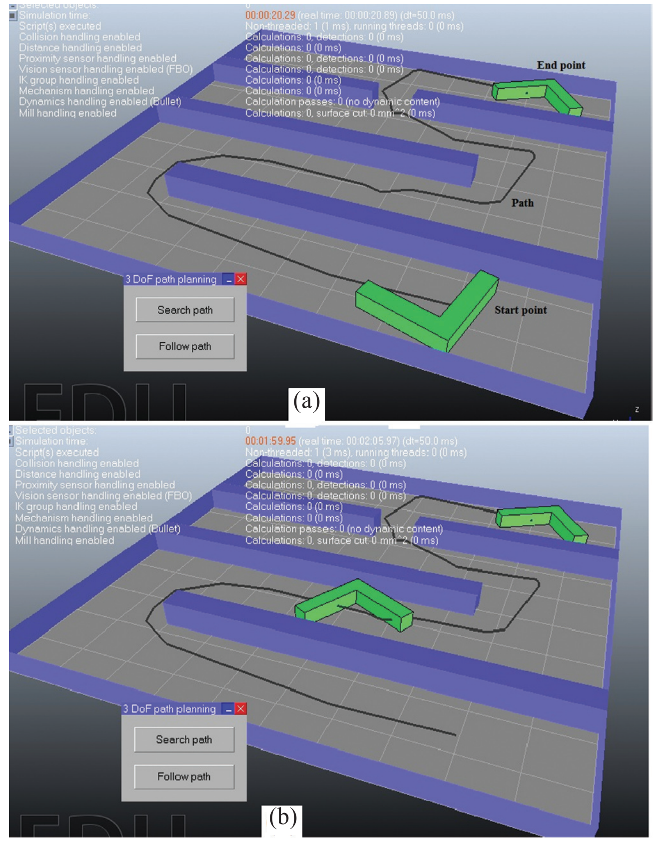


Figure 4. The snapshots of path searching algorithms: (a) Search for a path and (b) Following the searched path.

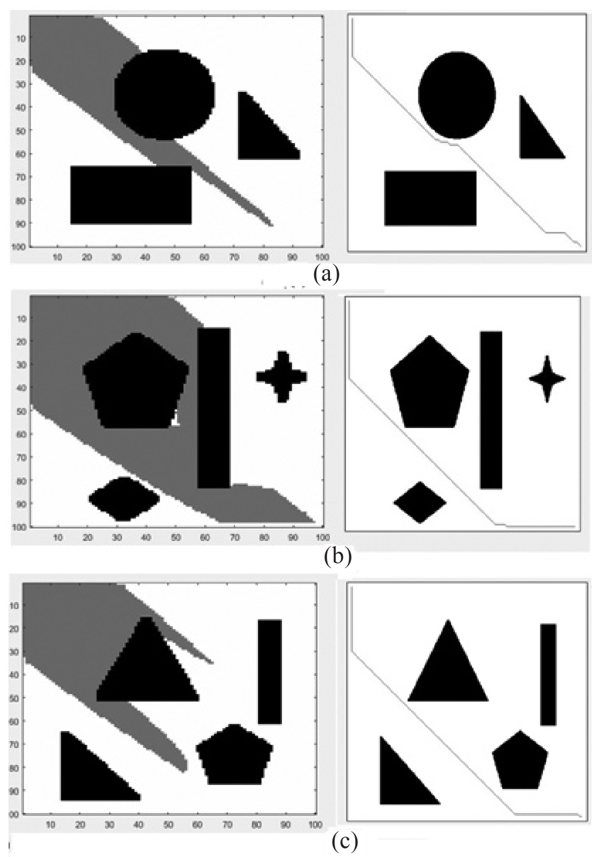


Figure 5. Snapshots of A* star algorithm: (a) Map 1 in sequence 1, (b) Map 2 in sequence 1; (c) Map 1 in sequence 2.

sequence 2 are as shown in Fig. 5. For an application of this algorithm, a map of the environment is needed, which can be produced using a camera or any other map building sensors within the robot or in a simpler way a centralised camera which captures the picture of the environment. The robot is also captured in the picture and represents the starting point and end location as the end point.

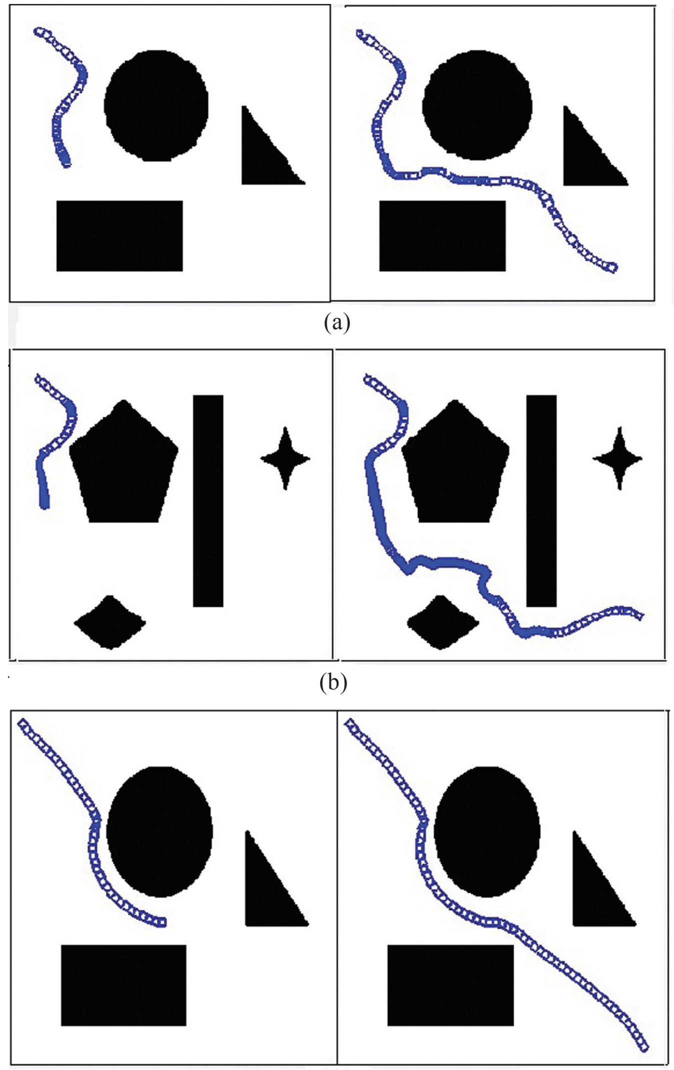
Using the Matlab code for A* star algorithm[3], following results are generated using two maps. The robot is located at [10,10] and the end point is at [100,100]. The black shapes are the obstacles which the robot has to avoid in reaching the goal point. The A* algorithm uses the input graph and searches each all regions one by one and try to find the shortest path between the source point and all other regions. This algorithm explores the closest regions first but also has a bias to regions close to the goal. This bias is explained by the heuristic function. The path produced in map 1 has a processing time =875 s, and path length = 726. The map 2 has processing time = 665 s and path length = 775 as shown in Fig. 5.

5.2.3 *Artificial Potential Fields and Fuzzy Logic Rule Base*

This is another method of reactive path planning. In this method, the immediate distance between the obstacles is evaluated and next move of the robot is decided, without considering the future consequences of the moves. This motion of the robot leads to it reaching the goal. The obstacles have a repulsive force over the robot and goal has an attractive force. The resultant force evaluated by combining both the forces is used to plan the move the robot towards goal. Using these techniques to generate maps, this algorithm has been deployed to find the path from source to goal. The map 1 has processing time =13 s and path length = 774. The map 2 has processing time = 39 s and path length = 801. The application of this algorithm is shown in Figs. 6(a) and 6(b). This is the last method explored in this article. Just like the potential field algorithm, this is also a reactive planning algorithm. The term reactive means that the immediate distance and position of obstacles are used to decide the next move of the robot, without concerning about the future consequences of the move. To apply this method, an appropriate input selection has been selected first, which best describes the current situation of the robot.

Further, the FL developed demonstrates the schemes used for traversing the robot along a planned path when an obstacle is encountered. These rules monitor the distance (D) and the angle (A). The abbreviations used are NB, NM, NS, PS, PM, PB. The first letter is an {N or P} which means negative or positive. The second letter is {B, M or S} for big, medium or small, respectively. These rules were implemented by traversing the robot in different positions by varying the values of A and D. If the robot is far away from the path and deviating from the set point, then swift action is required as developed by FL rules by quickly changing the direction of steering towards the path and slowing down the speed. The application of this algorithm is shown in Fig. 6(c).

The next motion move is based on these inputs and not considering the actual scenario. Six inputs are selected for this problem. These are obstacle distance front, front left diagonal obstacle distance, front right diagonal distance, the heading direction, and goal angle, robot distance and angle of preferred turn. Using the Matlab code by Harrabi & Braiek[5], a fuzzy logic algorithm is implemented. The map 1 has processing time = 8:6 s and path length = 902.



(a)

(b)

(c)

Figure 6. Snapshots of: (a) Map 1 in potential field algorithm, (b) Following the searched path in potential field algorithm, and (c) Map 1 in fuzzy logic algorithm.

6. RESULTS AND DISCUSSION

From Fig. 7, it can be noted that the robot starts traversing in a hazardous environment and later after a specified time, the contact angles for three different scenarios are shown in Fig. 7(a). The slip and dynamic pull force increase as the robot traverse in this environment as seen in Fig. 7(d) and Fig. 7(b) respectively. The output accelerations of the controller are depicted in Fig. 7(c). It is noted that, as the robot moves into the environment, moretorque is required to localize the robot from excess heat. The torque required for driving and corresponding rpm profiles are plotted in Fig. 7(e) and Fig. 7(f), respectively. After the initial phase, the commanded acceleration varies approximately with a fixed period to adapt to the motion of

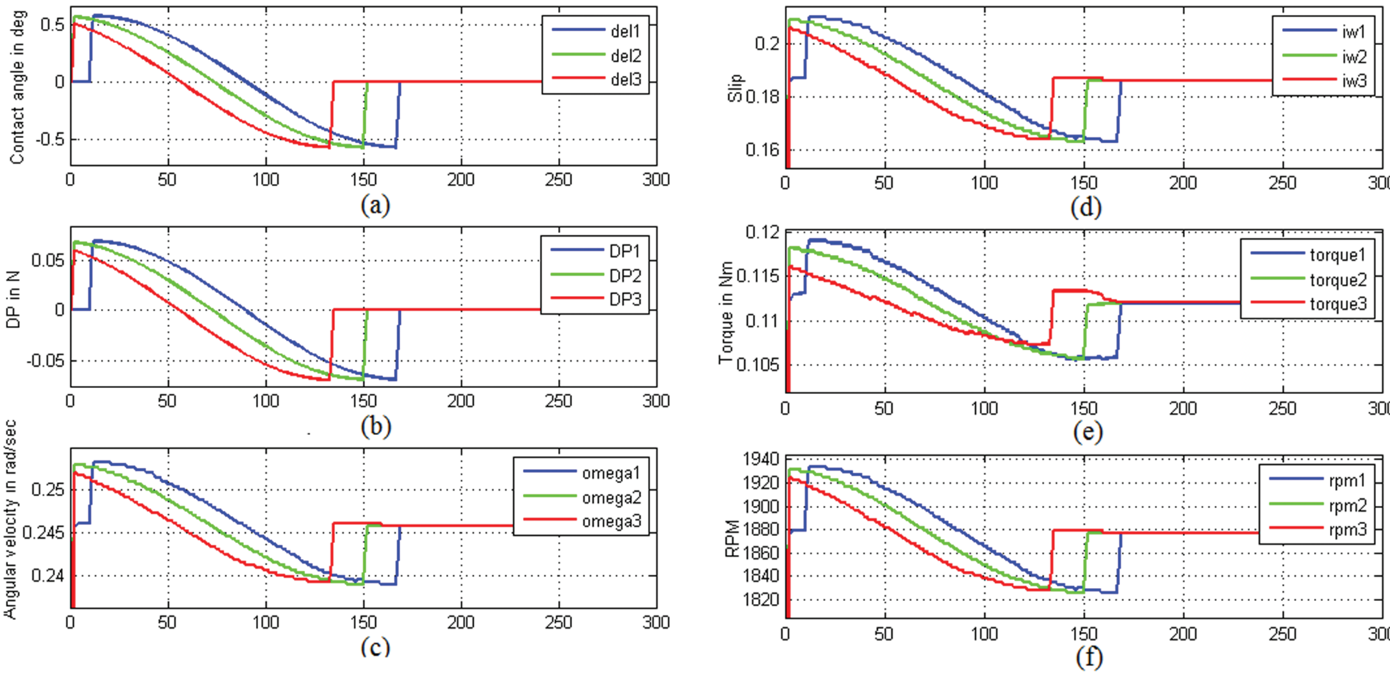


Figure 7. Estimation of angle, pull force, angular velocity, slip, torque and RPM.

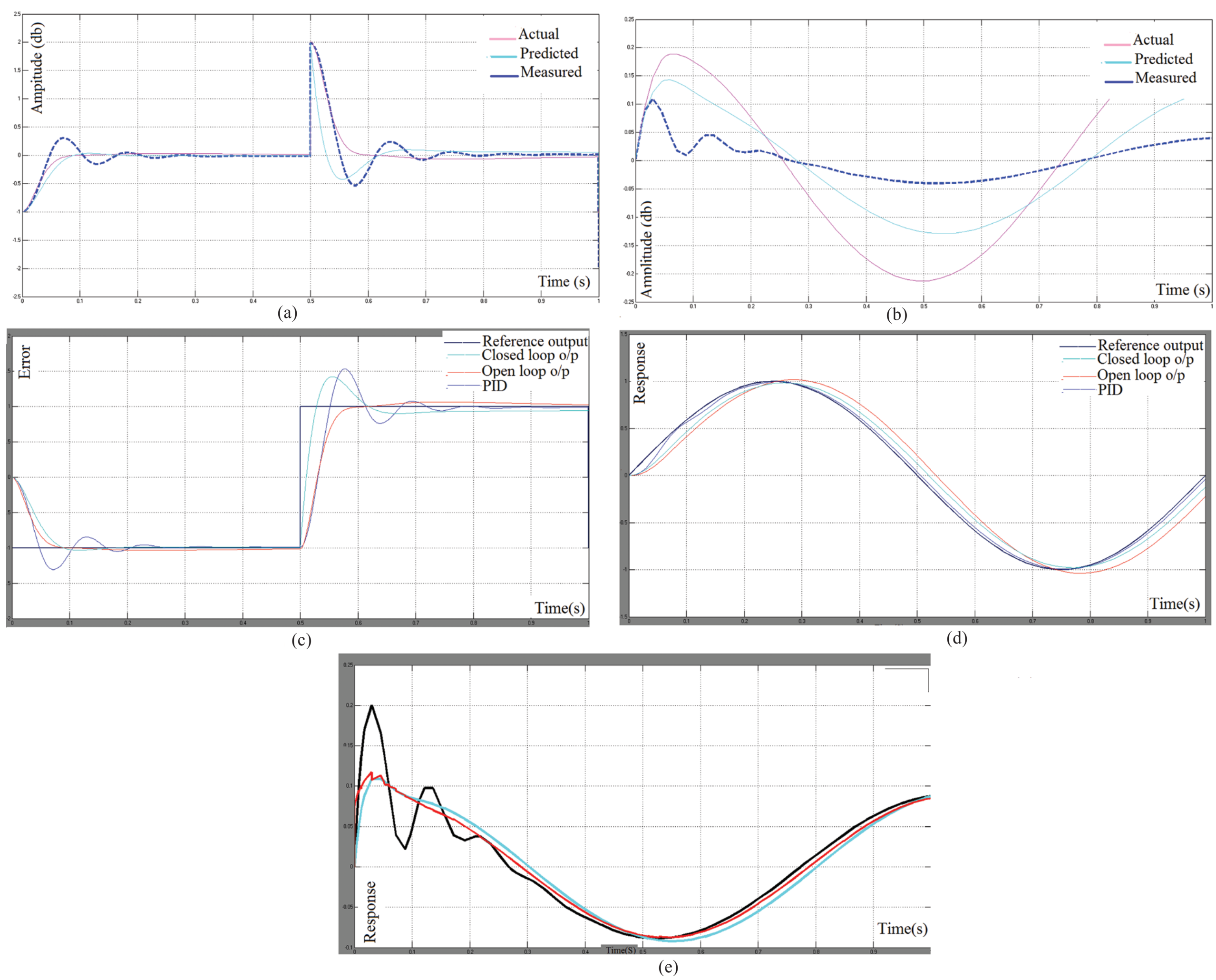


Figure 8. (a) Trajectory, (b) The relative distance between robot and victim, (c) The relative orientation between robot and victim, (d) Formation error for the robot, and (e) Response error.

the robot. The trajectory of the robot as recorded by the vision system is depicted in Fig. 8. The corresponding formation states are as shown in Fig. 8(a) and 8(b). As a result, the perturbation from the internal dynamics becomes larger. As depicted in Fig. 8(a) and 8(b), in this stage, the robot remains bounded and are influenced little by the varying of internal state, showing that the formation system is robust to the perturbation from the internal dynamics. Formation and response error are shown in Fig. 8(c) and 8(d), respectively. Both of the tracking errors are limited when the formation is in the steady state. When the trajectory of the robot is not smooth, the rotation motion of the robot varies with high frequency.

7. CONCLUSIONS

The current research work presents an interesting topic related to the social application based on fire fighting robot. Further, a scheme for identifying different system architectures for fire fighting robot has been presented. This scheme combines a PIR sensor, system architecture, positioning of robots with obstacle avoidance. Based on this, a mathematical model based on trilateration has been derived, which provides the precise position of the robots. In order to enhance the efficient performance of the robot, a sample model of obstacle avoidance algorithm is also proposed. This article gives an idea of what could be the best option for materials for the body and insulation of the robot and a 3D model is created using Solid works software. A virtual reality simulation of the tracked robot using V-REP software is done, which explains the behaviour of the robot in the fire environment. A detailed simulation study for fire fighting, using v-rep incorporating all constraints, applicable to an unknown environment has also been presented. Finally, various path planning method has been presented discussed and how they are implemented to find the optimum path is presented. Each algorithm is implemented using the maps and their processing time and path length are calculated. Simulation results from v-rep that shows the robustness of the algorithm.

REFERENCES

1. Hanumante, V.; Roy, S. & Maity, S. Low cost obstacle robot. *Int. J. Soft Comput. Eng.,* 2013, **03**, 52-55. doi: 10.6084/M9.FIGSHARE.1327873
2. Gasilov, N.; Dogan, M. & Arici, V. Two-stage shortest path algorithm for solving optimal obstacle avoidance problem. *IETE J. Res.,* 2011, **57**(3), 278-285. doi: 10.4103/0377-2063.83650
3. Tok, O. Design and manufacturing of a fire proof fire rescue robot. Izmir Institute of Technology, Turkey, 1994, (Master Thesis).
4. Ruangpayoongrak, N.; Roth, H. & Chudoba, J. Mobile robots for search and rescue. *In* International Workshop on Safety, Security and Rescue Robots, Kobe, Japan, 2005.
5. Harrabi, R. & Braiek, E.B. Color image segmentation using a modified fuzzy C-Means technique and different color spaces. *In* International Conference on Advanced Technologies for Signal and Image Processing, Sousse, Tunisia, 2014. doi: 10.1109/ATSIP.2014.6834612
6. Sunil, M.P. Design of an event based surveillance system using advanced technology. *Int. J. Eng. Sci. Innovative Technol.,* 2013, **2**(6), 451-459.
7. Wang, T.; Yao, Wu.; Jianhong Liang; Chenhao Han; Jiao Chen & Qiteng Zhao. Analysis and experimental kinematics of a skid- steering wheeled robot based on a laser scanner sensor. *Sensors.,* 2015, **15**(5),9681-9702. doi: 10.3390/s150509681
8. Rembold, B. & Shimon, Y.N. Modelling the performance of a mobile robot with RTM. *Int. J. Production Res.,* 2007, **29**(5),967-978. doi: 10.1080/00207549108930113
9. Thatcher, T.W.; Weston, R.H.; Moore, P.R. & Gascoigne, J.D. Software design for modular robots. *International J. Production Res.,* 2010, **22**(1), 71-84. doi: 10.1080/00207548408942431
10. Kala, R. Code for robot path planning using a* algorithm. Indian Institute of Information Technology Allahabad. http://rkala.in/codes.php#codes
11. Kala,R.Codeforrobotpathplanningusingartificialpotential fields. Indian Institute of InformationTechnology Allahabad,http://rkala.in/codes.php#codes
12. Kala, R. Code for robot path planning using fuzzy logic. Indian Institute of Information Technology Allahabad. http://rkala.in/codes.php#codes
13. Chandwani, M. & Chaudhari, N.S. Control algorithms for optimal derivation strategy in fuzzy rule-based systems. *IETE Journal of Research.,* 1993, **39**(4), 217-223. doi: 10.1080/03772063.1993.11437122
14. Rama Krishna, A. & Sowmya Bala, G. Design of a rescue robot assist at fire disaster, *Int. J. Comput. Appl.,* 2012, **47**(10),25-30. doi: 10.5120/7225-0056
15. Malis, E. Survey of vision-based robot control. Technical Report, INRIA, Sophia Antipolis, France, France, 2002. doi: 10.1.1.98.3861
16. Amoozgar; Hadi, Sadati,; Seyed Hossein & Alipour Khalil. Trajectory tracking of wheeled mobile robots using a kinematical fuzzy controller, *Int. J. Robotics Automation*, 2012, **27**(1). doi: 10.2316/Journal.206.2012.1.206-3476
17. Zhang, H.Z.; Xiong, S.Y. & Yue Liu. Real-time path planning of multiple mobile robots in a dynamic certain environment. *Int. J. Robotics Automation.,* 2013, **28**(1). doi: 10.1631/jzus.2006
18. Madhevan, B. & Sreekumar, M. Tracking algorithm using leader follower approach for multirobots, *Procedia Engineering,* 2013, **64**(3),1426-1435. doi: 10.1016/j.proeng.2013.09.224
19. Madhevan, B. & Sreekumar, M. A systematic implementation of role assignment in multi robots using leader follower approach: analytical and experimental evaluation. *In* 13th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, December,2014. doi: 10.1109/ICARCV.2014.7064588.
20. Madhevan, B. & Sreekumar, M. Implementation of

role assignment and fault tree analysis for multi robot interaction. *Int. J. Robotics Automation.,* 2015, **32**(3). doi: 10.2316/Journal.206.2017.3.206-4448

21. Madhevan, B. & Sreekumar, M. A Systematic approach for role assignment in multi robots using leader follower approach: experimental observation. *Int. J. Robotics Automation,* 2015. **32**(3). doi: 10.2316/Journal.206.2017.3.206-4578
22. Madhevan, b.; Sakkaravarthi, R. & Durgesh, Kumar. A Novel image intelligent system architecture for fire proof robot. *Arti. Intell. Evolutionary Comput. Eng. Sys.*, 2017, **517**, 805-817.

CONTRIBUTORS

Dr Madhevan B. did his PhD in mobile robot from Indian Institute of Information Technology Design and Manufacturing (IIITDM) Kancheepuram, India, where he worked on implementation of role assignment for multi robot. Presently working as Robotics Scientist at Digirobotics, London. Currently he focuses on robotics and automation technologies for humanitarian social needs.
In the current study, he designed of mobile robots with mathematical modelling.

Dr Sakkaravarthi R. is currently serving as an Associate Professor, School of Computing Science and Engineering, VIT University, Chennai. Currently he focuses on dependable computing, autonomic computing, fault to lerant computing, adaptive middle ware architecture, triple play resilience.
In the current study, he did architecture designed with artificial intelligence.

Mr Mandeep Gill is Mr Mandeep Gill did his master's in the stream of Mechatronics, VIT University, Chennai. His area of interest includes control system and intelligence system.
In the current study, he developed path planning modules and algorithms.

Ms Diya R. did her master's in the stream of Mechatronics, VIT University, Chennai. Her area of interest includes robotics and automation.
In the current study, she developed kinematics and dynamics model.

Mr Durgesh Jha did his master's in the stream of Mechatronics, VIT University, Chennai. His area of interest includes image processing and automation.
In the current study, he developed path planning modules in v-rep.