Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Procedia Technology 21 (2015) 232 – 239
Procedia Technology

SMART GRID Technologies, August 6-8, 2015

Assistive Autonomous Ground Vehicles in Smart Grid

Kuralamudhan Arutselvan[a*], Vijayakumari A[b]

[a]KTH Royal Institue of Technology, Stockholm 100 44, Sweden
[b]Amrita Vishwa Vidyapeetham, Coimbatore 641112, India

Abstract

Autonomous Ground Vehicles (AGVs) are nominated for a wide range of applications in smart cities. This paper studies and simulates the potential field algorithm used for path planning of such AGVs to extend there application for automation and control of smart grids. Different sensors are studied and implemented in the algorithm to help the AGV navigate from an initial position to a goal by avoiding any obstacles in its path. A solution is analyzed and simulated for the problem of AGVs getting trapped in local minima. A new application for the AGVs as assistance in smart grid is also discussed. A real implementation of the project has also been done at the KTH Smart Mobility Lab, Sweden using the Nexus robots and the algorithm was implemented successfully.

Peer-review under responsibility of Amrita School of Engineering, Amrita Vishwa Vidyapeetham University

Keywords:Autonomous Ground Vehicles(AGVs); Local Minima; Potential Field Algorithm; Virtual Force; Motion Capturing

1. Introduction

The key to the progress of our world is automation. Autonomous Ground Vehicles (AGVs) are one such key feature for the progress of today's smart cities. The first AGV was designed in 1954 by A.M.Barrett Junr [1]. It was used for pulling a trailer in a grocery ware house. Since then these autonomous vehicles have been used to carry out different tasks which reduce human effort such as floor cleaning, surveillance, military, space explorations and many more. This project designs and simulates an AGV for applications in smart grids. One major application for AGVs is surveillance. The AGVs can be programmed to monitor different locations of the grid by equipping them with cameras, microphones and other sensors. Major benefits of using AGVs is that, they can be moved to different locations in the grid, can be remotely controlled to carry out measures in locations inaccessible to the control

* Corresponding author. Tel.:919443867001
E-mail address kuralamudhan1@gmail.com

Peer-review under responsibility of Amrita School of Engineering, Amrita Vishwa Vidyapeetham University
doi:10.1016/j.protcy.2015.10.093

engineers and it is cost effective when compared to installing cameras and other sensors at every location in the grid. Equipped with micro phone, cameras, speakers, smoke detectors and thermometers the AGV can act as a multipurpose monitoring system. The information collected by the AGV can be sent to a centralized server, which monitors the environment over the entire grid. This design can also be used to monitor children, elderly people and differently abled people when they are left alone. The loud speakers can be added to announce information and communicate with the people in a specific location or even used to threaten intruders.

1.1. New Application

Smart grids are always prone to accidents and faults. Unexpected situation might arise at any hour of the day and there are no solid methods to predict them yet. This paper addresses this issue by designing an AGV to monitor, control and also navigate people to specific locations in the field.

The assistive vehicle can be fabricated as a moving platform on which people can stand. Holding supports are attached to the platform. It is also equipped with haptic arm to perform as a suitable replacement for human interaction with the grids and also with different sensors like temperature sensors to observe the condition of the grids. The vehicle speed is adjustable by the user and it will not exceed 1.5 meters per second speed to avoid disturbances in the grid.

The AGV can be modelled using a path planning algorithm to navigate from a starting position and reach the desired destination by avoiding the obstacles in its path, thus guiding the engineers to their destination. The destination can be the faulty grid or the control region. The AGV can be equipped with camera to remotely control or rectify faulty grids. The environment around the AGV is sensed using SONAR sensors. The environment is mapped into a coordinate system and each grid destination is specified in the memory. Multiple AGVs can be used to monitor and control different locations of the grid. This involves coordination and control between the AGVs.

The transformers, relays and other equipments in the grid can be fitted with emergency sensors which alerts the AGVs during serious situation. The AGVs can be equipped with fire extinguishing systems and can be designed to climb stairs and move in rough terrains. This however requires a completely new physical design for the AGV. The further sections discuss a model for the AGV, different sensors which can be used, an algorithm which can be used to move the AGV from one location to another and also an real implementation of the navigation of the AGV.

2. Model

2.1. The Design

Autonomous Ground Vehicles are designed in numerous way in accordance with there application. The number of wheels, degrees of freedom, type of sensors, type of motors and rating of the power supply are some of the major factors which need to be taken into consideration while designing. Taking the example of Mars rover, it is designed with high power motors and multiple wheels to navigate in rough terrains. It is also equipped with high definition cameras and communication systems to take pictures and send it back to Earth. Solar cells are fitted on the rover for self dependent power supply. As discussed in the previous section, AGVs can be designed to navigate through rough terrains like stair case or rocky uneven surfaces in military regions or disaster struck places. Such AGVs are specially designed using belt wheels and arms to help in navigation[2]. AGVs can be modeled in two ways on basis of their area coverage– complete coverage of an area or travel from one point to another in a planned path. Lawn mowers are the best example for complete area coverage AGVs.

2.2. Sensors

The applications and performance of the AGVs improve with the additional information about its external environment. Various sensors such as infra-red (IR), SONAR, Cameras, accelerometers, thermometers, odometers and external global positioning sensors help the AGV to study its environment. One of the common sensors used in smart cities is infra-red. In this sensor a transmitter transmits an IR wave and a receiver receives it. When an obstacle is encountered the IR wave is reflected by the surface of the obstacle and the receiver senses it. One fact to be noted here is, bright colored surfaces reflect the IR rays, while dark colored surfaces absorb them. Some common applications of IR sensor are in automatic lighting systems, automatic doors and escalators. Cameras are very good surveillance systems. They can be used to capture a number of frames per second and digitally process them to sense movements or face recognitions. Global Positioning System is a satellite based sensor commonly used for navigation. Accelerometers sense the acceleration of the object on which it is mounted.

Among the variety of available sensors for AGVs, this project seeks the implementation of Sound Navigation and Ranging (SONAR) sensors. The reason for choice of SONAR as the sensor is that it can have a range around it to exactly sense the distance of the obstacle from the AGV. SONAR sensors work on the same principle as that of the hunting mechanism used by bats. Bats emit ultra sonic sound waves in their environment. These waves reflect back to the bat when they hit a prey. The bat uses this information to detect the distance and direction of the prey. The SONAR sensors also transmit such ultra sonic vibrations and receive them when they reflect back from an obstacle. Calculating the time taken for the reception, the distance between the AGV and the obstacle can be measured, and also SONAR sensors can be used to calculate the location of the AGV with respect to a reference origin[3].

3. The Algorithm

The goal of this paper is to navigate an AGV from an initial position to a final destination. Among the available algorithms the algorithm chosen for the path planning of the AGV is potential field algorithm[4]. This algorithm can be illustrated by pouring water onto a slope. The water flows from the region of higher potential to the region of lower potential. The rate of flow is determined by the steepness of the slope. The water flows along the path of least resistance avoiding any obstacle in the slope to reach the bottom, utilizing the least amount of mechanical work.

This same concept can be simulated for an AGV by creating an artificial potential field. The goal can be made to generate an attractive force for the AGV. To avoid the obstacles, they can be configured to emulate repulsive force to the AGV. At each instance the total potential field acting on the AGV is calculated. With both attractive force and repulsive force from the goal and the obstacles respectively, the AGV can be made to plan its trajectory[5]. Representing the total potential field mathematically[6],

$$U(q) = U(q)_{goal} + \sum U_{obstacle}\,(q) \tag{1}$$

'q' represents the current position of the AGV. The force induced by this potential field is

$$F = -\nabla U(q) = \left(\frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}\right) \tag{2}$$

Defining the attractive potential field of the goal as a parabolic function,

$$U_{goal} = \frac{1}{2}(K_{goal} \times \rho(q)^2_{goal}) \tag{3}$$

Where, K_{goal} is a constant and $\rho = \|q - q_{goal}\|$, is the distance between the current position and goal. The potential field for the obstacles can also be formulated in the same way as

$$U_{obstacle}\,(q) = \begin{cases} \frac{1}{2} K_{obstacle}\,(\frac{1}{\rho(q)} - \frac{1}{\rho_0})^2, & \rho(q) \leq \rho_0 \\ 0, & \rho(q) \geq \rho_0 \end{cases} \tag{4}$$

Where, $K_{obstacle}$ is a constant and ρ_0 is the influence factor. Partial differentiation of U_{goal} and $U_{obstacle}$ will result in calculating F_{goal} and $F_{obstacles}$ respectively. Thus the total force acting at each instance can be calculated as,

$$F(q) = F_{goal} - F_{obstacles} \tag{5}$$

Negative sign is provided to indicate the negative charge of the obstacles. One major problem in this algorithm is that the AGV can get locked in local minima. Taking the case of an U-shaped trap and simulating, the results of simulation are obtained as shown in Fig. 1.

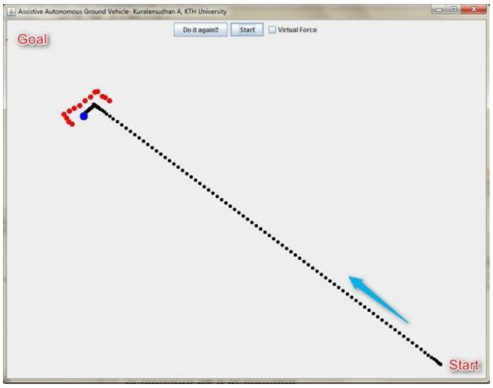

Fig. 1. Simulation of an AGV trapped in local minima (No axis required)

The attractive force from the goal and the repulsive force from the obstacle cancel out each other resulting in zero net force on the AGV. Thus the AGV is trapped in that position. To arrive at the conclusion that the AGV has been trapped, a position estimator is used to calculate the current position of the AGV. If the position does not change over a fixed interval of time, it reflects that the AGV has been trapped, which is shown in Fig. 2.

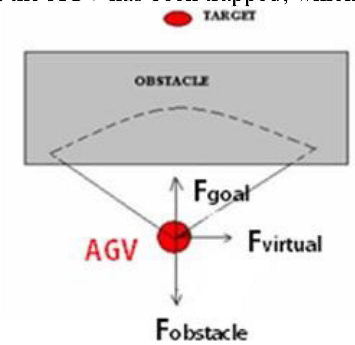

Fig. 2. The local minima trap situation

The solution to this problem is provided by implementing virtual force method. When the AGV gets trapped in local minima, an artificial force called virtual free space force is applied on it. This force is proportional to the amount of free space around the AGV and pulls the AGV out of the minima. The virtual force to be applied can be calculated as,

$$F_{virtual} = F_{constant} \; (\cos\theta \, e_x + \sin\theta \, e_y) \; [4] \tag{6}$$

Where, θ is the AGV to free space orientation and $F_{constant}$ is the force constant. The force acts in the direction opposite to that of the obstacle. This method indirectly alters the potential field and force acting at the position of local minima[5]. Thus, the total force can be written as,

$$F = F_{goal} - F_{obstacle} + F_{virtual} \tag{7}$$

Negative sign is provided to represent the negative charge of the obstacles.

4. Simulation

The simulation language used is JAVA and the environment is JAVA NETBEANS. The simulation has been inspired from a similar project [5] done in a different environment and modified the implementation in a different manner. The world consists of obstacles and an AGV. In the simulation the AGV and obstacles were considers as points of radius 0.8cm and 1cm respectively and SONAR was used as the sensor with a range of 5cm and beam width of 120 degree. The obstacles are initialized with negative charge to repel the AGV. The simulation is clocked and at each time frame the position of the AGV is calculated. Also the obstacles around the AGV are sensed at each time step and the trajectory of the AGV is planned using the resultant potential field, thus directing the AGV to its goal[5]. In a real time environment the objects such as chair and tables are not fixed, they can be moved, thus to address this issue the simulation provides option for adding obstacles during run time. An option is also provided to switch between using virtual force and normal algorithm. Noise is added in the simulation to accommodate for the disturbances in the real time implementation scenario. Fig. 3 and Fig. 4 show the Trajectory of AGV.

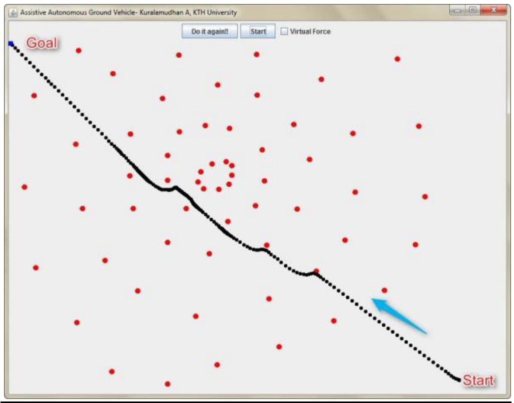

Fig. 3. Trajectory of AGV from the start to the goal by avoiding the obstacles in its path (No axis required)

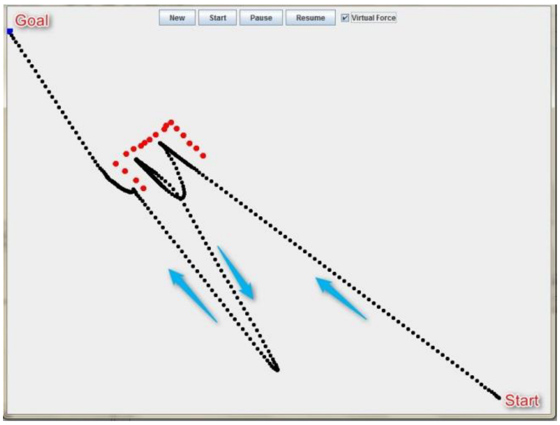

Fig. 4. Trajectory of AGV by avoiding an obstacle (No axis required)

5. Real Implementation

During the physical implementation, many external factors such as delays, slip of the wheel, accuracy of the sensors need to be considered. The sensors need to be placed with reference to the size and shape of the AGV to avoid collusion with the edges of the obstacles. A motion capturing system need to be used to sense the position of the AGV at constant intervals of time and use it to trace the path to the destination. Wireless communication system can be used to transmit this information to the AGV during run time. The Wireless communication can also be used for control and coordination of multiple AGVs. For the purpose of assistive AGV application mentioned in the introduction, the AGV-1 in the ground floor should communicate with the AGV-n in the n^{th} floor. The algorithm remains the same for both the AGVs, but the starting position and goal are split up differently before execution.

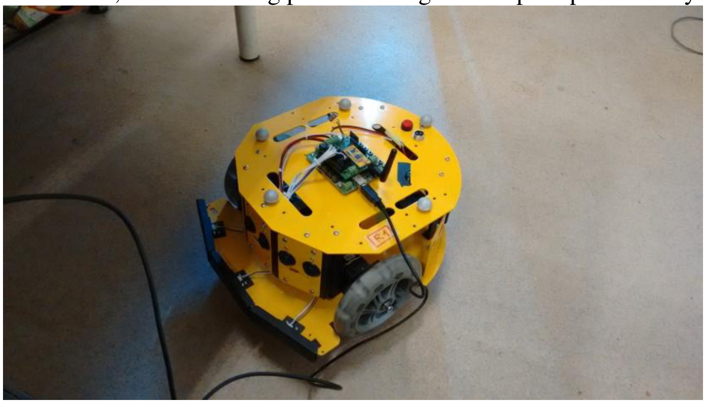

Fig. 5. The Nexus robot powered by Arduino

For this paper, Nexus robots shown in Fig.5 powered by Arduino microcontrollers was used for real time implementation of the algorithm. Qualisys- A motion capturing system at KTH Smart Mobility lab was used to track the position of the AGV, obstacles and the goal. The Qualisys maps the entire region and provides the coordinates of the AGV, obstacles and goal. The AGV was first calibrated with the reference origin in the Qualisys system. Then the system was configured to link with a personal computer via the internet. The coordinates and the angle of motion of the AGV and the obstacles were obtained from the motion capturing system through MATLAB libraries provided by Qualisys. This enabled the obtaining of the coordinates as a matrix in MATLAB.

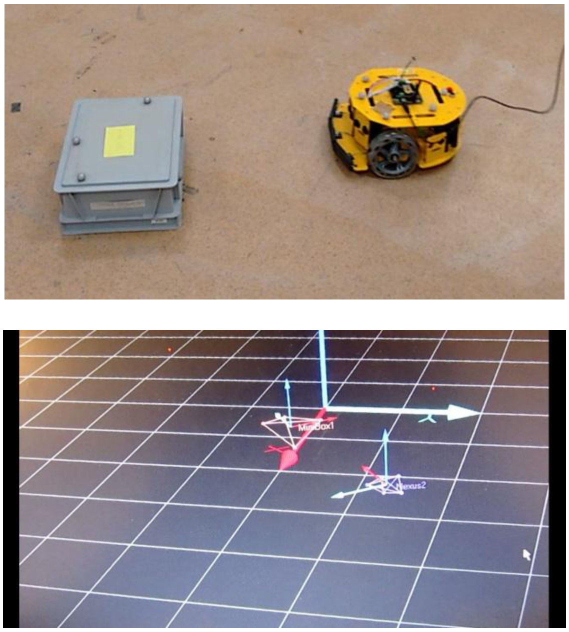

Fig. 6. The AVG facing the obstacle(grey box) in real world and Qualisys motion capturing system displaying the current positions of the AGV, obstacle and goal.

Moving on to programming the Arduino board, the motor ports were configured and commands were given for the movement of the AGV. Explaining this further : the Arduino was programmed to move the AGV forward if the character 1 was obtained as an serial input. Like wise 2- for moving back, 3- for turning left and 4- for turning right. The serial input was configured to be given through a text file. This was done using an Integrated Development Environment (IDE) called Processing IDE[7]. During run time, if a value is entered in this text file and saved, it would be reflected in the movement of the AGV.

The final step was to combine the Arduino and the MATLAB via the text file. The path planning algorithm was coded in the MATLAB with the calculations performed on the coordinates of the AGV, obstacles and the goal. The MATLAB provides the instructions for the movement of the AGV in the form of the numbers -1-Forward,2-Backward,3-Left,4-Right, then MATLAB was programmed to write these commands on to the text file. Explaining with an example- the command 1 is given by the MATLAB to the text file to make the AGV move forward. When an obstacle is detected in front of the AGV, 2 is written on to the text file to make the AGV move backward to avoid collusion. Thus, during run time the position of the obstacles and AGV keep updating and the commands are provided accordingly for the movement of the AGV to reach the goal Coordinate. This method of real implementation is unique and of own work. Fig. 6 shows the setup and results.

The working demo video of the AGV can be viewed in the following link : https://drive.google.com/file/d/0B5y9aqlgvS3qVEtxWEZ6WTlrUW8/view?usp=sharing

6. Discussion

In this paper, the potential field path planning algorithm for AGVs was simulated with and without the virtual force component and observations were noted. This algorithm was proved to be the simplest to implement when compared to other algorithms during literature survey[4]. The algorithm has few notable drawbacks. The path planning by this algorithm is not the shortest path to the goal. The trajectory of the AGV depends on the position of the obstacles and the goal. Thus the movement of the robot is random and does not follow any pattern(Fig. 4). The virtual force method helps the AGV to come out of the local minima, but it changes few capabilities of the original algorithm like moving through small space in between two obstacles. Defining the direction and force strength of the virtual force is random, it is more of an art than science[5]. The virtual force is directed opposite to that of the obstacle to escape from the local minima. Adding obstacles in the run time alters the potential field of the simulation and the trajectory changes respectively.The AGV follows random path even when the obstacles are not in its way. This is because of the fact that the obstacles induces a repulsive force to the AGV when it moves near them. The virtual force method is not a completely guaranteed solution of avoiding the local minima. At few orientations and field strength, the virtual force fails to drag the AGV out of the local minima. The speed of the AGV is not constant. It reduces as the AGV moves near to the obstacles because of the repulsive force from the obstacles. Thus, in a non-local minima position the AGV cannot stop for a long time. The real implementation was successful in making the AGV avoid obstacles and navigate to reach the goal. One important fact to be noted is that no sensor has been used and the complete algorithm was carried on using the coordinates provided by the motion capturing system, which has mapped the entire work area. This might not be feasible in real world, and in such case SONAR sensors can be used to find the exact distance of the obstacles and goal from the AGV. As expected from the algorithm, the path taken by the AGV was not the shortest path. Few random movements were also observed during the run time. There was a slight delay in updating the position of the AGV by the motion capturing system, which is unavoidable. One more observation made was, the random movements of the AGV was reduced when the motors ran slow due to battery discharge. This is because of the fact that the motor misses the exact position to change its direction of motion when commanded due to slip when running fast. Thus the program needs to calculate a different route again to complete the action. Running slow reduces the slip and the commands are executed at correct coordinates.

6.1. Future Work

The project provides a scope for a wide range of improvisations. One major field of work is to make the algorithm more efficient. A new idea for tuning the algorithm to make it plan the shortest path between the start and the goal positions need to be dealt with. During the real implementation the commands about the direction of motion was provided through serial connection via USB to Arduino. This could be made wireless using wireless transmission of the data. Another interesting work can be to try new methods to avoid trapping in the local minima. Virtual force is a random trial and error method and a more promising method to sense the local minima before being trapped need to be worked on. Physical implementation of the AGVs can be carried on with more advanced sensors and position tracking systems. Equipping the AGV with multiple sensors and connecting it to a centralized server can give way to a modern surveillance system which is more accurate and efficient.

7. Conclusion

The applications and model of Autonomous Ground Vehicles for smart grids has been studied and discussed. A two wheeled AGV was simulated using the potential field algorithm to navigate from a starting point to a goal by avoiding obstacles. The virtual force method has been studied and implemented to pull the AGV out of local minima traps. The same has been implemented in real time using Nexus robots by programming in MATLAB and Arduino successfully. This implementation is new and can be worked further for considering as a marketable product. The advantages and disadvantages of using this algorithm was discussed. A new application for the AGV has been suggested and the scope for future works on this paper was stated.

Acknowledgements

I would like to thank my project supervisors Stefan Magureanu and Jonas Mårtensson for their valuable support and guidance. I extend my gratitude to the Smart mobility lab staffs for helping me during the real implementation. I would also like to thank KTH Royal Institute of Technology, Sweden and Amrita School of Engineering, India for giving me an opportunity to work in my field of interest.

References

[1] H. Long, Q. Huihuan, C. Wing Kwong et al., System and design of a compact and heavy-payload AGV system for flexible production line., p. 2482-2488, 2013.
[2] W. Ming-Shyan, and T. Yi-Ming, "Design and implementation of a stair-climbing robot." p. 1-6, 2008.
[3] A. Carullo, and M. Parvis, An ultrasonic sensor for distance measurement in automotive applications, Sensors Journal, IEEE, vol. 1, no. 2, p. 143, 2001.
[4] F.-g. Ding, P. Jiao, X.-q. Bian et al., AUV local path planning based on virtual potential field.,p. 1711-1716 Vol. 4, 2005.
[5] H. Safadi. ,Local Path Planning Using Potential Field, 2015; http://www.cs.mcgill.ca/~hsafad/robotics/, 2007.
[6] R. Siegwart, Introduction to autonomous mobile robots, Cambridge, Mass.: Cambridge, Mass. : MIT Press, 2004.
[7] "Arduino Basics: Reading from a Text File and Sending to Arduino," 2015 http://arduinobasics.blogspot.se/2012/05/reading-from-text-file-and-sending-to.html, 2015.