ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 1, January 2014

Behavioral Strategy for Indoor Mobile Robot Navigation in Dynamic Environments

Ahmad Alsaab, Robert Bicker

Abstract*— Development of behavioral strategies for indoor mobile navigation has become a challenging and practical issue in a cluttered indoor environment, such as a hospital or factory, where there are many static and moving objects, including humans and other robots, all of which trying to complete their own specific tasks; some objects may be moving in a similar direction to the robot, whereas others may be moving in the opposite direction. The key requirement for any mobile robot is to avoid colliding with any object which may prevent it from reaching its goal, or as a consequence bring harm to any individual within its workspace. This challenge is further complicated by unobserved objects suddenly appearing in the robots path, particularly when the robot crosses a corridor or an open doorway. Therefore the mobile robot must be able to anticipate such scenarios and maneuver quickly to avoid collisions. In this project, a hybrid architecture control system has been adopted to navigate within dynamic environments. Experiments using the proposed control system on a Pioneer mobile robot showed that the mobile robot successfully avoided static dynamic obstacles. Furthermore, the mobile robot was able to reach its target within an indoor environment without causing any collision or losing the target.

***Index Terms*— Behavioral Strategies, Dynamic Obstacles, Mobile Robot, Unobserved Obstacles.**

I. INTRODUCTION

An autonomous mobile robot is an intelligent system which can navigate within its workspace without any human interference. In this context the robot must be able to perceive the surrounding environment through data collected from sensors, plan a trajectory which guarantees safe and accurate goal reaching, and then track this trajectory by generating suitable motion commands. The control architecture is ability of a mobile robot to develop and integrate its tasks. Many studies on mobile robot navigation has been developed, all of them aim to achieve robust flexible and reliable control architecture. However, the mobile robot architectures can be classified into three categories namely deliberative architecture, reactive architecture and hybrid architecture. The deliberative (top-down) architecture is considered the earliest intelligent mobile robot architecture which repeats a series of steps: sense, plan and act (SPA) [1],[2]. This architecture decomposes the robot tasks into five serial modules; perception, modeling, planning, execution and action. The SPA model has a powerful planning ability, but on other side fusing the sensory data to model the world which is sometime difficult to build and then planning a serial sequence of actions may produce a slow response. Also, if any task does not work properly the whole system may fail.

To overcome the weak points of the deliberative architecture, the reactive architecture decomposes the control architecture into multi _parallel tasks or behaviors, where each behaviour can access the sensor data and actuators directly [3]. However, It can classify the reactive architecture into two basic types namely subsumption architecture and motor schemas. In the subsumption architecture, the control model consists of multiply levels of competence, which are not total independent because the upper layer can control the lower layers by inhibiting the outputs or suppressing the inputs, that means the complexity of layer control function increases as the level number increases. Arkin developed motor schemas architecture [4] [5], which consists of independent tasks or behaviours, each behaviour produces its output vector depending on the sensory data, and then those output vectors are combined together to produces suitable motion commands.

To overcome the problems of lack of planning in the reactive architecture, the hybrid architecture combines the two previous architectures, and comprises three layers; the deliberative architecture forms the top level which is responsible for long-term planning or global path, localization and human interaction. The lower level is the reactive layer, which works to implement the global path in safe and accurate short path through fusing its behaviours, i.e. it implements local path planning. The intermediate layer is called the sequencer layer, which links the top and lower levels, and transfers the information from the top layer e.g. passing the waypoints, and observing the reactive layer to provide feedback to the top level [6],[7],[8].

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 1, January 2014
Localization, global path planning and obstacle avoidance are considered as key elements for autonomous mobile robot navigation. Many solutions have been proposed to solve the problem of mobile robot localization; these methods can be classified into two categories; relative localization and absolute localization. In relative localization the mobile robot location is calculated with respect to a known start location, where the location from this point is computed by integrating the wheel rotation speed measured by encoders [9],[10],[11]. This technique is simple, cheap and easy to implement in real time applications, but having a major disadvantage of error accumulation produced by the wheel slippage over long distance navigation. The absolute localization is more complex and more accurate than the relative localization. Different sensors can be employed for this method such as GPS, camera, ultra sonic and laser [12]. The global path planning means finding a safe and short trajectory from the current location to a destination. The path planning can be classified into either topologic planning or metric planning. In the first method, the route of the robot is determined by a set of landmarks, while in the second method, the robot path is divided into many sub goals [13]. Two examples of planning algorithms are A* search algorithm and wave front algorithm. The global path planning considers just fixed obstacles in the workspace map such as walls, but in the environment, the objects' locations may change. To achieve a collision-free navigation, the robot has to detect and avoid those obstacles. Mobile robot navigation with static obstacles is considered relatively simple because the robot needs to calculate only the distances to the obstacles; having an ample time to spend in making an optimal decision as to best approach to avoiding the obstacle. The three commonly used algorithms for static obstacle avoidance are Potential Field Algorithm (PFA), Vector Field Histogram method (VFH) and The Dynamic Window Approach (DWA) [14], [15], [16], [17], [18],[19], [20].

The linear velocity obstacle algorithm (VO) was proposed to avoid moving obstacles, where it defines the collision situation between two objects moving with constant velocities[21]. The non-linear V-Obstacle method was introduced as an extension of the VO algorithm to avoid obstacles moving on arbitrary trajectories [22]. For avoiding dynamic obstacles, the mobile robot has to collect information about the surrounding environment using sensors, clusters the sensors data and utilizes a tracking algorithm to estimate the velocities and locations of dynamic obstacles. Recently 2D laser sensor has been popular in the mobile robot application because it provides accurate information about the mobile robot workspace, requires short computation time and illumination does not affect its reading. There are two types of clustering laser data methods; the distance-based clustering methods [23],[24],[25] and KF-based methods[26]. The KF-based methods detect the segments and their directions precisely; however, they are more complex than the distance-based methods and require relatively large computation time.

The commonly used methods for tracking dynamic obstacles are the kalaman filter and particle filter. The extended particle filter algorithm proposed in [27] is simple to implement than the kalman filter for tracking multiply moving obstacles.

Dynamic obstacles could be other robots or humans within indoor environments. For tracking a human, two legs must be defined from the laser data. The changes in the relative distance between the two legs during the human walk and the changes in their shapes according to the trouser flexibility are considered the main challenges of the human detection algorithm. In [28], three parameters were proposed to determine a leg namely Girth, Width and Depth. They have not considered the situation when the distance between the legs is too short therefore the two legs may be interpreted as one leg. Furthermore their leg detection method may fail if a leg is hidden partly or totally behind another leg.

In this paper, hybrid control architecture has been proposed for mobile robot navigation in dynamic environments. A 2D laser sensor was utilized to collect information about the robot's workspace, where the laser data was segmented using the clustering method proposed in [25]. The extended particle filter algorithm (EXPF) proposed in [27] has been used to estimate obstacles' speeds. A method has been proposed for tracking humans. Also, the virtual obstacle principle has been proposed to avoid unobserved obstacles which may appear from open doors or corridor crosses.

II. HYBRID CONTROL ARCHITECTURE

Fig. 1 shows the block diagram of the proposed control architecture which includes three layers; the deliberative, intermediate and reactive levels. The deliberative level includes two models namely *long path planning*, and *localization*. The intermediate layer is response to transfer the information between the reactive level and deliberative level. The reactive model includes five behaviors namely *Go to Goal*, *Obstacle avoidance*, *unobserved obstacle avoidance*, *Human avoidance* and *Emergency*. The coordination model fuses the outputs of the reactive behaviours to produce the optimal control commands.

A. *Localization System*

The adaptive montecarlo Localization approach (AMCL) has been used in this paper to estimate the mobile robot position. AMCL has three inputs; the workspace map, external sensor data (laser data) and odometry data, while its output is the estimated position of the mobile robot. The particle filter is considered as the core of the AMCL localization system. More details of the AMCL system can be found in [29],[30].

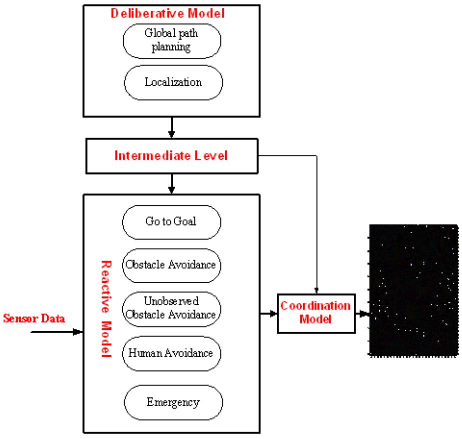


Fig. 1: Block diagram of the proposed control system

B. *Global path planning*

In this paper, the wave front path planning algorithms has been used to establish the global path. The inputs of this algorithm are the workspace map, estimated current location of the mobile robot and the goal location, while the output is a group of sub goals or waypoints, which represents odometry landmark.

C. *Go to Goal behavior*

This behaviour utilizes the current robot position coming from the localization system and the goal position to define the speed and direction to the target. In the global reference frame, the angle and the distance between the robot and the target are given as:

$$\theta_t = tan^{-1}(y_t - y_r, x_t - x_r) \quad (1)$$

$$d_t = \sqrt{(x_t - x_r)^2 + (y_t - y_r)^2} \quad (2)$$

Where (x_t, y_t) and (x_r, y_r) represent the target and robot position. d_t represents the distance between the robot and the target.

The reference speed to the target is given as following:

$$v_t = \begin{cases} v_{max} & if\ d_t > d_{th} \\ v_{max}\dfrac{d_t}{d_{th}} & otherwise \end{cases} \quad (3)$$

Where v_{max} and d_{th} denote the robot's maximum speed and threshold distance respectively.

D. *Obstacle Avoidance behaviour*

The Velocity Obstacle (VO) algorithm has been used to avoid the static and dynamic obstacles [14]. The VO approach is considered an easy and simple method to avoid moving obstacles. According to the VO algorithm, the robot is treated as one point by growing the obstacle by the robot radius as shown **Fig. 2**. There will be collision if the relative speed v_l between the robot and obstacle is located inside the collision cone. The left and right angles of the tangents are given as following:

$$d = \sqrt{(x_o - x_r)^2 + (y_o - y_r)^2} \quad (4)$$

$$L = \sqrt{d^2 - r^2} \tag{5}$$

$$\emptyset = tan^{-1}((y_o - y_r)/(x_o - x_r)) \tag{6}$$

$$\emptyset_l = tan^{-1}(\frac{r}{L}) + \emptyset \tag{7}$$

$$\emptyset_r = -tan^{-1}\left(\frac{r}{L}\right) + \emptyset \tag{8}$$

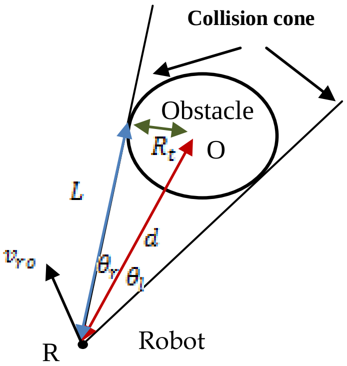


Fig. 2: collision cone

where d is the distance between the robot centre and the obstacle centre, L is the tangent length, $\emptyset$ is the angle between the obstacle and the robot in the global frame, $\emptyset_l, \emptyset_r$ are the left and right tangent angles, $(x_r, y_r), (x_o, y_o)$ are the robot and obstacle coordination.

A function has been proposed in this study to for weighing the robot velocities, in which all free collision velocities are given weight a numeric value of 1 and collision velocity weight is given as following:

$$W_{obs} = \begin{cases} 0 & if\ t_c(i) < t_{cth} \\ exp\left(-\frac{a}{t_c(i)}\right) * (1 - b * \delta * t_c(i)) & \end{cases} \tag{9}$$

t_c is the collision time for the robot velocity $(v_i, \omega$, a are constants.

$$\delta = min\ (\theta_l - \theta_i, \theta_i - \theta_r) \tag{10}$$

E. *Unobserved Obstacle Avoidance*

In indoor environments, some unobserved moving objects may appear suddenly in the robot path; particularly when the robot crosses a corridor or passes an open door. Therefore the robot has to consider these obstacles and implement an action to minimize the collision risk with them. In this project, a simple method introduces to meet the requirement. The virtual obstacle principle has been proposed to avoid unobserved dynamic objects, in which a virtual circular obstacle is created at the start point of each open door and corridor cross as shown in **Fig. 3**. The virtual obstacle radius is modified depending on the robot speed as the following equation:

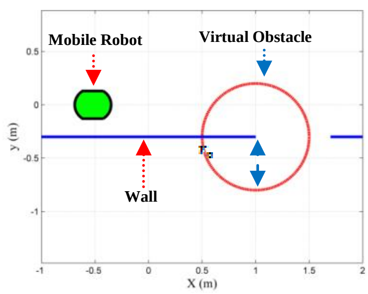


Fig. 3: Virtual obstacle

$$r_o = R.v_i \tag{11}$$

where is a constant positive value. represents the robot speed

The collision cone of the virtual obstacle is calculated depending on the Equations [4],[5],[6],[7],[8], while the robot velocities weights W_{un} are calculated by applying the weight function written in Equation (9).

F. Human avoidance behavior

Human avoidance requires fast and robust detection algorithm, in which a person is detected from 2D laser data by two leg clusters. The three parameters (width W, Girth G , and Depth d_{depth}) proposed in [28] have been used to detect the human leg. A laser cluster is classified as a leg if it meets the conditions $W_{min} < W < W_{max}$, $G_{min} < G < G_{max}$ and $d_{depth} < d_{max}$. As mentioned earlier, sometimes the leg detection algorithm may fail because some parts of the leg are hidden by the other leg, or the distance between the two legs may be too short. To overcome this problem, an algorithm has been developed in this paper, in which the laser clusters are classified into four classes as follows:

- L_0 if the cluster width is bigger than $2W_{max}$
- L_1 if the condition $W_{max} < W < 2W_{max}$
- L_2 if the condition $W_{min} < W < W_{max}$ is met, but other conditions of the leg are not met.
- The cluster which meets the conditions of the leg is classified as L_3.

Human tracking at the interval time n involves two steps; the first step is to associate the detected humans at the interval time $n-1$ with the laser clusters at *n*, while the second step is to detect new humans appearing in the laser data at *n*. The previous detected humans are associated with the nearest laser clusters in the current moment if one of the following situations is met:

- Two clusters of the class L3.
- Two clusters of the class L2.
- One cluster of the class L3 and another of the class L2.
- One cluster of the class L1.
- One cluster of the class L3 if there is no cluster of L3 or L2 within the human step.

A new human is defined by one of the following situations:

- Two clusters of the class L3.
- One cluster of the class L3 and other of the class L2.

In this behavior, the robot velocities weights W_{hum} are calculated by applying the weight function written in Equation (9).

G. Emergency behavior

The emergency behavior should be active and the robot must immediately stops if any of following scenarios occur; distance between the robot and obstacle become very short or all speed weights produced by the obstacle avoidance, human avoidance and unobserved obstacle avoidance behaviors are zero which means the robot does not have any free collision velocity. Therefore, the inputs of the emergency behavior are the sensor data and the maximum speed weight.

H. Intermediate Level

The intermediate level links the deliberative and reactive levels. In this paper, the intermediate level monitors the Go to Go behavior; when the distance between the current sub goal and the robot become less than a threshold distance, the next sub goal is passed. Also when the emergency behaviour becomes active and the robot stops, the intermediate level makes the robot to retreat until the emergency behaviour becomes inactive.

I. Coordination Model

The main task of this block is to fuse the outputs of the behaviors and then generate the control commands which minimize the collision risk and maximize the speed to the goal. The robot velocities are weighted depending on the output of the Go to Goal behaviour and the weights coming from the obstacle avoidance, human avoidance and unobserved obstacle avoidance behaviors. The weight of the robot velocity is calculated according to the following function:

$$Ws = \min\left(W_{unob}, W_{obs}, W_{hum}\right) \tag{12}$$

where W_{un}, W_c and W_{h_1} represent the robot velocity weights coming from the unobserved obstacle avoidance , obstacle avoidance and human avoidance behavior respectively.

To maximize the speed to the target and minimize the collision risk, the following weighting function is applied:

$$W_i = Ws_i * (a + \cos(\theta_t - \theta_i)) * (b - \|v_t - v_i\|) \tag{13}$$

where and are constants.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 1, January 2014

The maximum weight velocity $(v_c, \theta$ is chosen to produce control commands. The rotation speed is given as:

$$\omega_c = k_w . \theta_c \quad (14)$$

where l is constant

The linear speed is given as:

$$V_c = \begin{cases} v_c & if\ v_c < V_{cmax} \\ 0 & if\ 0 > V_{cmax} \\ V_{cmax} & otherwise \end{cases} \quad (15)$$

$$V_{cmax} = v_{max} - k_v . \frac{\omega_c}{\omega_{max}} \quad (16)$$

where is a constant, v_m and ω_m represent the maximum linear and rotation speeds of the mobile robot.

III. RESULT AND DISCUSSION

Experiments were implemented using two robots; the Pioneer P3-DX robot and Nubot robot which was made in Newcastle University. The maximum linear and rotation speeds of the robots were adjusted to 0.5 n and 100 degree respectively. In equation [3], the threshold distance was chosen to be $d_{th} = 1$. The parameters of the weighting function in equation [9] for the obstacle avoidance and unobserved obstacle avoidance behaviours were chosen to be =5, $b =$. The constant value in equation [13] was chosen to be 0.. The parameters and in equation [16] were assigned to the values 0.1 and 0.6 respectively. The parameters and were selected to be 1 and respectively. According to [28] the estimated values of the leg parameters are $d_{max} = 0.25$, $[W_{max}, W_{min}] = [0.09m, 0.18$ and $[G_{max}, G_{min}]$:0.091m,0.34m].

A. Signal obstacle avoidance

The intelligent mobile robot (Pioneer robot) avoided a dynamic obstacle (Nubot Robot) **Fig. 4**. The Nubot robot moved with a constant speed 0.5 n. The start positions of the robot and obstacle were chosen as $(x_r = 0m, y_r = 0$ and $(x_t = 5m, x_t = 0$ respectively. **Fig. 5** shows that The Pioneer robot was able to avoid the dynamic obstacle without any collision. It can be seen that the Pioneer robot moved on a straight line from $t =$ till t $= 3.$, and then avoided the moving obstacle, while at $t = 7.$ after the moving obstacle passed the robot turned to its goal.

B. Unobserved obstacle avoidance

Three tests were implemented for avoiding unobserved obstacles which may appear from an open door or corridor cross. For the first and second experiments, the initial position of the mobile robot and its target were chosen to be $(x_r = 0\ m, y_r = 0\ m)$ and $(x_t = 5\ m, y_t = 0\ m)$ respectively, while the start and end points of the door was chosen to be (x_{start}=2.6 m, y_{start}=-0.5 m) and $(x_{end} = 3.38\ m, y_{end} = -0.5m)$ respectively **Fig. 6**.

Fig. 4: The Nubot robot and Pioneer robot

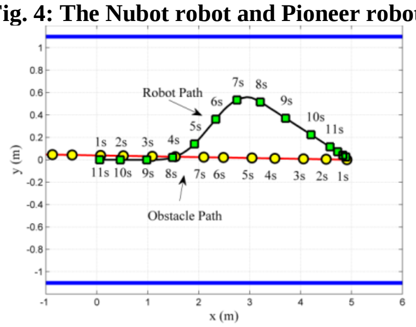


Fig. 5: The robot avoided a dynamic obstacle

Fig. 6: Avoiding unobserved obstacle may appear from an open door

For the first path shown in **Fig. 7**, the unobserved obstacle behavior was inactive. The mobile robot implemented a straight line path from its initial position to its goal. As result, the measured distance between the robot center and the door threshold at $x = 2.6\ m$ was $0.50\ m$. Therefore, any obstacle which may appear from the open door had $0.27\ m$ to collide with the robot (the robot width is $45\ cm$).

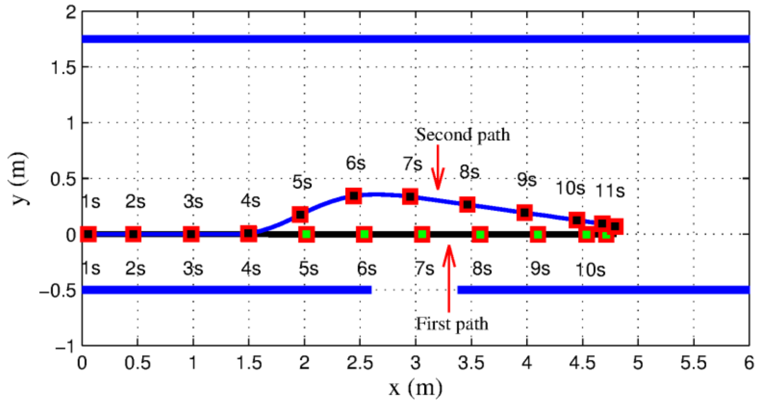
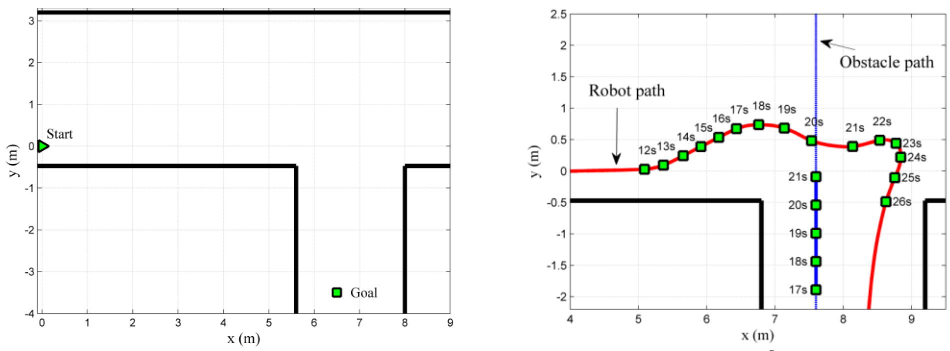


Fig. 7: The first path was implemented without using the unobserved obstacle behavior, while the second path was with using the unobserved obstacle behavior.

For the test, the robot path using the unobserved obstacles behavior was coincided with that without using it till $x = 1.5\ m$, after that the robot started to go away from the door threshold. The shortest distance between the robot and the door threshold was $0.75\ \mathrm{m}$. As a consequence the robot direction and the distance to the open door minimized the collision risk with any unobserved obstacle appearing from the open door.

For the third test **Fig. 8**.a, the robot had to reach its goal $(\mathrm{x} = 6.5\mathrm{m}, \mathrm{y} = -3.3\mathrm{m})$ without causing colliding with any obstacle. **Fig. 8**.b shows that the robot started to go away from the corner at the moment t=12s, while the dynamic obstacle appeared in the laser data at the moment t=17s, at the moment t=18s the robot turned right to maximum the speed to its goal, but at the moment t=20s, the robot turned left again to avoid the collision with the dynamic obstacle. After the dynamic obstacle passed the robot returned to its goal.



a b

Fig. 8: The mobile robot (Pioneer) avoided the unobserved dynamic obstacle (Nubot)

C. Avoiding a human and dynamic obstacle

In this test, the robot had to avoid an obstacle moving in the same direction (Nubot) and an obstacle moving in the opposite direction (Human). The speed of the Nubot robot was adjusted as 0.35 m/s. As shown in **Fig. 9**, the human appeared in the laser data at the moment t=4s and disappeared at t=7s. It can be seen that the robot changed its

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 1, January 2014

direction to follow the moving obstacle (Nubot) and avoid the human at the moment t=4.8s, and after the human passed, the robot started to pass the obstacle and reach the goal.

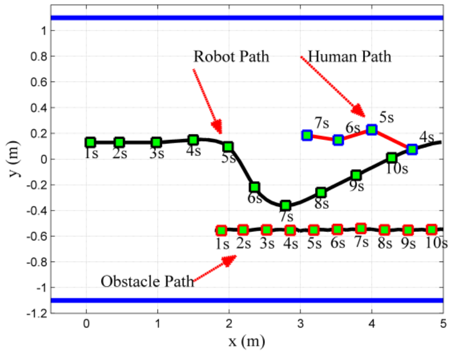


Fig. 9: The robot avoided an obstacle and human

D. Navigation in a crowded environment

In this test, the robot had to implement relatively a long path in a crowded environment (The ground floor of Stephenson Building, Newcastle university) **Fig. 10**. The workspace was divided into four reigns A, B, C, and D; the reign A was very crowded with the static obstacles, while the reign D was free of dynamic and static obstacles except the walls. In the reigns B and C, a lot of students were moving to their lecture rooms using stairs and left. The robot task was to travel from P0 (robotics lab) to P1, P2 and then return to the start point P0. **Fig. 11**.a shows the robot path from its start point to its target, in which the global path planning produced 5 waypoints; also the robot was able to reach its target without any collision or losing the target. It can be noticed that there is a lot of oscillations in the robot path produced during avoiding the static and dynamic obstacles in reigns A, B and C, while the path was smoother in the reign D. By zooming the area around the Waypoint-4 **Fig. 11**.b, It can be seen that the robot stopped, moved back and continued to its target. That was because a lot of students were passing the narrow gap, as consequence there was no enough space for the mobile robot to pass. Therefore the emergency behavior stopped the robot, while the intermediate level made the robot to move back. When the students made a space for the robot it continued to its target.

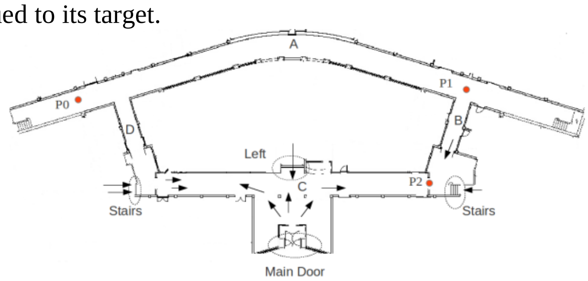


Fig. 10: The robot workspace (arrows refer to humans)

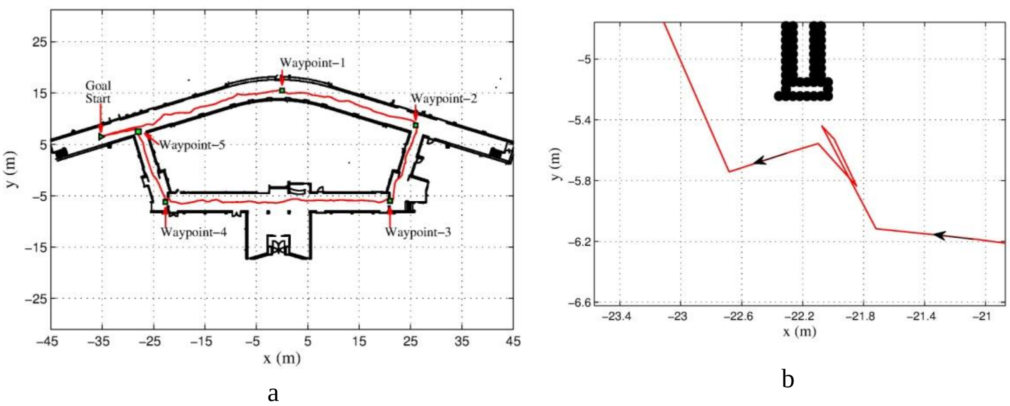


Fig. 11: The robot navigated in a crowded environment

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 1, January 2014

IV. CONCLUSION

This paper has proposed a robust hybrid control system for the mobile robot to navigate autonomously in dynamic environment. The control system has included three levels; the top level gave the ability for the mobile robot to estimate the current position and plan a safe long-term path. The behaviors in the low level worked together to implement the global path without causing any collision. The intermediate level linked the top and low level. A method was proposed to detect and track the human by classifying the laser clusters into four classes. Also, a simple method has been demonstrated to minimize the collision risk with unobserved obstacle that may appear form open doors or corridor cross. The experiments showed that the robot is able to avoid dynamic obstacle, unobserved dynamic obstacles including humans. The proposed control system allowed the robot to navigate in very crowded environment without any collision or losing the target. Finally many experiments using the proposed control system have been posted in the following website:
https://www.facebook.com/pages/Robotics-Group/173949736063997

REFERENCES

[1] R. Chatila and J. Laumond, "Position referencing and consistent world modeling for mobile robots," in *Robotics and Automation.* Proceedings*. 1985 IEEE International Conference on*, 1985, pp. 138-145.

[2] A. Pruski and S. Rohmer, "Robust Path Planning for Non-Holonomic Robots," *Journal of Intelligent & Robotic Systems,* vol. 18, pp. 329-350, 1997.

[3] R. Brooks, "A robust layered control system for a mobile robot," *Robotics and Automation, IEEE Journal of,* vol. 2, pp. 14-23, 1986.

[4] R. Arkin, "Motor schema based navigation for a mobile robot: An approach to programming by behavior," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, 1987, pp. 264-271.

[5] Y. Yan, Q. Zhu, and C. Cai, "Hybrid Control Architecture of Mobile Robot Based on Subsumption Architecture," in *Mechatronics and Automation, Proceedings of the 2006 IEEE International Conference on*, 2006, pp. 2168-2172.

[6] M. Milford and G. Wyeth, "Hybrid robot control and SLAM for persistent navigation and mapping," *Robotics and Autonomous Systems,* vol. 58, pp. 1096-1104, 2010.

[7] X. Li and l. Jiang, "Study on control system architecture of modular robot," in *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on,* 2007, pp. 508-512.

[8] P. Li, X. Huang, and M. Wang, "A Hybrid Method for Dynamic Local Path Planning," in *Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC '09. International Conference on*, 2009, pp. 317-320.

[9] Y. Jin-Xia, C. Zi-Xing, and D. Zhuo-Hua, "Dead reckoning of mobile robot in complex terrain based on proprioceptive sensors," in *Machine Learning and Cybernetics, 2008 International Conference on*, 2008, pp. 1930-1935.

[10] H. J. von der Hardt, D. Wolf, and R. Husson, "The dead reckoning localization system of the wheeled mobile robot ROMANE," in *Multisensor Fusion and Integration for Intelligent Systems, 1996. IEEE/SICE/RSJ International Conference on*, 1996, pp. 603-610.

[11] X. De, T. Min, and C. Gang, "An improved dead reckoning method for mobile robot with redundant odometry information," in *Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on*, 2002, pp. 631-636 vol.2.

[12] G. Reina, A. Vargas, K. Nagatani, and K. Yoshida, "Adaptive Kalman Filtering for GPS-based Mobile Robot Localization," in *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*, 2007, pp. 1-6.

[13] D. Kortenkamp and T. Weymouth, "Topological mapping for mobile robots using a combination of sonar and vision sensing," in *Proceedings of the National Conference on Artificial Intelligence*, 1994, pp. 979-984.

[14] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, 1985, pp. 500-505.

[15] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, 1991, pp. 1398-1404 vol.2.

[16] M. K. a. R. Chatila, "An extended potential field approach for mobile robot sensor-based motions," *Proc. of the Intelligent Autonomous Systems,* pp. pp. 490–496, 1995.

[17] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *Robotics and Automation, IEEE Transactions on,* vol. 7, pp. 278-288, 1991.

[18] I. Ulrich and J. Borenstein, "VFH+: reliable obstacle avoidance for fast mobile robots," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, 1998, pp. 1572-1577 vol.2.

[19] I. Ulrich and J. Borenstein, "VFH*: local obstacle avoidance with look-ahead verification," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, 2000, pp. 2505-2511 vol.3.

[20] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *Robotics & Automation Magazine, IEEE,* vol. 4, pp. 23-33, 1997.

[21] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *International Journal of Robotics Research,* vol. 7(7), pp. 760–772, July 1998.

[22] Fr, d, L. ric, L. Christian, and S. Zvi, "Navigation among Moving Obstacles Using the NLVO: Principles and Applications to Intelligent Vehicles," *Auton. Robots,* vol. 19, pp. 159-171, 2005.

[23] K. J. Lee, "Reactive navigation for an outdoor autonomous," in *Mechanical and Mechatroinc Engineering* vol. Master: University of Sydney, 2001.

[24] K. C. J. D. Dietmayer, Sparbert, Streller, "Model Based Object Classification and Object Tracking in Traffic Scenes from Range Images," *In Proceedings of intelligent Vehicles Symposium,* 2001.

[25] S. Santos, J.E. Faria, F. Soares, R. Araujo and U. Nunes, "Tracking of Multi-Obstacles with Laser Range Data for Autonomous Vehicles," in *3rd National Festival of Robotics Scientific Meeting (ROBOTICA)* Lisbon, Portugal, 2003, pp. 59-65.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 1, January 2014

[26] K. Rebai, A. Benabderrahmane, O. Azouaoui, and N. Ouadah, "Moving obstacles detection and tracking with laser range finder," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*, 2009, pp. 1-6.

[27] M. Marron, J. C. Garcia, M. A. Sotelo, D. Fernandez, and D. Pizarro, ""XPFCP": an extended particle filter for tracking multiple and dynamic objects in complex environments," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, 2005, pp. 2474-2479.

[28] C. Woojin, K. Hoyeon, Y. Yoonkyu, M. Chang-bae, and P. Jooyoung, "The Detection and Following of Human Legs Through Inductive Approaches for a Mobile Robot With a Single Laser Range Finder," *Industrial Electronics, IEEE Transactions on*, vol. 59, pp. 3156-3166.

[29] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*. vol. 2, 1999, pp. 1322-1328 vol.2.

[30] Z. Lei, R. Zapata, and P. Lepinay, "Self-adaptive Monte Carlo localization for mobile robots using range sensors," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, pp. 1541-1546.