2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)
19-20 December 2015, BUET, Dhaka, Bangladesh

Development of Intelligent Wheelchair Simulator for Indoor Navigation Simulation and Analysis

Rajesh Kannan Megalingam, Vishnu G B, Meera Pillai
Amrita Vishwa Vidyapeetham University, Kollam, Kerala 690525
rajeshkannan@ieee.org, vishnugb890@gmail.com, meerapillai186@gmail.com

***Abstract*— Disabilities affect majority of the world population nowadays. Autonomous wheelchair is one of the major assistive systems proposed for the disabled people to meet their daily locomotion challenges. Different algorithm and techniques can be used to find the optimal path in these types of intelligent wheelchair to navigate from a source to destination. Notable change in performance can be observed while testing with different algorithms and techniques. As the real world scenarios are so dynamic, replication of different test cases each time with the changing specification of the intelligent wheelchair for study is a strenuous process. Simulation study helps to analyze the different characteristics and behavior of the intelligent wheelchair without much strain. In this paper we describe the detailed study of fixed path algorithm and behavior of the autonomous wheelchair using simulation technique. We hope that such systems would be of great help in the rehabilitation process of the people affected by disaster, conflicts etc. apart from supporting elders.**

Keywords*— *Wheelchair, Simulation, elders, physically challenged

I. INTRODUCTION

Disability affects hundreds of millions of families in developing countries. According to W.H.O surveys, about 15% of the world's population lives with some form of disability, of whom 2-4% experience significant difficulties in functioning. The global disability prevalence is higher than previous WHO estimates, which date from the 1970s and suggested a figure of around 10%. People aged 65 or over were much more likely to be disabled (59 percent) than adults under 65 years (21 percent) or children under 15 years (11 percent). This phenomenon is receiving increasing attention from the scientific community and several solutions are being proposed in order to provide a more independent life to the people belonging to those groups. Intelligent wheelchairs were the result of attempt of assisting people with mobility problems. An intelligent wheelchair can be defined as a robotic device built from an electric powered wheelchair, provided with a sensorial system, actuators and processing capabilities[1]. At the same time, it is assumed that IW may present at least some skills such as autonomous navigation, autonomous planning, extended human-machine interaction, semi-autonomous behavior with obstacle avoidance, cooperative and collaborative behaviors[1].

Different algorithms may show different performance range when observed. In order to reduce the complexity in performance analysis of intelligent wheelchair, simulation, the imitation of the operation of a real-world process or system over time was found very useful. The act of simulating something first requires that a model be developed; this model represents the key characteristics or behaviors or functions of the selected physical or abstract system or process. The model represents the system itself, whereas the simulation represents the operation of the system over time[2]. Analyzing the major wheelchair simulator prototypes like Simulator of Powered Wheelchair, Virtual Env. Mobility Simulator, IntellSim[3], it was found that the prototypes are designed for interactive simulation which is comparatively complex and expensive.

In this paper, we are describing the detailed study on development of the intelligent wheelchair simulator to simulate various behavioral patterns of a fully automated intelligent wheelchair [4] with constrained specifications. Performance analysis is done based on time taken for simulating navigation and time taken for simulating the complete simulation obtained by simulating within the defined scope. These studies helps to verify several theoretical facts that we have consider for the development of a real-time intelligent wheelchair system without having the actual system.

II. PROBLEM DEFINITION

People become physically challenged due to various reasons. It might be due to birth, accidents, stroke, conflicts, terrorism disaster etc. Mobility being the primary problem for such people, developed countries support such people in more coordinated way compared to developing/ under developed nations. Autonomous wheelchairs would be of great help to certain category of those people who are severely disabled and who could not use continuous navigation methods. Performance of the autonomous wheelchair can be improved a lot by optimizing the algorithm used in auto navigation. Studying the performance each time with real-time implementation is a strenuous process. In this paper we address the problem of replicating the real-time implementation without having the actual implementation. The algorithm can be easily analyzed by obtaining the total time in simulating the navigation and the overall time taken for simulation.

III. RELATED WORKS

Nowadays extensive number of wheelchair simulators is available for research purposes. Craighead et al. [7] had spotted the strengths and weakness of 14 commercial and open source simulators. In our simulator we concentrate on providing a good support for analyzing the behavioral pattern of the intelligent wheelchair. Considering Carpin et al.'s [8] claim that the simulation of robotic platforms should not consider a robot as an isolated entity, but as an entity which interact and is affected by the environment where it is situated, to efficiently simulate the navigation speed we have to set up an interactive nature. But in our case the we are trying to remove the interactive nature of the wheelchair to reduce the cost. Simulation is done based on some assumptions. Factors like speed and orientation can be simulated efficiently only in an interactive nature because it is dynamic based on the rotation encoder, but we are concentrating to simulate factors that are less affected by the hardware and speed is kept fixed for our convenience.

IV. SYSTEM ARCHITECTURE

The system consists of a powered wheelchair, a microcontroller unit to control the motor drivers, a microprocessor unit equipped with a OS as shown in the Figure 1. The simulator is installed in the microprocessor unit. The user may use a gadget to use the simulator. The initial floor plan is given to the simulator as input to the simulator. User may specify static or dynamic obstacle in the path of navigation which may refer any kind of entity that may obstruct the wheelchair from navigation. The optimal path from a source to destination is obtained by a custom Location Aware and Remembering Navigation (LARN) which is developed on top of Dijkstra's algorithm. The simulation time and navigation time are obtained by using two timers at two different phases of the simulation process. The overall simulation timer starts first and keeps track of the time till the end of the total simulation which includes the optimal path calculation as well as time for simulating the navigation. The navigation time is obtained by starting the timer at the second phase, i.e. the simulation of navigation after optimal path calculation. It denotes only the time taken for simulating navigation of wheelchair from source to its destination.

In this paper we discuss about the implementation of the best case in which the path to be traversed is identified in advance by the user. The user is expected to choose the shortest paths between the sources and destinations. This path is stored in memory and retrieved as and when required by the wheelchair. The control mechanism used to select the destination is not specified here as the user may use switch based, gesture based, voice based etc., control mechanisms to select the destination. For example if a house has four rooms and the user is currently located in room 1, the user might want to go to room 2 or 3 or 4. For this purpose the user can used any of the control mechanisms as mentioned above.

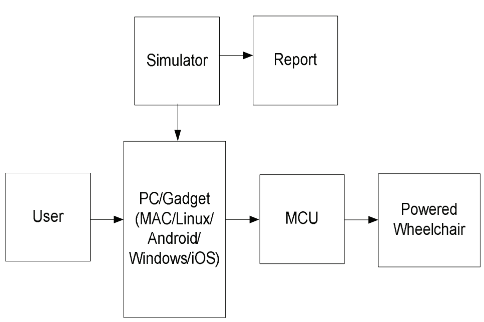


Fig. 1. System Architecture

The MCU is integrated with the rotation encoder, a digital compass and motor drivers for driving the powered wheelchair motors. The rotation encoder keeps track of the distance travelled, here the number of grids traversed by the wheelchair. In real time, the grid size depends on the wheelchair base dimensions. In our case we assume that the grid size is equal to the wheelchair base dimensions. The digital compass is used to find the orientation of the wheelchair and keep track the direction in which the wheelchair is moving. The wheelchair can move only in straight lines with 90o turns. It can either move left, right, reverse or forward directions only. It cannot move in diagonal directions.

The simulator can support three cases, the best case, the worst case and the hybrid case. The best case is where there is no algorithmic time is involved, in which user predefines the path between the sources and the destinations. The worst case is the case in which LARN algorithm is used where the path to be traversed is calculated only during the run time. In this case the user can choose any grid as source and any other grid as destination. The hybrid case is the one in which the user involves the fixed path case as well as the worst case. As mentioned earlier, in this paper we are going to discuss about the best case only.

A. Polygon of any shape and size

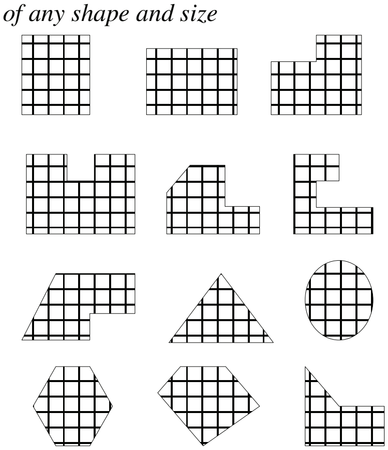
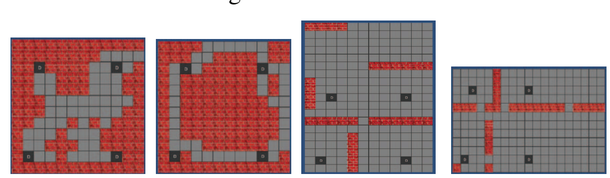


Fig. 2. Floor Plans of Various Shapes and Sizes

The simulator supports polygon of any shape and size for simulation within the given overall outer dimension provided at the initial stage. The user can create the house layout of any shape, including circles and triangles by placing obstacles in the floor plan. The walls and obstacles can easily be placed by the user on left mouse button click. The partial occurrence of walls/obstacle are also considered as obstacles which is said to occupy one grid completely. Some of the layouts/shapes supported by the simulator are shown in the figure 2. The maximum size of the layout supported by this simulator is 200x200 grids in the form of square. All the shapes shown below are to be created by the user within a square by appropriately placing the obstacles inside the square. Some of the sample floor plan generated by the simulator is show in figure 3.



(a) (b) (c) (d)

Fig. 3. (a) - (d) Sample Floor Plans created using the simulator.

B. Static and Dynamic obstacles

Within a house the obstacles are classified as static and dynamic obstacles. Static obstacles are those which are not going to be hindrance to the wheelchair path as their grid position will be known in prior. Wall, Dining table, Television stand, computer table etc. falls under the category of static obstacles. On the other hand, the obstacles which are going to hinder the smooth movement of the wheelchair while navigation are considered as dynamic obstacles. For example, while the wheelchair navigates, a person might come and stand in the path of the wheelchair. Or a bag might have been placed on the path. These are considered as dynamic obstacles which may be moved away later. Still they are obstacles for the simulator and hence it waits for a fixed time period on detecting a dynamic obstacle and again check for the existence of the dynamic obstacle. If the dynamic obstacle is still there, it is considered as a static obstacle and the path to the destination is recalculated. The simulator supports both static and dynamic obstacles. Static obstacles can be placed in any of the grids before the simulation starts. As the wheelchair starts traversing the grids as per the calculated path to reach the destination, the user can place an obstacle - which is dynamic, on its path.

C. Possible Path

Best Case - Fixed Path

In this scenario, the wheelchair always traverses in a fixed path between sources and destinations. The path to be traversed is chosen by the user. Each source can act as destination and vice versa. If there are N number of source/destinations then there are [N x (N-1)]/2 possible paths. For example, if there are three source/destinations then 3 possible paths are available for the user as shown in the figure 4. The blue color grids are obstacles. The immovable assets including walls inside a house are considered as obstacles. The user can place the obstacle in any of the grids prior to the start of the navigation. This case doesn't support dynamic obstacles, i.e. the user cannot place any obstacle after the wheelchair started the navigation.

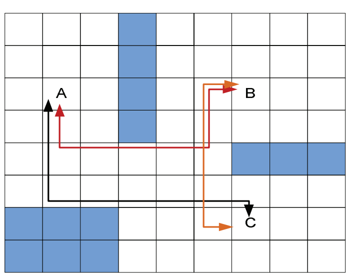


Fig. 4. An example floor plan with three sources/destinations and the possible shortest paths. We have 3 sources/destinations and hence 3 fixed paths are enough to travel among those paths.

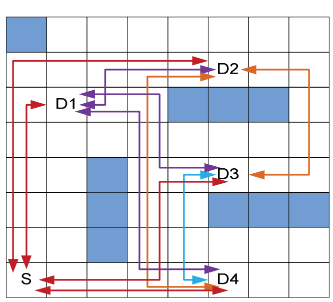


Fig. 5. Another example floor plan with five sources/destinations and the possible shortest paths. We have 5 sources/destinations and hence 10 fixed paths are enough to travel among those paths.

In the floor plan shown in figure 5, S, D1, D2, D3 and D4 are the grids chosen as source/ destination in five different room of the house. It is assumed that the house has five rooms and S, D1, D2, D3 and D4 are the source/ destination grids. The user has no option to go to any other grid in the house as destination. As mentioned earlier, the house layout can have any number of grids and any number of rooms. The two floor plans shown in the figure are the same except the path traversed that is shown in solid black lines, from S to D1 to D2 to D3 to D4. Note that if the user wants to reach D2 from D1, then D1 becomes the source and D2 is the destination. We see different paths taken to reach the destinations in both the layouts, though the destination grids are the same. As the path is already fixed and there is no algorithm involved in calculating the path to a destination, there is no delay incurred to start the wheelchair to auto navigate towards a particular destination.

V. RESULTS AND EVALUATION

Table I shows the simulation time and the navigation time for the different floor plans with four source/destinations. Different shapes are considered in this case: circle, hexagon, pentagon, rectangle, square, triangle, and three random shapes. The source/destinations might be inside each room in the house. These timings are listed in millisecond units and we see a difference between the simulation time and navigation timings.

Simulation time includes algorithm path calculation time and the time taken by the wheelchair to reach the destination. Navigation time is only the time taken by the wheelchair to reach the destination. But we don't see a difference between simulation time and navigation time in certain cases. This is due to several factors. First the timings are measured and listed in terms of milliseconds and in some cases the algorithm path calculation time is in terms of microseconds. As the execution of the source code is sequential in nature, for some conditional check to become true, the CPU has to execute several instructions prior to reaching this conditional instruction. These two situations are the primary reasons for the zero difference and one to two milliseconds difference between the simulation time and the navigation time.

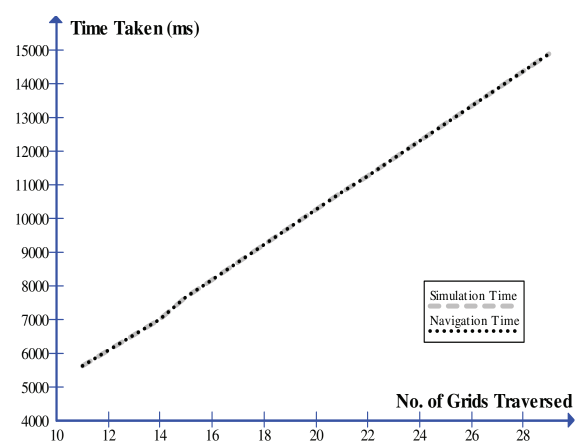


Fig. 6. Plot between the number of grids traversed and the time taken to reach the destination best case

TABLE I

SIMULATION TIMINGS OF DIFFERENT SHAPES FOR THE BEST CASE

Layout Shape	No. of Grids traversed to reach D1	Time taken to reach D1 (Simulation, Navigation) in ms	No. of Grids traversed to reach D2	Time taken to reach D2 (Simulation, Navigation) in ms	No. of Grids traverse d to reach D3	Time taken to reach D3 (Simulation, Navigation) in ms
Circle	13	6644, 6641	6	3057, 3057	19	9763, 9763
Hexagon	17	9219, 9218	20	9210, 9209	21	10801, 10801
Pentagon	10	5128, 5128	11	5647, 5647	17	8196, 8196
Rectangle	18	9316, 9313	14	7186, 7185	23	11776, 11775
Square	11	5631, 5629	16	8241, 8240	18	9256, 9255
Triangle	10	5114, 5111	10	5149, 5149	9	4630, 4629
Random1	11	5624, 5622	25	12871, 12871	21	10778, 10778
Random2	11	5620, 5618	21	10818, 10818	29	14926, 14926
Random3	9	4620, 4618	19	10284, 10284	32	16929, 16929

The graph in the figure 6 shows linearly increasing characteristics which is as expected. As the number of grids traversed increases, the simulation time too increases. There is a very little difference in simulation and navigation timings of 1 to 3 milliseconds which can be ignored compared to the simulation timings in the ranges of 10s of thousands of milliseconds.

The characteristics plot between the number of grids traversed and the time taken to traverse from source to destination is illustrated in figure 6. The blue color line indicates the navigation time and the red dots indicate the simulation time. As the difference between these two timings is in the range of 0 to 3 ms, whereas the Y axis time units is in the scale of 1000 time units, there is no significant difference between these two timings observable in the plot. The figure 7 shows the characteristics of the time difference between the navigation and simulation timings. The zero time difference is not included in this one. Only the time difference in the range of 1 to 3 ms is between simulation and navigation time for the grids traversed is shown in the figure 7.

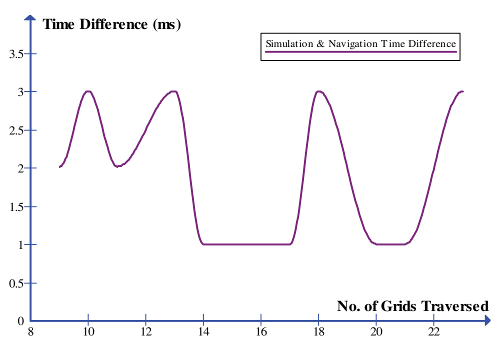


Fig. 7. Time difference between the simulation time and navigation time

VI. CONCLUSION

As the intelligent wheelchair simulator does not use any algorithm for optimal path calculation, the time taken for path calculation is zero. Theoretically the navigation time and simulation time should be equal in the best case which implies stable performance of the simulator. The result obtained from sample simulation setup shows a small amount of variation in some cases due to the delay in stopping timer due to the sequential nature of the program or overload in hardware of the device in which the simulator operates or host operating system delays. The chi-square testing done on the sample data ensure the validity of the data by giving an error percentage of 0.001148%, a negligible amount of error which means a higher rate of accuracy of the simulator. In the work presented here we have used only the best case where in the user identifies the shortest path in advance. The tests were carried out a simulation level only. The integration of the simulator with the wheelchair is currently going on. The worst case and the fair case also to be tested.

ACKNOWLEDGEMENT

We wish to thank Almighty God who gave us the opportunity to successfully complete this venture. The authors wish to thank Amrita Vishwa Vidyapeetham and the Humanitarian Technology Lab of ECE dept., for aiding us in this endeavor.

REFERENCE

[1] Luis Paulo Reis, Rosaldo Rossetti, Marcelo Petry, and Antonio Paulo Moreira, "Intelligent Wheelchair Simulation:Requirements and Architectural Issues", Proceedings of the 11th International Conference on Mobile Robots and Competitions, Robotica 2011

[2] Banks J. Carson, B. Nelson, D. Nicol (2001). "Discrete-Event System Simulation", Prentice Hall. p. 3. ISBN 0-13-088702-1.

[3] Brígida Mónica Faria, Luís Paulo Reis, and Nuno Lau, "A Survey on Intelligent Wheelchair Prototypes and Simulators", Springer, New Perspectives in Information Systems and Technologies, Volume 1, pp 545-557, 10.1007/978-3-319-05951-8_52

[4] Rajesh Kannan Megalingam, Deepak D, P R Ananthakrishnan Abhiram Thejus S, " LARN: Indoor navigation for elderly and physically challenged", 2013 IEEE GHTC, Pages: 326 - 330, DOI: 10.1109/GHTC.2013.6713705.

[5] "Development of Indoor Navigation Software for Intelligent Wheelchair" Uğur YAYAN1, Bora AKAR, Fatih INAN, Ahmet YAZICI

[6] Ojala, J., Inoue, K., Sasaki, K. and Takano, M. (1991), Development of an Intelligent Wheelchair Using Computer Graphics Animation and Simulation. Computer Graphics Forum, 10: 285–295. doi: 10.1111/1467-8659.1040285

[7] J. Craighead, R. Murphy, J. Burke, B. Goldiez, "A survey of commercial and opensource unmanned vehicle simulations", in IEEE Int. Conf. Robotics and Automation, Rome, Italy, pp. 852-857, 10-14 April 2007. ISBN: 1-4244-0601-3.

[8] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, C. Scrapper,"Usarsim: A robot simulator for research and education", IEEE Int. Conf. Robotics and Automation - ICRA, Rome, Italy, pp. 1400-1405, April 2007. ISBN: 1-4244-0601-3