Indoor Navigation Framework for Mapping and Localization of Multiple Robotic Wheelchairs

Yasith Lokuge
School of Computing
University of Colombo
Colombo 07,Sri Lanka
yasith1@gmail.com

Prashan Madumal
School of Computing
University of Colombo
Colombo 07,Sri Lanka
chikuwaprashan@gmail.com

Tharindu Kumara
School of Computing
University of Colombo
Colombo 07,Sri Lanka
hatkumara@yahoo.com

Naveen Ranasinghe
School of Computing
University of Colombo
Colombo 07,Sri Lanka
deshh.max@gmail.com

***Abstract* — The Indoor Navigation Framework we have proposed allows any wheelchair user to be guided to a desired location on his own, as long as the building itself is adopted to the novel system. Unlike the state of the art, where no automation exists for guiding a wheelchair in modern buildings, this allows the owner of the least sophisticated wheelchair to explore public buildings with more ease than a normal person. In this project we intend to implement a working prototype, which fulfills the fundamental functions of the system. It can be further customized to the needs of system owners.**

Keywords — Framework, Indoor, Localization, Mapping, Navigation, Robotics, Wheelchair

I. INTRODUCTION

Use of wheel chairs among people with walking disabilities is common in most countries, and is considered to be the aid of highest sophistication for persons with such impairments, apart from high tech artificial limbs, which tend to be extremely costly. Despite wheel chairs allowing such physically impaired people in moving to a place of their desire, they consist of several shortcomings as well. For instance, a person who appreciates independence would want to perform his/her daily chores without the assistance of another, but would be deprived of such ability, since doing all the work while burdened with the necessity of guiding the wheel chair itself would be almost impossible. More often than not, such persons require the assistance of attendants to perform their work and maneuver the wheelchair in public places such as hospitals, supermarkets, offices etc. Even though most such modern buildings focus on improvements in facilities and accessibility methods for the benefit of such physically impaired people; lack of interest towards introducing an automated wheel chair system itself has greatly limited the conveniences which could be granted to users of wheel chairs.

When discussing the prevalent situation in the majority of countries when it comes to measures they've taken towards ensuring ease of access in public buildings for users of wheel chairs, there are two important aspects one must take into consideration; improvements made to the buildings itself, and the evolution of wheel chairs to serve the purpose.

Design manual proposed by the United Nations [1] for creating a barrier free environment gives a list of specifications, which are expected to be adhered to, during the construction phase of public places. Most significant among these specifications is the inclusion of accessible entrances (wheelchair ramps) and/or elevators which could accommodate wheel chairs, in addition to stairways, since moving to higher levels in buildings is a major hindrance to such persons.

The conventional wheelchair has seen numerous changes through the years, and has been highly customized to suit various purposes according to each individual's lifestyle. For instance, there exist electric powered wheel chairs, which are moved via the means of electric motors and navigational control, often a small joystick mounted on the armrest. Furthermore, all-terrain wheelchairs allow users to wade in water, while providing with better mobility on uneven terrain such as beach, sand or snow. Various other developments too exist which are not necessarily related to the aspect of mobility; hence such advances have not been mentioned here.

Despite the convenience of disabled persons being taken into consideration when building modern buildings, and the wheelchair itself having seen vast improvements [2], certain short comings are still apparent in measures taken to improve accessibility to public buildings for users of wheelchairs: (a) Lack of relation between advances of buildings and the wheelchair itself have lessened users acquiring the most out of such improvements (b) If the wheelchair user happens to be in unfamiliar surroundings, and is unaware of where specifically the location he desires to go to is located, he may face with immense inconvenience having to maneuver a wheelchair all over a building. No thought has been given to this matter, when adopting the existing improvements. (c) Even though there do exist electric wheelchairs, most persons with disabilities prefer using the conventional ones, mainly due to the cost of more sophisticated ones. Maneuvering a wheel chair on your own in a massive building would be next to impossible, and the assistance of another would be indispensable. Hence, independence of wheelchair users has not been ensured at all instances, in these adaptations.

2166-0662/14 $31.00 © 2014 IEEE
DOI 10.1109/ISMS.2014.39

II. RELATED WORK

Several algorithms have been proposed for multi robot mapping such as, Simultaneous Localization and Mapping (SLAM), Kalman Filter based implementations, Balanced Distributed Graph-based Framework [3] etc. But the main disadvantages of those algorithms are complexities and the need of much expensive hardware infrastructure. Position measurements acquired by those methods are not accurate and more erroneous. For an example, need of expensive laser radar obstacle detection sensors for SLAM can be taken as a main disadvantage. Several adaptations of single robot mapping techniques have been proposed for the multi-robot context. The standard Extended Kalman Filter is not considered suitable for multi-robot extension [4] due to the quadratic complexity of the algorithm that would be reflected into a quadratic complexity of the communication.

III. MAPPING ALGORITHM

Our mapping and localization is based on processing an image of the scaled building map fed to the central monitoring computer of the system. It creates set of nodes in the areas which can be navigated by the wheelchair itself. Those nodes are in size of the robot, and uses our own modified Dijkstra's shortest path algorithm. We are having both map and the static obstacle matrix of the building. First we run Dijkstra's algorithm and save the shortest path, then

```
void Graph::dijkstra(Vertex s){

      Vertex v,w;
      Initialize s.dist = 0 and set
      dist of all other vertices to
      infinity

      while (there exist unknown
      vertices, find the one b with
      the smallest distance)
      b.known = true;
      }
      for each a adjacent to b
      if (!a.known)
            if (b.dist + Cost_ba <
            a.dist){
            decrease(a.dist to=
            b.dist + Cost_ba);
            a.path = b;
            }
      }
}
```

it calculates the turning points. Finally connect each turning point if there is no obstacle between them (by examining obstacle matrix) Pseudo code of the basic Dijkstra's algorithm is as shown above.

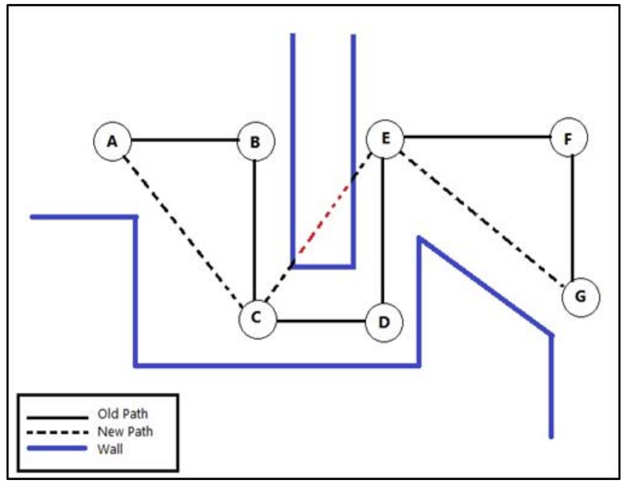

Fig. 1. Mapping by Connecting Nodes

Initially, the path is planned connecting A→ B→ C→ D→ E→F→G nodes. Then the optimization algorithm search for the obstacles between adjacent nodes, if there is no obstacle exists, path is rearranged by connecting next node as shown in Fig. 1. Ex. A→B→C is rearranged to A→C. But if there is an obstacle exists as in C → E path, it is considered as an invalid route. So it doesn't change the initial path and sticks to the C→D→E route.

IV. METHODOLOGY

Initially, a scaled map of the premise will be fed to the central monitoring computer. This is also there on the onboard control computer of the wheelchair. When we know the initial position, real time location of the wheelchair can be detected through the two encoders at the wheels and the digital compass set to the wheelchair. When the user needs to go to a specific location, he only needs to enter that location through onboard computer GUI. Then that computer will calculate the best possible path to the said location and the wheelchair will be navigated there.

Ultrasound sensors are installed in the system to identify obstacles. In case the chair runs into one, a signal would be sent to the control computer and it will calculate an alternate route which will then be used by the wheelchair for navigating.

Having only the encoders and the compass will not be sufficient in real time mapping and localization of the robot. There could be slight differences in the position mapped at the central monitoring computer and the actual physical position of the robot due to errors of the compass and the two encoders. In order to rectify this, UV markers will be printed on the floor to detect the exact X and Y coordinates of the robot.

If there arises a time when both the user and the wheelchair system has no definite idea about the position they are at (in case of lost or reboot), the wheelchair can be navigated to the closest anchor point with the aid of invisible UV markers on the floor of the building. The exact position of the system can then be identified from the RFIDs at the anchor points.

There are four types of UV markers. Halt marker is used to separate a section of a building to which the robot doesn't have access. Section Divide marker is used to separate different sections of a building. X Coordinate marker is used to make the robot aware of the X coordinate and to resolve the position error. Y Coordinate marker is used to make the robot aware of the Y coordinate and to resolve the position error.Once a marker type is detected by the photo transistor array installed with the system and is processed by the microcontroller, it is then send to the onboard control computer to aid in making further decisions.

Central Monitoring layer is responsible in controlling multiple wheelchair units at the same time. And the current position of the robots will be mapped repeatedly in the central monitoring software, in order to determine the places with heavy traffic, collisions and other information relevant to navigation.

V. LAYERED ARCHITECTURE

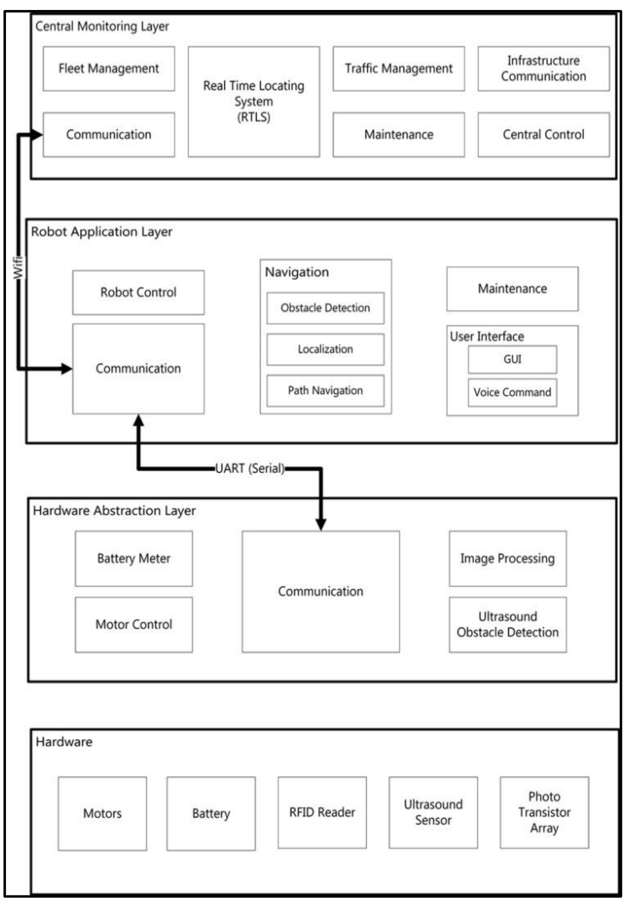

Fig. 2. Layered Architecture of the Entire System

VI. CENTRAL MONITORING LAYER

A. *Real Time Location System*

Calculates the real time location of the system using following core features.

B. *Fleet Management*

This system takes care of the following tasks:

- Receiving robot location
- Receiving dispatcher status
- Handling user/dispatcher requests
- Returning the total number of units in the network (from the length of the vehicle array)
- Sending the idle units to a dispatcher according to their statuses.

C. *Communication*

This system communicates with the Robot Application Layer through Wi-Fi, using a pre-defined structure. Many tasks are there of achieved through this. They are as follows:

- Receiving the location of the wheelchair unit from the server
- Sending map information to the unit through the robot and updating the infrastructure map of the said unit as needed.
- Sending information on traffic in the paths from the server to the unit.
- Sending user/fleet requests.
- Sending emergency details from the server when needed.
- Sending the status of the robot to the server.
- Connecting wheelchair phone to a human operator if needed.
- Sending the firmware from the server.
- Registering the wheelchair unit in the network when it is turned on.

D. *Traffic Management*

This is the system responsible of checking for possible collision routes using the all-paths array and the vehicle array. It will also reroute the unit path to avoid collision according to the priority of the requests.

E. *Maintenance*

This system checks the health of the vehicle from the vehicle array and also checks the availability of the path, because it is possible that a path is closed at a certain time (for cleaning purposes Etc.). It also takes care of updating the firmware of the robot and updating the map when needed.

F. *Infrastructure Communication*

Accessing and controlling the doors and elevators is done by this system.

G. *Central Control*

This system manages the emergency override of a robot if needed. (In case of fire etc.)

VII. ROBOT APPLICATION LAYER

A. *Robot Control*

Calculates which signals should be sent to the communication layer. Ex. go forward, turn left, turn right, and stop commands.

B. *Navigation*

- Obstacle detection: Reads ultrasound data and calculates the safe zone/ critical zone according to standards.
- Localization: Localization consists of the following tasks. Getting the values of grid junction count and wheel rotation count and using these to locate the robot in the map. Getting the values of anchor points. Set the value of according to the anchor point Signal received. Identify the unique anchor point according to the signal and sets its info. Returns the anchor point details, Using anchor point set the absolute location. Returns absolute location. Guide vehicle to next anchor point from anchor marking info.
- Path Navigation: Updates the location in the map, sets the path and calculates alternate paths.

C. *Maintenance*

Sets the critical Battery Level, monitors motor status in given time intervals.

D. *User Interface*

Interacts with the user through the GUI or through voice commands.

VIII. HARDWARE ABSTRACTION LAYER

A. *Battery Meter*

Checks the battery level and informs the robot application layer when a command is received.

B. *Motor Control*

Controls the direction and speed of the motor using PID based calculations according to the speed and distance commands given by the robot application layer.

The PID control scheme is named after its three correcting terms, whose sum constitutes the manipulated variable (MV). The proportional, integral, and derivative terms are summed to calculate the output of the PID controller. Defining u (t) as the controller output, the final form of the PID algorithm is:

$$\boldsymbol{u(t) = MV(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{d}{dt} e(t)}$$

Where,
K_p: Proportional gain, a tuning parameter
K_i: Integral gain, a tuning parameter
K_d: Derivative gain, a tuning parameter
e: Error = SP - PV
t: Time or instantaneous time (the present)
τ: Variable of integration; takes on values from time 0 to the present t.

C. *Communication*

Communicates with the robot application layer through a predefined packet structure using the UART serial protocol.

D. *Image Processing*

Processes the raw data acquired by the photo transistor array and searches for marker types.

E. *Ultrasound Obstacle Detection*

Searches for obstacles during set time intervals and detects them using ultrasound sensors. (Time interval can be changed as needed)

IX. CONCLUSION

In this paper we proposed an Indoor Navigation Framework for Mapping and Localization of Multiple Robotic Wheelchairs. This system is implemented and tested so we found that this will be useful for the modern buildings to improve their accessibility methods. We made a hardware prototype and it worked meeting all project goals.

ACKNOWLEDGEMENT

This research is mentored by the Zone 24x7 Pvt. Ltd and University of Colombo School of Computing.

REFERENCES

[1] Department of Justice, United States, “2010 ADA Standards for Accessible Design”, September 15, 2010.

[2] H.A. Yanco, A. Hazel, A. Peacock, S. Smith and H. Wintermute. “Wheelesley: a robotic wheelchair system”, In *Proceedings of the IJCAI-95 Workshop on Developing AI Applications for the Disabled*, Montreal, Canada, August 1995.

[3] Dario Lodi Rizzini, Stefano Caselli, “A Balanced Distributed Graph-based Framework for Multi-Robot Mapping”, *IROS Workshop on Probabilistic Graphical Models in Robotics,* Graphbot 2010, October 22.

[4] E. Nettleton, H. Durrant-Whyte, P. Gibbens, and A. Goektogan, *Sensor Fusion and Decentralized Control in Robotic Systems III*. G.T. McKee and P.S. Schenker, editors, 2000, vol. 4196, ch. Multiple platform localization and map building, pp. 337–347.