2020 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
April 15-16, Ponta Delgada, Portugal

Driverless Wheelchair for Patient's On-Demand Transportation in Hospital Environment*

André Baltazar*, Marcelo R. Petry†, Manuel F. Silva‡†, and António Paulo Moreira*†
*Departamento de Engenharia Eletrotécnica e de Computadores
Faculdade de Engenharia da Universidade do Porto, Porto, Portugal
†CRIIS - Centro de Robótica Industrial e Sistemas Inteligentes
INESC TEC - Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência, Porto, Portugal
‡Departamento de Engenharia Eletrotécnica
Instituto Superior de Engenharia do Porto, Porto, Portugal
0000-0003-4047-1395, 0000-0002-7023-8562, 0000-0002-0593-2865, 0000-0001-8573-3147

***Abstract*—The transport of patients from the inpatient service to the operating room is a recurrent task in the hospital routine. This task is repetitive, non-ergonomic, time consuming, and requires the labor of patient transporters. In this paper is presented the design of a driverless wheelchair under development capable of providing an on-demand mobility service to hospitals. The proposed wheelchair can receive transportation requests directly from the hospital information management system, pick-up patients at their beds, navigate autonomously through different floors, avoid obstacles, communicate with elevators, and drop patients off at the designated destination.**

***Index Terms*—Intelligent Wheelchair, Mobility-On-Demand, SONHO**

I. Introduction

In a hospital environment there are many services where external consultations, examinations, analysis and surgeries are carried out that involve the displacement of patients that go to these services. This deserves some attention especially in situations of people with reduced mobility, such as citizens with disabilities or fractures / problems in lower limbs and elderly. Currently, the transportation of patients in hospitals is done by patient transporters, non-specialized personnel, which is not very pleasant. It could often be done using a wheelchair, and that would be more pleasant and done in an autonomous way. As a result, it would avoid delays and failures due to the lack of personnel to carry out the transportation.

In this context, and in the scope of a collaboration with the Shared Services of the Health Ministry (Serviços Partilhados do Ministério da Saúde - SPMS), it was proposed to study and develop a prototype of a system that is capable of helping people to be transported. Given this idea, it was decided to develop this prototype according to the concept of an intelligent wheelchair, capable of autonomous navigation with safety and avoiding obstacles, flexible and robust interaction with the patients and also communicating with the hospital system, other devices and other wheelchairs.

The transport of inpatients from the urology service to the operating room was presented as a good option to implement this prototype because there is the possibility of the patient making the journey sitting on a chair instead of lying in bed, where he would have a much more boring or stressful transfer.

Bearing these ideas in mind, in this paper is presented an approach to integrate a driverless wheelchair (named Connected Driverless Wheelchair in the scope of this project) in an hospital environment. With that purpose, an architecture is proposed to this integration that includes the wheelchair architecture, containing the hardware used and also the software based on Robot Operating System (ROS) nodes. Next the integration with Sistema Integrado de Informação Hospitalar (SONHO) is presented, being described the necessary communication with the system with example messages to show the structure of Health Level 7 (HL7). Lastly, as the wheelchair needs to travel between floors, there will be an integration with the elevators so that the wheelchair is also autonomous in this process.

II. Related Works

Intelligent Wheelchairs (IW) are locomotion devices used to assist users with some kind of physical disability, where artificial control systems augment the user capabilities, reducing or eliminating the need for the user to drive [1]. They are usually controlled by a computer, apply several algorithms to derive meaningful information from sensors and act according to the user goals and the conditions of the environment. According to the literature, the main functions of an IW include [2], [3]:

- Multiple forms of interaction with the user (i.e. voice, hand, facial and head gestures);
- Autonomous navigation in dynamic environments;
- Communication with other devices (i.e. elevators, service robots, other wheelchairs).

Since the 80s a significant number of research and development projects tackled topics related to intelligent wheelchairs. Two quite comprehensive literature reviews were presented by Simpson and Richard [1], and Faria et al [4].

One of the first IW projects was proposed by Madarasz et al in 1986 [5]. In 1994, Wellman et al [6] proposed a wheelchair equipped with two legs allowing for it to climb stairs and move on uneven terrain. Next year, Miller et al [7] developed Tin Man that contained three operating ways: human guided with

This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project UIDB/50014/2020

2020 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
April 15-16, Ponta Delgada, Portugal

obstacle avoidance, move forward along a heading and move to a point (*x,y*). Later, due to the need to decrease dependence on contact sensors, modify the user interface according to the needs of the community and increase the operating speed, the system evolved to Tin Man II.

The prototype NavChair was developed at a later time, in 1994 [8], [9] based on an electric power wheelchair. The machine shared control with the user, it had obstacle avoidance and modified the user's voice input command to achieve safe travel.

In 2007, Philips et al [10] proposed a control that worked according to the user's need: the better the user dopes alone less assistance he received. Basically, the user had full control of the wheelchair until he requested help. User input was collected through a brain-computer interface.

In 2008, Gao et al [11] placed a Light Detection And Ranging (LiDAR) sensor in a lift platform and evaluated the performance of the sensor for docking an IW system equipped with retro-reflectors.

As most of wheelchair projects applied complex hardware and software architectures, the IntellWheels project [12] was created with the focus to develop an IW with a flexible and multimodal interface whose integration in wheelchairs available on the market could be done with few modifications with a good target price [13].

The Autonomous Vehicle team of an alliance between a research group of National University of Singapore (NUS) and the Massachusetts Institute of Technology (MIT) tested a self-driving wheelchair in Change General Hospital in 2016 [14]. They used the same technology that they developed for autonomous driving and all the user had to do was point where he wants to go and the wheelchair would drive him autonomously. The vehicle used three LiDAR to detect obstacles and to navigate. However, this sensor is expensive and using three of them in a wheelchair results in a very expensive prototype.

In 2017, a Wheelchair Mounted Robotic Arms (WMRA) was developed with the objective to make it autonomous, both at the level of localization and at object manipulation [15].

III. Proposed Architecture

The Connected Driverless Wheelchair (CDW), which is being developed in the scope of this project, shares most of the design principles established in our previous works, such as our concern with the user, with the interference in the wheelchair use, and with the visual impact [12], [13]. The base platform is the Evolution powered wheelchair from Vassilli. It is chosen for its low cost as well as its compact and foldable form factor, such that it is able to navigate through crowded areas and around tight corners easily. The system is retrofitted with the necessary sensors, actuators, power systems and computing units to perform the autonomous functions safely and comfortably. A compact Next Unit of Computing (NUC) computer runs Ubuntu 16.04 with Kinetic release ROS [16]. The computer is fitted with an Intel Core i5-4210U CPU running at 1.70 GHz, 3.0 GB RAM, and 256 GB SSD. The computer is connected to the internet via WiFi connection for communication with other systems. Two encoders (OMROM E6A2-CW5C 100P/R), coupled directly in the motor shafts, detect the rotation of the wheelchair's powered wheels. A Roboteq SDC2130 motor driver, with a 32-bit microprocessor, runs the algorithms used to control wheel speeds in closed loop. Environmental sensing is achieved through two Hokuyo UST-10LX laser scanners. LiDAR data is fused with odometry to perform mapping, localization, and obstacle avoidance. A Raspberry Pi-based Human-Machine Interface displays real-time information and allows local interaction in a unified and intuitive user interface. Power for the computer, sensors and motor driver is drawn from the wheelchair batteries though a voltage regulator. The hardware architecture and its connections are presented in Fig. 1.

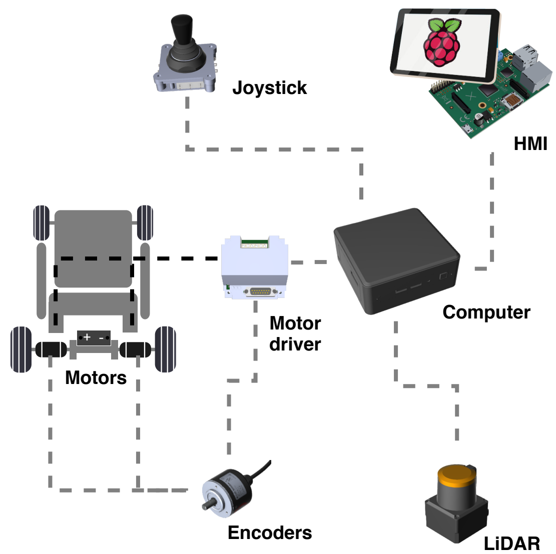

Fig. 1. Proposed hardware architecture

The software architecture of the prototype is based on ROS and depicted on Fig. 2. All ROS nodes were developed in C++ and run in the wheelchair's compact computer. The core subsystems of the vehicle can be broadly categorized into four categories: hardware abstraction, navigation, decision, and system interface.

Hardware abstraction comprises all ROS packages used to interact with the robot's hardware. The essential hardware the robot needs to interact with are LiDAR, motor controller, and joystick. This peripheral allows controlling the robot devices at an early stage of the process.

Navigation refers to the robot's ability to represent the environment, determine its own pose and execute a trajectory towards its goal. Mapping is performed with GMapping [17], a Rao-Blackwellized particle filter to create grid maps from LiDAR data. In its current implementation mapping consists of two steps performed one time for each desired environment. First the wheelchair is driven manually around the environ-

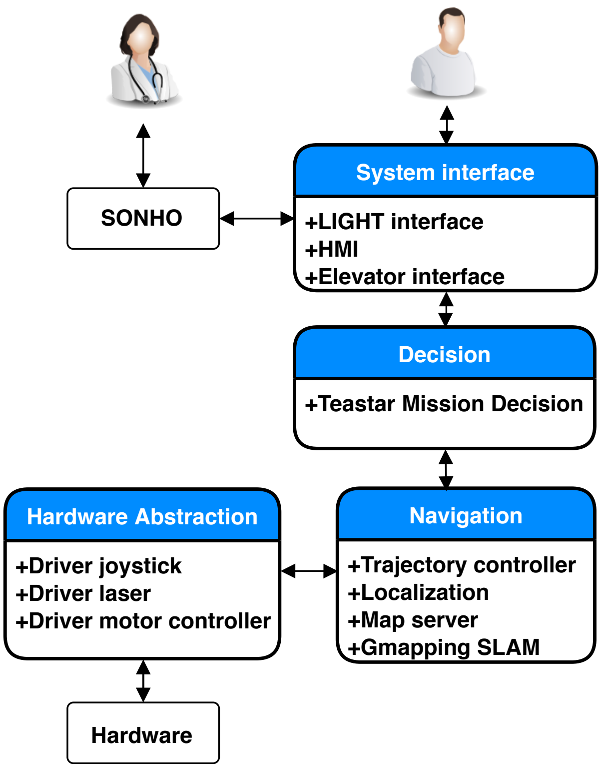

Fig. 2. Proposed software architecture

ment while recording the laser scan and odometry data of the trip. Next, GMapping processes the raw sensor data and creates a 2D occupancy grid map. The map is then made available to the localization system trough a map server. The implemented localization mainly consists of a map-matching algorithm referred as Perfect Match (PM) [18]. To estimate the position of the robot, PM minimizes the matching error between the data acquired by the LiDAR and the 2D environment map stored in the robot's database. A Kalman Filter is used to merge odometric data provided by wheel's encoders with those from LiDAR sensors, providing a more robust pose estimation [19]. The trajectory controller receives information about the trajectory from the mission control algorithm and outputs linear and angular references to the motor controller.

Decision consists on the wheelchair's ability to determine a trajectory it has to perform to transport a given patient to the operating room. Predetermined paths are stored in a directed graph, allowing a high degree of motion predictability and repeatability. In the present scenario those are important features considering that the wheelchair will share the environment with humans. Graphs are constituted by vertexes (possible stopping poses) and edges (parametric curves that connect two vertexes). Each edge has its own linear velocity associated, which is constant when the robot is moving along that specific path. On the other hand, each vertex may have some actions associated with it, such as exchange map or interact with elevator. The specific trajectory performed by the robot for each objective is calculated and optimized by a modified routing A* algorithm that takes into account the speed of the obstacles [20].

The system interface allows information flow between the wheelchair and external systems. The Humam-Machine Interface displays information to identify the user it is transporting, its origin and destination. It also displays virtual buttons to allow the health care team start, cancel and finish the trip. The other two interfaces (LIGHt interface and Elevator interface) allow communication with the hospital information system and the elevators, and are further detailed in the next Subsections.

A. Hospital Information System - SONHO

SONHO is an information system developed and managed by the Portuguese governmental organization Serviços Partilhados do Ministério da Saúde - SPMS (Health Ministry Shared Services). It was created in the 80s in a partnership between the Instituto de Engenharia de Sistemas e Computadores - INESC and the Serviço de Informática em Saúde - SIS (Health Informatics Service) to respond to the growing need to make administrative work in Portuguese hospitals more efficient. The system has been deployed in Portuguese public hospitals and health care units since the 90s, and currently is adopted in most public hospitals and health care units [21]. SONHO is the base system of the Portuguese public hospitals because it provides a great amount of information about the patients to other health information systems, such as patient number, name, age, contact details, clinical history, among others. Its second version (SONHO V2) was designed with seven vertical modules, to support each of the hospital activity areas, and one horizontal module, for system integration [22].

SONHO V2 adopts the Local Interoperability Gateway for Healthcare (LIGHt) middleware to exchange, integrate, share, and retrieve electronic health information with external systems. The LIGHt platform uses a publish/subscribe architecture and it is event-oriented. This architecture encourages the detachment of the different software layers, maximizing the reuse of the code in the several modules of the project. Communication is based on the (HL7) standard, which defines how information is packaged and communicated from one party to another [23].

LIGHt has an architecture divided in four modules: Mirth Connect Services (MCS), responsible for integrating HL7 v2.5 and HL7 FHIR; SPMS Event Controller (SEC), which allows event-oriented architecture; Web Services (WS), which has a web services layer for cases on which systems choose to implement a web interface; and the Backoffice (BO), which provides a graphical interface that facilitates the management and configuration of LIGHt.

In this integration the system interacts with the MCS, which processes HL7 messages through a set of channels, where it is expected to receive messages via Minimum Lower Layer Protocol (MLLP). The workflow of these messages can be asynchronous or synchronous.

In the asynchronous workflows, the routing channel, according to the LIGHt configuration, places the message in the correct queue of the processing channel. In this phase, the

```
MSH|^~\&|HL7_DEFAULT|INST|SONHO|INST|20150812100000||QBP^ZI1^QBP_Q11|2cacfeba-f795-
4d79-919d-c03779fe05b1|D|2.5
QPD|ZI1^Find Inpatient||@InpatientID^14000443|
RCP|I|5^RD
DSC|1 |
```


Fig. 3. Inpatient search [23].

Message Header Segment (MHS) is also checked and if an error is detected, the message processing ends and a NACK is sent as response. These message queues are processed in order of arrival, which creates some problems since the messages will have to be processed one at a time. To reduce this problem, the architecture allows the existence of several subscribers for the same type of event.

In the synchronous workflows, the messages received are processed directly by the channel that receives them. In these situations it is necessary to wait for the end of the processing to receive the respective acknowledgments. They are normally used in the searching process.

B. Integration with the Hospital Information System

In this project, patients will be transported from the inpatient service to the operating room. The beginning of this process is generated by a request to pick up a specific patient who is identified by his process number. This order also includes the room where the patient should be taken to.

Upon receiving the request, the CDW will request the system for the patient's location and it will go to the patient's infirmary when it receives an answer. Fig. 3 shows an example of an internment search by the inpatient number.

The messages are divided into segments and each segment has an header. The first segment is Message Header Segment (MSH) and consists of several fields separated by a vertical bar. The MSH-9 is the message type, that is divided into components separated by "^". The first component, "QBP", is the message code, the second, "ZI1", corresponds to the HL7 event code and the last one, "QBP_Q11", is the message structure. So, by analysing this field it is possible to determine the sort of message being exchanged. The second segment, Query Parameter Definition (QPD), is composed by several fields, namely the query tag, "Find Inpatient", and the user parameters, in this case the patient ID. There is also the Response Control Parameter Segment (RCP) field that can be used to define a search priority and to limit the number of returned results.

The answer to this search is shown in Fig 4, where it is also possible to see several segments, namely the Patient Identification Segment (PID), that contains some personal data about the patient, and the most important segment taking into account this project objective, the Patient Visit Segment (PV1), where it is possible to find all the information that the CDW needs: the room code 4012, bed 5253, building 1 and floor 4.

After arriving to its destination infirmary, the wheelchair will have to wait for either an assistant or a nurse to sit the patient on it. The order to proceed with the patient to the operating room is given by the same assistant or nurse, through


```
MSH|^~\&|SONHO|INST|HL7_DEFAULT|INST|20151218121747||RSP^ZI1^RSP_ZI1|4387b161-
a4aa-4e3b-a2c3-f15ec1c1d047|D|2.5
MSA|AA|0cacfeba-f795-4d79-919d-c03779fe05b9 QAK||OK||1|ZI1^Find Inpatient||1|0
QPD|ZI1^Find Inpatient||@InpatientID^14000443
PID|1||174846^^^SONHO^NS~360980071^^^RNU^SNS||GOMES
DUARTE^RICARDO^^^^^L||19720807000000|M|||RUA DO ALGAR LOTE 4-
2,DTO^^CARTAXO^140000^2070-
0^PT^PORTUGUESA^140600^140601|||||||99234960^^^SONHO
PV1|1|INT|30011^4012^MAC^5253^^Maca^1^4^SALA
3^^SONHO|P||30015^7014^MAC^958^^Maca^1^7^SALA
5^^SONHO|||30011||||5|||||14000443^^^SONHO||||||||||||||||||||||||20150915123219|||||||V
IN1|1|935601^SNS^SONHO|||||||||||||||||||||||||||||||||||||||||||||||360980071^^^SONHO |
```


Fig. 4. Response to an inpatient search [23].

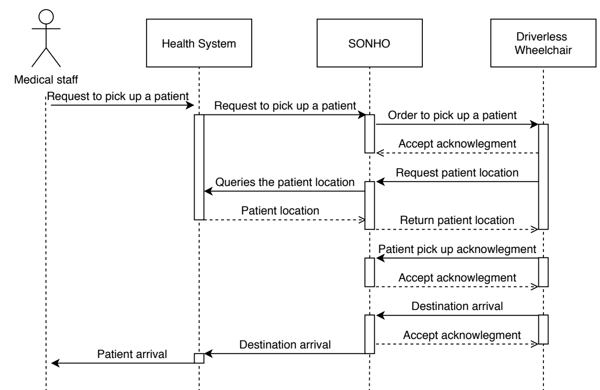

Fig. 5. Sequence diagram of communication with SONHO

the touchscreen. Given this order, the CDW notifies the system that the patient will be transported. If, by any chance, it is no longer necessary to transport the patient, it is possible to cancel the process, also via the same touchscreen, and the wheelchair returns to its starting point.

When the CDW gets to the door of the operating room, it sends a message to notify the system of its arrival and waits for someone to help remove the patient and conclude the procedure. After this, the wheelchair checks if there is an order waiting, in which case it goes get the patient in question; otherwise it goes to its starting point.

At any point in the process, a cancellation order may be given by the operating room. If the wheelchair has not yet picked up a patient, it simply ignores the request; if it is already carrying a patient, it takes him back to his bed in the inpatient area. A sequence diagram of all this process is shown in Fig. 5.

C. Integration with the Elevators

Considering that the inpatient service and the operating rooms are often located at different floors of the hospital building, the CDW is required to communicate with the elevators. While modern Programmable Logic Controller (PLC) support network communication, most of the PLC currently used in the elevator control systems of the Centro Hospitalar Universitário do Porto - Hospital de Santo António

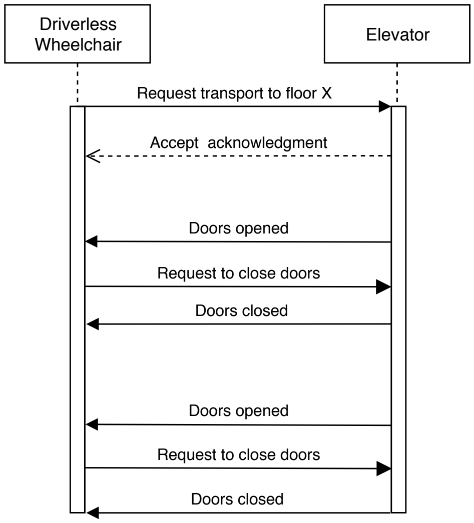

Fig. 6. Communication protocol between the wheelchair and the elevators

(CHPORTO) do not support this communication interface. Therefore, the communication between the wheelchair and the elevators will be implemented trough an IoT Ethernet I/O Module (ADAVANTECH WISE-4060). This module has four Digital Input and four Relay Output channels, and supports several communication protocols (Modbus/TCP, TCP/IP, UDP, DHCP, HTTP, MQTT). The module will be installed close to the PLC, and wired to its digital I/O. To exchange data with the CDW, the module will be connected to the hospital network trough its Ethernet port. A ROS-MQTT bridge will serialize messages from ROS to MQTT, and deserialized messages from MQTT to ROS topics.

Since there is no standard protocol to support the elevator-CDW interaction, we have partnered with CHPORTO and the company providing maintenance to their elevators to design the protocol depicted in Fig 6:

1) The CDW requests transport from an origin to a destination floor;
2) The elevator acknowledges the request;
3) The elevator informs that it is parked on the origin floor and it's doors are opened;
4) The CDW informs that it is inside the elevator cabin and is ready to start the trip;
5) The elevator informs that it has closed it's doors, and will start the trip;
6) The elevator informs that it is parked on the destination floor and it's doors are opened;
7) The CDW informs that it is outside the elevator cabin;
8) The elevator informs that it has closed it's doors and will proceed regular operation.

IV. Conclusions

In this paper was presented the system architecture of a driverless wheelchair, named Connected Driverless Wheelchair (CDW) in the scope of this project. The wheelchair was designed with robustness, reliability, and safety in mind, and is able to provide on-demand transportation of patients in hospitals - presently a case test is being developed at the Centro Hospitalar Universitário do Porto - Hospital de Santo António (CHPORTO). In order to enable the wheelchair to perform its requested tasks, its software and hardware architectures are described, as well as the approaches that will be followed to integrate it with elevators and with the hospital information system. In the future it is intended to deploy and test the wheelchair in the hospital, as well as test it over longer periods of time. It is also intended to conduct two inquiries: one to users of the system to see how transport can be made more pleasant, to understand the level of stress that certain people may have while being transported by an autonomous system and how to avoid it, and another for people who help with transport in order to understand whether they prefer to do this type of work or other tasks.

References

[1] R. C. Simpson, "Smart wheelchairs: A literature review," *Journal of rehabilitation research and development*, vol. 42, no. 4, p. 423, 2005.
[2] R. A. M. Braga, M. Petry, A. P. Moreira, and L. P. Reis, "Concept and design of the intellwheels development platform for intelligent wheelchairs," in *Informatics in Control, Automation and Robotics*, ser. LNEE, J. A. Cetto, J.-L. Ferrier, and J. Filipe, Eds. Heidelberg: Springer-Verlag, 2009, vol. 37, pp. 191–203.
[3] J. Gray, P. Jia, H. H. Hu, T. Lu, and K. Yuan, "Head gesture recognition for hands-free control of an intelligent wheelchair," *Industrial Robot: An International Journal*, 2007.
[4] B. M. Faria, L. P. Reis, and N. Lau, "A survey on intelligent wheelchair prototypes and simulators," in *New Perspectives in Information Systems and Technologies, Volume 1*. Springer, 2014, pp. 545–557.
[5] R. L. Madarasz, L. C. Heiny, R. F. Cromp, and N. M. Mazur, "The Design of an Autonomous Vehicle for the Disabled," *IEEE J. Robot. Autom.*, vol. 2, no. 3, pp. 117–126, 1986.
[6] P. Wellman, V. Krovi, and V. Kumar, "An adaptive mobility system for the disabled," *Proc. - IEEE Int. Conf. Robot. Autom.*, no. pt 3, pp. 2006–2011, 1994.
[7] D. P. Miller and M. G. Slack, "Design and testing of a low-cost robotic wheelchair prototype," *Auton. Robots*, vol. 2, no. 1, pp. 77–88, 1995.
[8] D. A. Bell, J. Borenstein, S. P. Levine, Y. Koren, and L. Jaros, "Assistive navigation system for wheelchairs based upon mobile robot obstacle avoidance," *Proc. - IEEE Int. Conf. Robot. Autom.*, no. pt 3, pp. 2018–2022, 1994.
[9] R. C. Simpson, S. P. Levine, D. A. Bell, L. A. Jaros, Y. Koren, and J. Borenstein, "NavChair: An assistive wheelchair navigation system with automatic adaptation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1458, no. September 2017, pp. 235–255, 1998.
[10] J. Philips, J. D. R. Millán, G. Vanacker, E. Lew, F. Galán, P. W. Ferrez, H. Van Brussel, and M. Nuttin, "Adaptive shared control of a brain-actuated simulated wheelchair," *2007 IEEE 10th Int. Conf. Rehabil. Robot. ICORR'07*, no. June 2014, pp. 408–414, 2007.
[11] C. Gao, I. Hoffman, T. Miller, T. Panzarella, and J. Spletzer, "Performance characterization of LIDAR based localization for docking a smart wheelchair system," *Int. Conf. Intell. Robot. Syst. San Diego*, no. July, pp. 4–6, 2008.
[12] R. A. M. Braga, M. P. BEng, L. P. Reis, and A. P. Moreira, "Intellwheels: Modular development platform for intelligent wheelchairs." *Journal of Rehabilitation Research & Development*, vol. 48, no. 9, 2011.
[13] M. R. Petry, A. P. Moreira, B. M. Faria, and L. P. Reis, "Intellwheels: Intelligent wheelchair with International Conference on e-Health Networking, Applications and Services (Healthcom 2013). IEEE, 2013, pp. 414–418.

[14] *SMART FM trials self-driving wheelchair*, Singapore-MIT Alliance for Research and Technology, 2017, accessed on 27/01/2020. [Online]. Available: https://smart.mit.edu/news-events/news/smart-fm-trials-self-driving-wheelchair

[15] N. Shafii, P. Farias, I. Sousa, H. Sobreira, L. Reis, and A. Moreira, "Autonomous interactive object manipulation and navigation capabilities for an intelligent wheelchair," pp. 473–485, 08 2017.

[16] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3. Kobe, Japan, 2009, p. 5.

[17] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[18] H. Sobreira, M. Pinto, A. P. Moreira, P. G. Costa, and J. Lima, "Robust robot localization based on the perfect match algorithm," in *CONTROLO'2014–Proceedings of the 11th Portuguese Conference on Automatic Control*. Springer, 2015, pp. 607–616.

[19] M. Pinto, H. Sobreira, A. Moreira, H. Mendonça, and A. Matos, "Self-localisation of indoor mobile robots using multi-hypotheses and a matching algorithm," *Mechatronics*, vol. 23, pp. 727–737, 09 2013.

[20] T. Nascimento, P. Costa, P. Costa, A. Moreira, and A. Scolari Conceicao, "A set of novel modifications to improve algorithms from the a* family applied in mobile robotics," *Journal of the Brazilian Computer Society*, vol. 19, 06 2012.

[21] M. M. Cruz-Cunha, *Encyclopedia of E-health and Telemedicine*. IGI global, 2016.

[22] *SONHO v2 - Sistema Integrado de Gestão Hospitalar (v2)*, SPMS - Serviços Partilhados do Ministério da Saúde, 2015.

[23] *LIGHt - Documentação*, SPMS - Serviços Partilhados do Ministério da Saúde, 2017, accessed on 23/01/2020. [Online]. Available: https://spmspt.atlassian.net/wiki/spaces/PD/overview