


ROS Based Multi-sensor Navigation of Intelligent Wheelchair

View metadata, citation and similar papers at core.ac.uk

brought to you by  CORE

provided by Univ

School of Computer Science and Electronic Engineering, University of Essex

Colchester CO4 3SQ, United Kingdom

Email: {rlib, masgha, kdm, hhu} @essex.ac.uk

Abstract—Our society is moving towards an ageing society and the number of population with physical impairments and disabilities will increase dramatically. It is necessary to provide mobility support to these people so that they can live independently at home and integrated into the society. This paper presents a ROS (Robot Operating System) based multi-sensor navigation for an intelligent wheelchair that can help the elderly and disabled people. ROS provides an easy to use framework for rapid system development at a reduced cost. Some experimental results are given in the paper to demonstrate the feasibility and performance of the developed system.

Keywords—Robot operating system, Intelligent wheelchair, Multi-sensor navigation, Ageing society, well-being.

I. INTRODUCTION

The demographic change towards ageing is a global phenomena in modern society. In the EU demographic statistic report [1], it projects that the number of elderly people will quickly increase in 2008 – 2060. From 2015 on, deaths are projected to outnumber births in the EU27 and almost three times as many people will be 80 or more in 2060.

The growing number of elderly population is accompanied with rapid increasing number of people with age-related diseases and impairment. A vast of elderly individuals suffer from mobility decline and other chronic diseases. They have difficulty with the activities of daily life independently and they must rely on caregivers [2]. Wheelchairs are usually used to assist people with mobility impairment. However, many people struggle to drive the wheelchair, especially for those have shoulder stroke, cognitive disease and apoplexy. Consequently, to accommodate this population to live independently and reduce reliance on caregivers, the research communities have proposed and developed electric-powered wheelchairs with robotics technologies, namely intelligent wheelchair or robotic wheelchair. Intelligent wheelchairs are able to provide navigational assistance in domestic and outdoor environments, which allows the users to drive more efficiently and securely. One paradigm is an autonomous navigation enabled powered wheelchair automatically arrive the goal its user tell to.

The research on intelligent wheelchair emerged in 1980s, and a series of long term projects have been carried out have been carried out, such as SENARIO [3], Wheellesly [4], NavChair [5], and VAHM [6]. Most of these intelligent wheelchair projects share similar concept- shared control. User can interact with the wheelchair via HMI(Human-Machine Interface) devices such as voice control, gesture control, and

BCI-based control etc. The navigation behaviours are triggered by human commands. Users may just give a destination and let the wheelchair achieve the goal in an autonomous way. Recently years, a new approach, collaborative control of intelligent wheelchair has been proposed [7] [8] [9]. By this approach, wheelchair works in a reactive way rather than just execute the command from the user. It provides multiple modality to adapt different user profiles. Cooper [10] presents a prototype of personal mobility and manipulation appliance (PerMMA). PerMMA is a mobile robot base with full power wheelchair with a custom track system around the wheelchair, and it is equipped with two robotic manipulators. PerMMA co-robot systems that can work work in collaboration with the user.

Navigation is a major module of an intelligent wheelchair, which performs some type of perception, planning, reasoning and control. However, to get a robotic system up and and running in real world, a much larger complex software system needs to be developed. Unfortunately, there are few commercial intelligent wheelchairs in the market so far. The current intelligent wheelchairs are still unaffordable for widely use. A huge gap therefore exists between the research and the use in reality. In order to reduce the cost of intelligent wheelchair systems, we have to develop universal standard systems and hardware that are possible to be promoted for every one and everywhere.

An open source project named ROS (Robot Operating System) [11] is a robot operating system that provides various functions to be deployed into different robots. ROS integrates some state-of-the-art robot control technologies from the robotics research and development community. Nevertheless, it contains the reusable software stacks for everything to build a robot control system, from basic controller, models, to autonomous navigation functions. These reusable code and tools are based on various open source projects among the communities. For example, Mobile robot navigation module, mapping and localization planning algorithm module, motion controller is from PR2, Stage/Player [11], Rescue robot, Humanoid project etc. ¹ Vision module is from OpenCV and OpenNI. Voice module is supported by Sphinx, ROS Nao project contributes to the humanoid motion planning and navigation module. ROS allows developers to share knowledge and access to other capabilities in it. It is a successful "ecosystem" for robotic research and software development, which promote the latest contribution to the community and reduce

¹<http://www.ros.org/wiki/>

the cost of complex robot system development. ROS now has been being accepted and employed by many projects, such as [12] [13] [14] [15] [16] for their robotics research.

In this paper, the development of an ROS based control architecture for an intelligent wheelchair is presented. The rest of the paper is organized as follows. Section II introduces the system architecture that is designed for our intelligent wheelchair. Section III describes how ROS is implemented in an intelligent wheelchair. In Section IV, some experimental results are given to demonstrate the feasibility and performance of the ROS based control architecture being developed. Finally, a brief conclusion and future extension are discussed in Section V.

II. SYSTEM ARCHITECTURE

A. Electric Powered Wheelchair Hardware

For this project, we have equipped a commercial powered wheelchair with embedded computers, multiple sensors, control electronics, and wireless networks, such as optical odometry encoders, two Hokuyo laser scanners, 12 sonar sensors, Mongoose 9DF IMU, camera, microphone and GPS. As shown in (Fig.1), the embedded PC is connected to the sensors and motors, which runs a Linux operating system and a client-server control software. A server programme running at the embedded PC can publish sensor data and receive actuator commands to motors. Serial boot-loaders have been installed in the modules with the embedded PC to facilitate future development. On the high level control, a ROS based system is developed, which is a client. It provides autonomous navigation and interaction interface. With the HMI bridge, users can actuate the navigation by the interface according to their preference.

B. Wheelchair Navigation System

The infrastructure of the wheelchair control system is based on ROS. ROS provides a framework for large-scale complex robot system development. It consists of a number of process and hosts connected in a peer-to-peer topology. Each process is a node which communicates with each other by XML-RPC based message via topics/services as Fig. 2 shows. The module can be deployed into other robots as well as integrated with users own software by some configuration. In ROS, *launch* file, XML like scripts, is used to bring up the module together and to configure parameters of nodes to compose a control system.

For instance, in a ROS package, it receives sensor data through a communication node, sensor data is represented as sensor message. Once the sensor message and transformation(TF) message is published, map server can subscribe the message with listener and then generate map for navigation. If map is launched, the planner and action node utilize TF message and pose message to compute the trajectory for the actionlib, which is used for motion control. The action message consists of a sequence of twist message in ROS. Then the motor drive node subscribe the twist message and generates motor strength to the actuators to drive a robot.

Fig 3 indicates the architecture of the navigation system based on ROS. The system consist of two part, low level

controller and high lever controller. The low level controller is installed on a embedded PC and . It provides the interfaces to motor actuators and sensors. A server/client program is design for communication with high level controller through Ad-hoc network. The high level navigation system is based on ROS navigation stack and differential_drive package. A communication node is used to encapsulate the client program, subscribe sensor message and publish actuator command.

III. IMPLEMENTATION OF NAVIGATION

Here we describe the implementation of the navigation system we have developed for the wheelchair as Fig 3 displays. The implementation is delivered through out communication module, model and transformation configuration, motion control, navigation stack configuration.

1) *Communication*: As it is mentioned above, ROS topics/services broadcast task request and response to the request among nodes via messages. The server programme and client program enable the communication between an external PC and an embedded PC via wireless network. Unified data type and data structure of sensor information and actuator commands are defined inside. On the high level control platform, the client acts as a message bridge and is wrapped into the ROS navigation package as a *comm* node. Thus, the module in ROS can take the message from sensors and then process it to generate relevant actions.

2) *Wheelchair Model and Transformation*: ROS provides various of functional utilities for the robot development. URDF is based on XML language to produce robot models that can be displayed in the simulator *Gazebo* and the robot visualization GUI *Rviz*. A differential driving vehicle model is created by URDF corresponding to the wheelchair. Fig. 6 shows the model of a wheelchair in *Rviz* visualization. *tf* package allows users to specify multiple coordinate frames over time. *tf* maintains the relationship between coordinate and transform points between any different coordinate frames. The *tf* configuration is set within a *launch* file, including the *tf* of wheelchair base_link, laser and sonar. For the wheelchair, the range and orientation of SONARs and lasers is specified to the centre of rear axle. *tf* messages are subscribed and broadcast between the nodes overtime.

3) *Motion Control with differential_drive*: The *differential_drive* package provides motion controller which is independent to motor hard driver. It takes twist messages from navigation stack and broadcasts the twist messages as the strength(or velocity) of motor driver. Contemporaneously, it receives odometry message from the *comm* node and publishes it to navigation stack. Within the *differential_drive* package, a PID controller is implemented for wheels velocity control. The parameters of PID, K_p, K_i, K_d is set in a launch file similar as following.

```
<launch>
...
<node pkg="differential_drive"
  type="pid_velocity.py"
  name="rpid_velocity">
  <remap from="wheel" to="rwheel"/>
  <remap from="motor_cmd"
  to="rmotor_cmd"/>
```

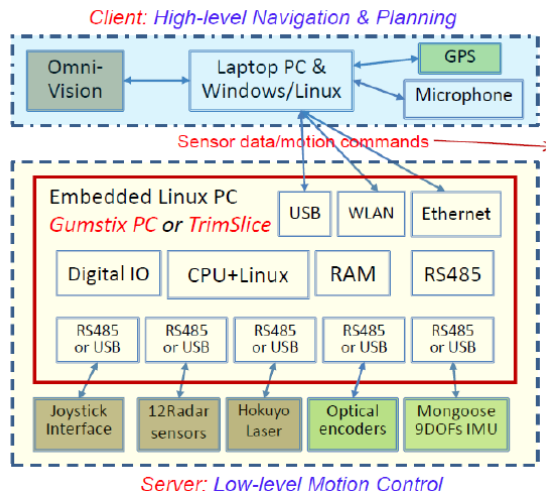


Fig. 1: Hardware of Essex Wheelchair

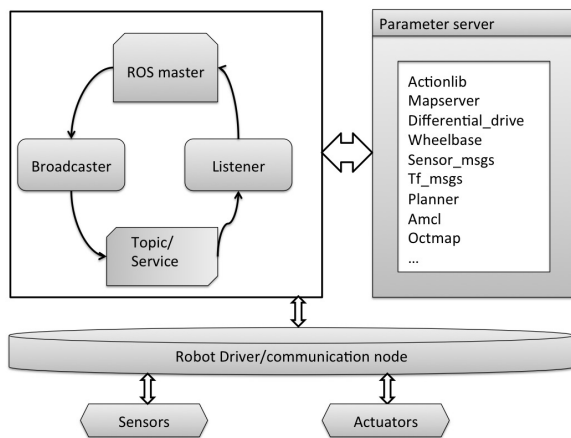


Fig. 2: ROS node communication

```

<remap from="wheel_vtarget"
to="rwheel_vtarget"/>
<remap from="wheel_vel"
to="rwheel_vel"/>
<rosparam param="Kp">40</rosparam>
<rosparam param="Ki">10</rosparam>
<rosparam param="Kd">0</rosparam>
...
</node>
</launch>

```

4) *Navigation stack configuration*: ROS contains a sequence of packages that process the information taken from sensors, odometry and a goal pose to generate proper motion commands to the a robot. *slam_gmapping* provides laser-based SLAM (Simultaneous Localization and Mapping) and can take in sensor data from the world, builds a 2D or 3D occupancy grid of the data for navigation. *map_server*

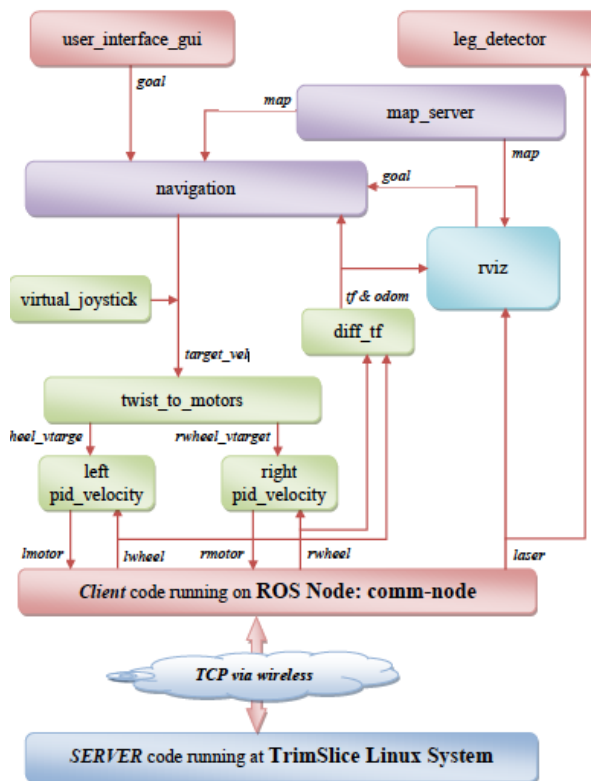


Fig. 3: Navigation system for a Electric Powered Wheelchair

is able to translate map data as ROS service and generate map file with sensor data. *amcl* provides a implement of the adaptive Monte Carlo localization approach to track robot in a known world map. *nav_core* provides global planner and local planner names *base_local_planner* and *base_local_planner*. *base_global_planner* uses *navfn* that wraps the Dijkstra's algorithm to compute a navigation path on a grid cost map.

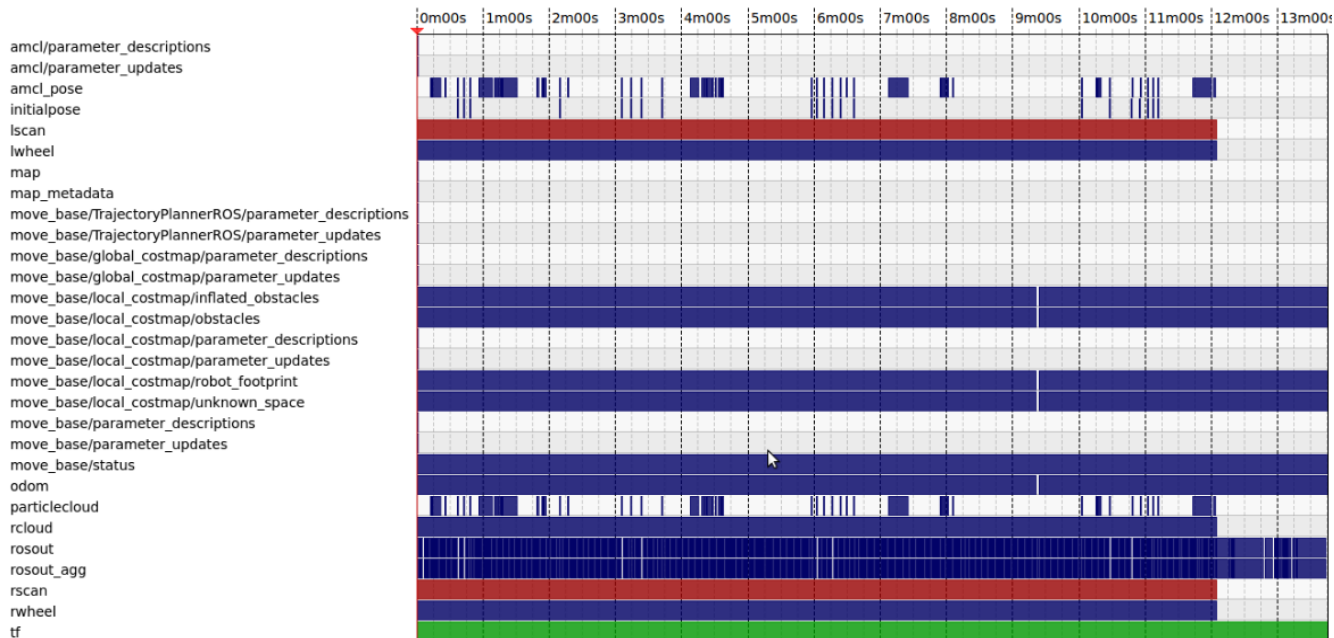


Fig. 4: ROS TFs and topics visualization

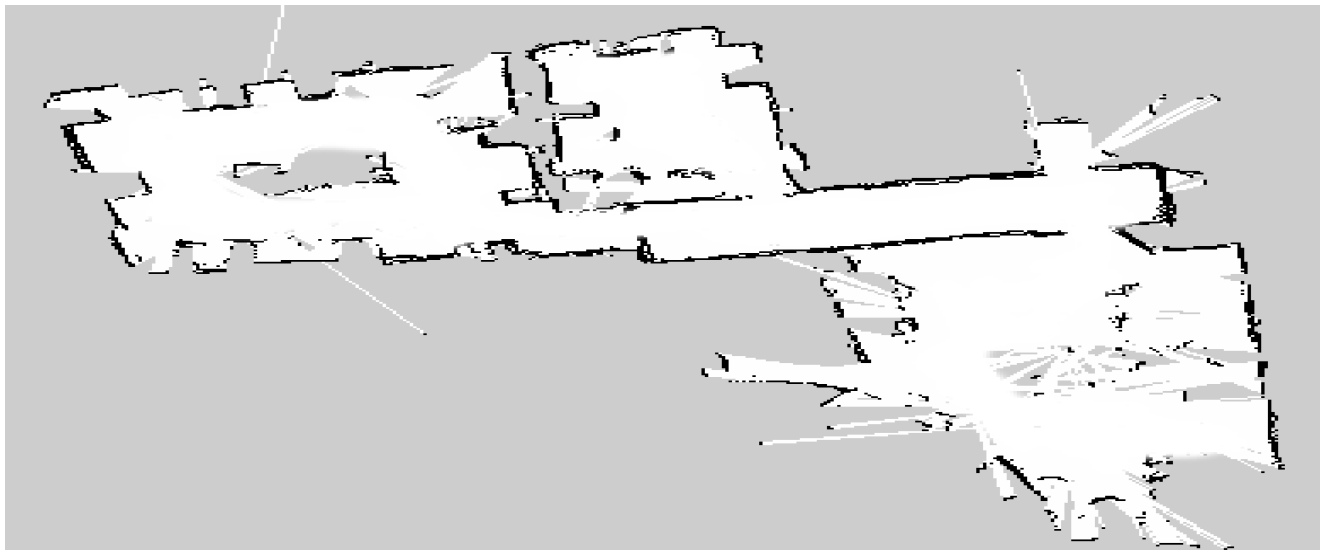


Fig. 5: Map of test environment built by laser data

Trajectory Rollout and Dynamic Window approaches are implemented as *base_local_planner*. It produces trajectory and motion commands as messages passing to a robot for path following within a given map.

In order to accomplish the navigation task, it is necessary to integrate the mapping, localization, planning together. ROS *move_base* plays such a role to link the global and planner together. It also maintains two cost maps, one for global planner and one for local planner. Since the navigation module in ROS lack of doorway passing capability for wheelchair, a Bézier Curve Trajectory approach is implemented [17]

for doorway passing navigation. Finally, to activate all the navigation components, a launch file is generate to specify the parameters and dependence, function of the nodes and packages that are employed for the navigation task. A simple GUI allows users to point a position as a goal for robot.

IV. EXPERIMENTAL RESULTS

This section presents the demonstration of the ROS based wheelchair autonomous navigation. The test was taken place real environment at our robot arena and robotics research office. Fig 4 displays the visualisation of each topic associated

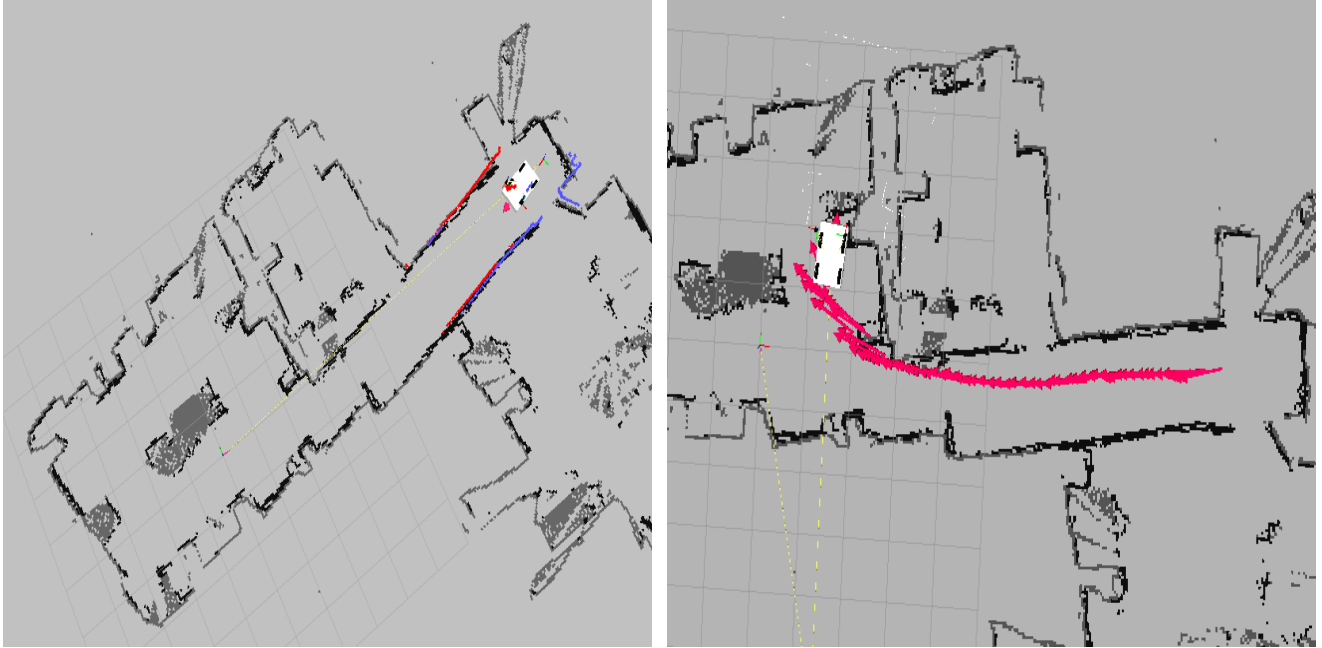


Fig. 6: Navigation with the grid map

to the navigation through time. A list of running nodes/services shows in the left side of the visualization.

A. Motion Control Test

In order to improve the accuracy of the wheelchair motion controller, We firstly tested the wheelchair by sending specific twist topic to the differential_drive which corresponds to the relevant motion command to the motor. Then we configured the parameters of PID in *launch* file until the wheelchair move correctly with regard to the twist message. In this experiment, the parameters, K_p , K_d and K_i of the PID controller are set to be 40, 10 and 0 respectively.

B. Mapping and Localization Test

Fig. 5 is the grid map of a clutter environment generated with laser scan. Laser data and odometry is recorded by exploring the environment. In order to get better map, it is necessary to drive a bit more after a close loop of the environment. The laser data and odometry is stored in rosbag which can be replay after exploring. The following two commands is used to process sensor data and produce a binary map image file. *map_server* and *slam_gmapping*. Run the following commands, it can produce a binary grey image map file.

- `$roslaunch slam_gmapping slam_gmapping scan:=base_scan _odom_frame:=odom`
- `$ roslaunch map_server map_saver`

C. Autonomous Navigation Test

Once the map and localization of the environment is known, a autonomous navigation can be achieved by the

planners and controllers. User can point on map to set a goal position, then the navigation schema is actuated. In Fig 6 the red arrow shows the origin position. The red trail is the visualization of the wheelchair trajectory in Rviz. The status of visualization model and real wheelchair are contemporary with regard to the map and environment.

V. CONCLUSION

In this paper, we proposed an ROS based intelligent wheelchair system for mobility impairment people. Users can interact with the robots and smart home via the HMI(Human-Machine Interface) devices. The system enables capabilities of autonomous navigation, inter-operation by multiple modality, interaction and collaboration with the wheelchair. A autonomous navigation for the wheelchair is implemented. It enable user point a goal and arrive the goal automatically in a static environment.

In future work, we will improve the navigation by more adaptive planner and wrap the doorway passing navigator into ROS. RFID can be beneficial to the accuracy of indoor navigation. Higher level planner with multiple-layered map(grid map and semantic map) will be implemented. We have already developed some HMI devices such as voice control, gesture control, EEG/EMG based control, for robots. To integrate the HMI devices for robot interaction, we design a bridge node in ROS to subscribe messages from the HMI devices and to deliver command to robot action. This bridge allows user to trigger the actuators for autonomous navigation of the robots via proper HMI to accomplish their missions. Therefore, it enable the robot to adapt to difference user profile and preferences. The HMI bridge is a client/server program that wraps the message data type and structure. To extend to adaptation of the usage of wheelchair for different user

profiles, more sophisticated collaborative control strategies will be applied for the reaction and interoperation between user and the intelligent wheelchair.

ACKNOWLEDGEMENTS

This research is financially supported by the EU COALAS project. The COALAS project has been selected in the context of the INTERREG IVA France (Channel) England European cross-border co-operation programme, which is co-financed by the ERDF. Our thanks also go to Robin Dowling for his technical support.

REFERENCES

- [1] "World population ageing: 1950-2050." [Online]. Available: http://ec.europa.eu/health/ph_information/dissemination/documents/com2009_380_en.pdf
- [2] United Nations, "World population ageing: 1950-2050." [Online]. Available: <http://www.un.org/esa/population/publications/worldageing19502050/>
- [3] N. Katevas and N. Sgouros, "The Autonomous Mobile Robot SENARIO: A Sensor-Aided Intelligent Navigation System for Powerd Wheelchair," *Robotics and Autonomous Magazine*, no. December, 1997.
- [4] H. Yanco, "Wheelesley: A robotic wheelchair system: Indoor navigation and user interface," *Assistive technology and artificial intelligence*, 1998.
- [5] R. Simpson, S. Levine, and D. Bell, "NavChair: an assistive wheelchair navigation system with automatic adaptation," in *Assistive Technology and AI*, 1998, pp. 235–255.
- [6] G. Bourhis and P. Pino, "Mobile robotics and mobility assistance for people with motor impairments: rational justification for the VAHM Project." *IEEE transactions on rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 4, no. 1, pp. 7–12, Mar. 1996.
- [7] C. Urdiales, M. Fernández-Carmona, J. M. Peula, U. Cortés, R. Annichiarico, C. Caltagirone, and F. Sandoval, "Wheelchair collaborative control for disabled users navigating indoors." *Artificial intelligence in medicine*, vol. 52, no. 3, pp. 177–91, Jul. 2011.
- [8] A. Escobedo, A. Spalanzani, and C. Laugier, "Experimental Setup for Human Aware Navigation," *Control Architectures of Robots*, 2012.
- [9] T. Carlson and Y. Demiris, "Collaborative control for a robotic wheelchair: evaluation of performance, attention, and workload." *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, vol. 42, no. 3, pp. 876–88, Jun. 2012.
- [10] R. Cooper, G. Grindle, and J. Vazquez, "Personal Mobility and Manipulation Appliance Design, Development, and Initial Testing," *Proceedings of the IEEE*, 2012.
- [11] M. Quigley and K. Conley, "ROS: an open-source Robot Operating System," in *In ICRA workshop on open source software*, no. Figure 1, 2009.
- [12] J. Mason and B. Marthi, "An object-based semantic world model for long-term change detection and semantic querying," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3851–3858, Oct. 2012.
- [13] R. Borja, J. de la Pinta, a. Álvarez, and J. Maestre, "Integration of service robots in the smart home by means of UPnP: A surveillance robot case study," *Robotics and Autonomous Systems*, vol. 61, pp. 153–160, Nov. 2012.
- [14] R. Arumugam, V. R. Enti, K. Baskaran, and a. S. Kumar, "DAVinCi: A cloud computing framework for service robots," *IEEE International Conference on Robotics and Automation*, pp. 3084–3089, May 2010.
- [15] J. Schulman, A. Lee, and I. Awwal, "Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization," 2013.
- [16] I. Lane, V. Prasad, G. Sinha, A. Umhuza, S. Luo, A. Chandrashekar, A. Raux, N. Ames, and M. Field, "HRI tk : The Human-Robot Interaction ToolKit Rapid Development of Speech-Centric Interactive Systems in ROS," *NAACL-HLT 2012 Workshop on Future directions and needs in the Spoken Dialog Community: Tools and Data*, pp. 41–44, 2012.
- [17] S. Wang, L. Chen, H. Hu, and K. McDonald-Maier, "Doorway Passing of an Intelligent Wheelchair by Dynamically Generating Bézier Curve Trajectory," in *IEEE International Conference on Robotics and Biomimetics*, 2012, pp. 1206–1211.