
Journal of Intelligent and Robotic Systems **34:** 301–314, 2002.

A Wheelchair Steered through Voice Commands and Assisted by a Reactive Fuzzy-Logic Controller ⋆

GABRIEL PIRES and URBANO NUNES
Institute of Systems and Robotics – Polo II, University of Coimbra, 3030 Coimbra, Portugal;
e-mail: {gpires, urbano}@isr.uc.pt

Abstract. This paper describes new results with a Reactive Shared-Control system that enables a semi-autonomous navigation of a wheelchair in unknown and dynamic environments. The purpose of the reactive shared controller is to assist wheelchair users providing an easier and safer navigation. It is designed as a fuzzy-logic controller and follows a behaviour-based architecture. The implemented behaviours are three: *intelligent obstacle avoidance*, *collision detection* and *contour following*. Intelligent obstacle avoidance blends user commands, from voice or joystick, with an obstacle avoidance behaviour. Therefore, the user and the vehicle share the control of the wheelchair. The reactive shared control was tested on the RobChair powered wheelchair prototype [6] equipped with a set of ranging sensors. Experimental results are presented demonstrating the effectiveness of the controller.

Key words: voice human–machine interface, shared control, fuzzy control, behaviour-based architecture.

1. Introduction

RobChair project aims to apply robotic algorithms on a powered wheelchair in order to improve mobility and safety. Users with severe motor handicaps such as tetraplegia and general muscle degeneration are unable to steer their own wheelchair through a conventional joystick, often depending on other persons. By endowing the wheelchair with new Human–Machine Interfaces (HMI) and increasing the wheelchair navigation autonomy [7], it is possible to contribute to the social independence of this group of wheelchair users.

In order to extend the RobChair accessibility, a new interface has been incorporated on the wheelchair: a voice HMI. The voice/speech is a natural form of communication and suits perfectly for users with severe motor limitations. However, this new interface does not solve completely the steering problem. Low-level voice commands are discrete and give rough direction information. In domestic/office environments usually with narrow spaces and dynamic obstacles it is very

⋆ This research was partially supported by FCT (Portuguese Science and Technology Foundation) under contract POSI/1999/SRI/33594. The first author would like to thank all the support given by the Polytechnic Institute of Tomar.

difficult, or even impossible, to drive a wheelchair with low-level voice commands. To make wheelchair voice-steering practicable, an assistive navigation module is being developed. The navigation module follows a hybrid architecture [5]. One part of the proposed architecture, fully implemented, is an assistive reactive controller, which turns feasible voice-steering. Three goals were achieved:

(1) the risk of collisions substantially decreased;
(2) the obstacles are safely and smoothly avoided taking into consideration the user intention, given by setting a rough direction; and
(3) a contour following task is performed autonomously.

This paper describes in detail the Reactive Shared-Controller (RSC), with particular emphasis on the fuzzy-logic implementation of the Intelligent Obstacle Avoidance (IOA). The paper is organised as follows. Section 2 presents the system architecture and sensorial system. Section 3 presents the RSC. The implementation of the fuzzy-logic IOA behaviour is described in Sections 4 and 5. Section 6 presents experimental results. Finally, Section 7 draws conclusions.

2. RobChair System Architecture and Sensorial System

2.1. SYSTEM ARCHITECTURE

Figure 1 illustrates the overall system architecture. Sensor data acquisition, joystick readings and motor commands are performed on a MC68332-based microcontroller. Sensor information is sent via a serial link to a laptop with Linux operating system. Here, there are three processes that inter-communicate through shared-memory: the process that handles the communications with the microcontroller; the process that runs navigation algorithms (RSC); and the Graphical User Interface (GUI) that allows to draw simple environment maps and to visualise, in real time, wheelchair movements and raw sensorial data (see Figure 2). Taking advantage of the Xwindow system (X protocol), the GUI can be displayed on any workstation of the network. Voice HMI is based on a Dragon^{TM} speech recognition software [2]. A small set of low-level commands providing direction and speed information is used to steer the wheelchair.

2.2. SENSORIAL SYSTEM

RobChair is equipped with 14 range triangulation infrared sensors (GP2D12 IR Sharp distance measuring sensors), 14 reflective On/Off IR sensors (Sunx), a ring of 7 Polaroid sonars and a front bumper as shown in Figure 3. The IR sensors are strategically positioned to cover all front and side regions. IR sensors have a high directionality (in comparison to Polaroid sonars), which is, on one hand advantageous due to the small angular uncertainty, but on the other hand, it is disadvantageous, because thin objects such as table legs are often non-detected. The range of the IR sensors we are using is also quite short. They provide reliable measures for distances less than 80 cm. To improve local obstacle detection, sonars

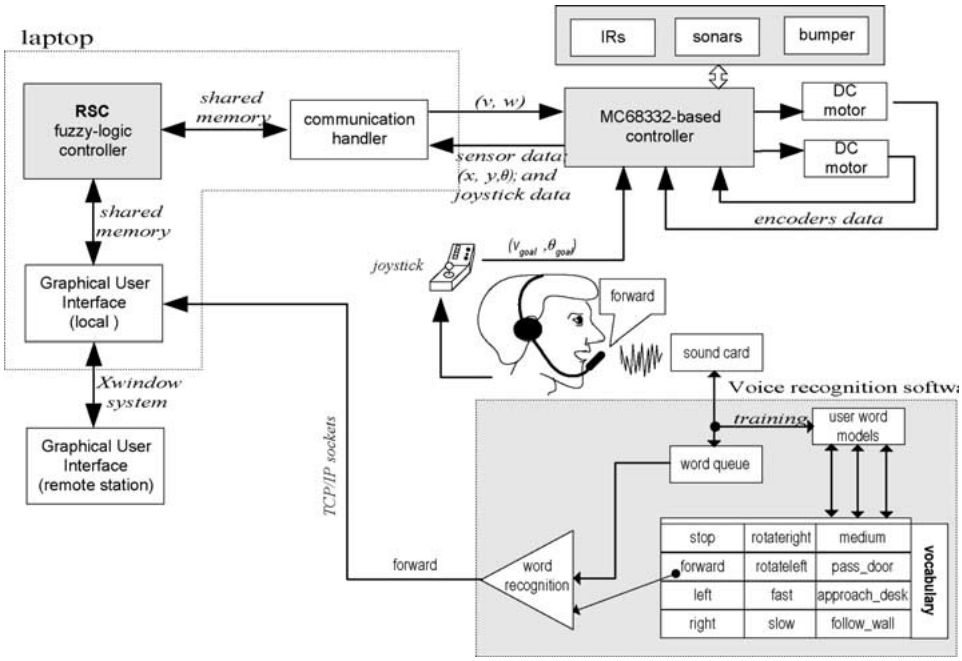


Figure 1. Overall software and hardware architecture.

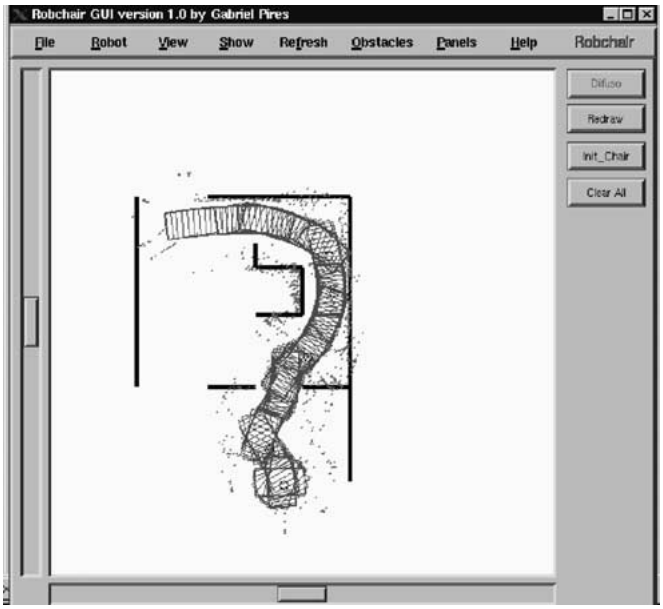
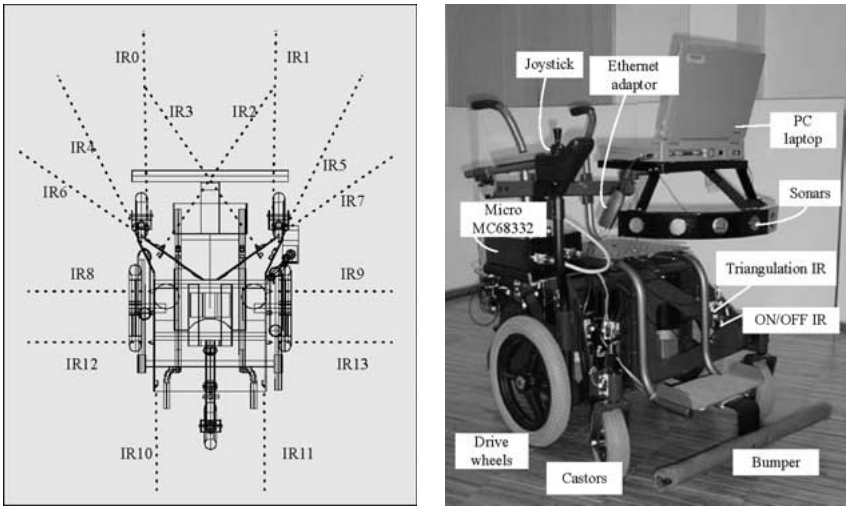


Figure 2. Graphical User Interface displaying a real lab experiment.



(a) (b)

(c)

Figure 3. (a) IR Sensor layout (top view). Each IR_i denotes a pair of an analog IR and a On/Off IR sensor; (b) RobChair wheelchair picture; (c) 3D model with IR beams.

were included. They have a much higher range and can be used to build local maps [1]. However, for the time being, the implemented RSC controller just uses the IR sensors. Navigation relies only on actual raw sensor measures, which return obstacle information on the vicinity of the wheelchair. The IR sensors are grouped in four detection regions (see Figures 3(a) and (c)):

1. IR0, IR1 and IR2, IR3: used to detect front obstacles. These sensors are the most important to perform obstacle avoidance manoeuvres;

2. IR4, IR6 and IR5, IR7: used to detect obstacles localised lightly to the left and right, respectively;
3. IR8, IR12 and IR9, IR13: besides being used to detect lateral obstacles, they are also used to follow autonomously long surfaces (contour following behaviour);
4. IR10 and IR11: used to detect back obstacles.

3. RSC Architecture

The RSC controller follows a behaviour-based architecture (Figure 4). It is composed by three behaviours: IOA, collision detection and contour following. IOA combines guidance information (system goals) with obstacle avoidance manoeuvres. Here, guidance information (direction and speed) comes directly from the user; however, it could come from a local trajectory planner. These behaviours are briefly described in the following.

IOA – this behaviour aims to relieve user's steering. It has to safely avoid static or dynamic obstacles, following at the same time, rough voice commands (see vocabulary in Figure 1). Door-passage and desk-docking are successfully accomplished in certain approach conditions.

Collision Detection – this is a simple behaviour that uses front bumper information to detect front collisions, and IR sensors information to detect potential collisions. When a front collision occurs or a front potential collision is detected, e.g., in a reduced space to perform a manoeuvre, only backward movements are allowed.

Contour Following – this behaviour is used to follow, at a fixed distance, long surfaces, e.g., long corridors, without intervention of the user. It uses distance and angle error information, in respect to the surface to follow, obtained from actual sensory data.

Guidance commands and obstacle avoidance behaviour are fused in the rule base of the fuzzy-logic controller. A decision-making module decides which of the three behaviours is currently active. In the present stage of development of the overall architecture, the decision-making module consists basically of an arbitrator. By default, IOA is active. If collision detection activates, it overrides the output of IOA or contour following behaviours. Contour following is explicitly activated from a user command (*follow_wall* command).

4. Control Behaviours

IOA and contour following are fuzzy-logic controllers. The choice of fuzzy-logic control relies on its versatility that makes it suitable for mobile robotic applications, such as reactive navigation. It does not require heavy processing and therefore

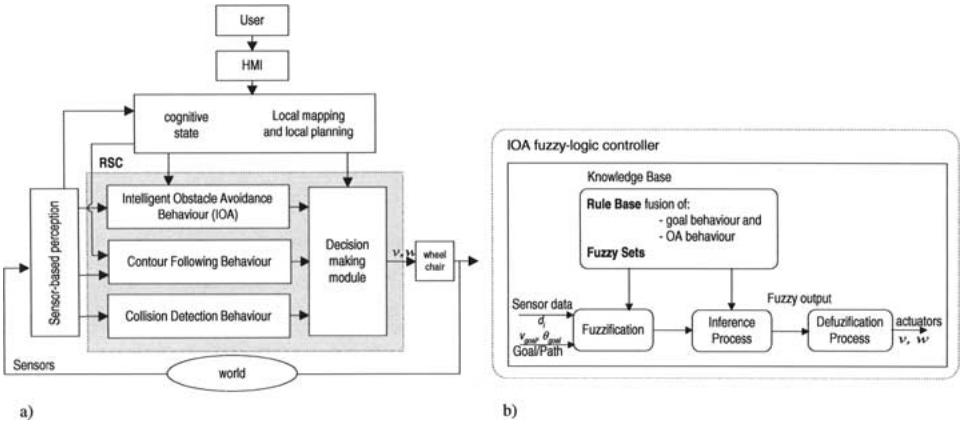


Figure 4. (a) RSC controller; (b) IOA fuzzy-logic controller.

allows real-time execution; it dispenses a mathematical model of the system to control; navigation algorithms are written through heuristic rules based on users experience; multiple-input/multiple-output (MIMO) controllers are easily designed; and finally, it is well suited to implement the fusion of behaviours and therefore appropriate to implement a shared-control system. The following sections describe the structure of the fuzzy logic controllers (as an example see in Figure 4(b) the IOA fuzzy-logic controller diagram).

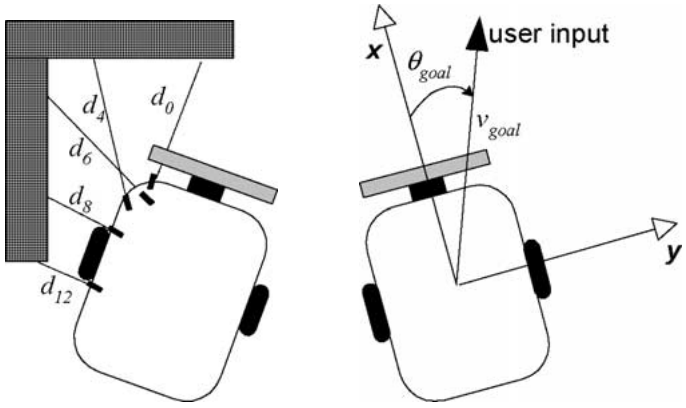
4.1. CONTROLLER INPUT/OUTPUT

The linear velocity v, and angular velocity w, are the output variables of the RSC controller. There are two types of input variables: sensor data and guidance information. There are fourteen sensor range variables (d_i, $i = 0, \ldots, 13$) each one corresponding to an analog IR sensor (Figure 5(a)), and two guidance variables: direction and speed. Sensor variables are used in different behaviours: IR0, ..., IR7 are used in IOA; IR8, 12, 9, 13 are used in contour following; and two bumper variables are used in collision-detection behaviour. The input guidance variables are the following: angular direction (θ_{goal}) and speed (v_{goal}) for joystick control; and θ_{goal} for voice control (Figure 5(b)).

4.2. FUZZIFICATION PROCESS

The fuzzification module transforms numerical variables in fuzzy sets, which can be manipulated by the controller (Figure 6). The linguist terms are presented in Table I. IOA controller uses fuzzy singletons to encode:

1. *voice command variables:* voice commands are provided at irregular time periods and give rough direction information;



(a) (b)

Figure 5. Input variables: (a) sensor variables; and (b) goal variables (guidance).

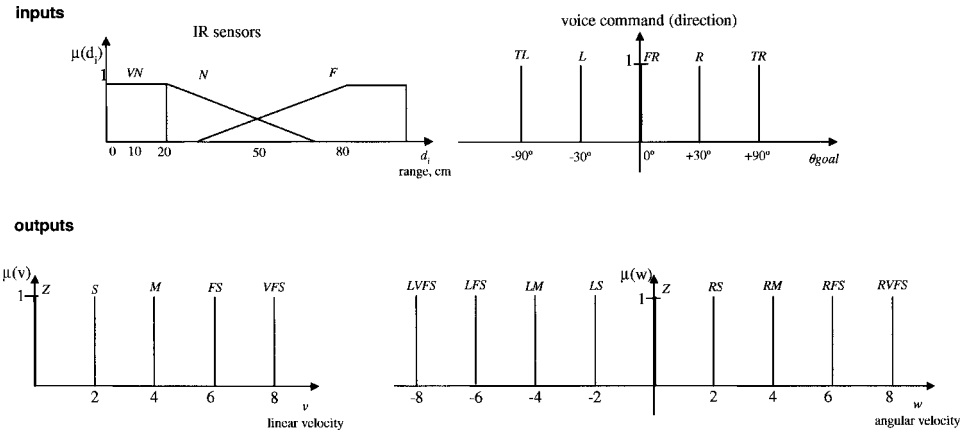


Figure 6. Fuzzy set variables.

2. *controller outputs variables:* linear and angular velocity variables are also modelled as discrete fuzzy sets. This simplified approach is used to increase the computation speed of the inference and defuzzification processes.

The other membership functions are:

Shoulder – used to encode analog infrared variables. The sets represent the possibility to find an obstacle at a given distance. Due to the short range of IR sensors, only Far and Near fuzzy sets were considered.

Rectangular – that is not a fuzzy set but instead a classical set. It aims to create a distance range for which the wheelchair movements are not allowed (Very Near fuzzy set).

Table I. Linguistic terms

VN	= Very Near
N	= Near
F	= Far
TL	= TurnLeft
L	= Left
FR	= Forward
R	= Right
TR	= TurnRight
Z	= Zero
S	= Slow
M	= Medium
FS	= Fast
VFS	= VeryFast
$LVFS$	= LeftVeryFast
LFS	= LeftFast
LM	= LeftMedium
LS	= LeftSlow
Z	= Zero
RS	= RightSlow
RM	= RightMedium
RFS	= RightFast
$RVFS$	= RightVeryFast

4.3. INFERENCE PROCESS

The inference process defines the connective, implication and rule combination operators. The controller uses *min* and *max* connectives and a singleton *sum-product* inference mechanism. This choice relies on two reasons. First, the *product* preserves the shape of the output fuzzy set⋆, and second, with the *sum* the result is influenced by different rules reaching the same conclusion [4]. The inference process is performed in three steps (see Table II and Figure 7).

4.4. DEFUZZIFICATION PROCESS

Two output fuzzy sets result from the inference process, one for linear velocity and the other for angular velocity. The defuzzification module obtains, from these sets, numerical values to send to the actuators. It was chosen the Centre of Gravity method (COG) because it takes into account, better than any other method, the dis-

⋆ It is however irrelevant in this case because output fuzzy sets are singletons, thus *min* operator could be used

Table II. Sum-product inference

1.	Computation of the activation degree for each rule l, β_l $(l = 1, \ldots, N)$ given by: $\beta_l = \min(\mu_{Ai}(d_1), \ldots, \mu_{Ai}(d_k))$, A_i are the input fuzzy sets (defining the linguistic terms) and d_j $(j = 1, \ldots, k)$ are the input variables.
2.	Computation of the output fuzzy sets B'_l, for each rule l, using *product* operation: $\mu_{B'_l}(v) = \beta_l \mu_{B_l}(v)$, B_l are the output fuzzy sets and v is the output variable.
3.	Combination of the output fuzzy sets B'_l, into a single fuzzy set B' using the *sum* operation: $\mu_{B'}(v) = \mu_{B'_l}(v) + \cdots + \mu_{B'_N}(v)$.

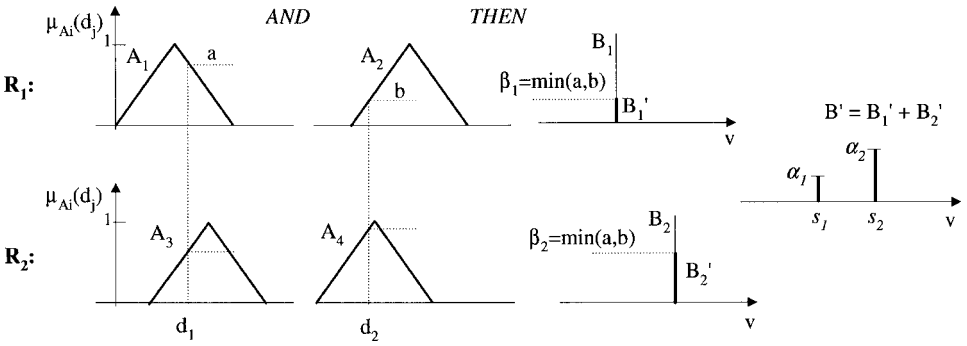


Figure 7. Schematic representation of the sum–product inference applied to singletons.

tribution of the resultant fuzzy set. The COG method, applied to fuzzy singletons, simplifies the computation of inference mechanism and reduces processing time without degradation of the defuzzified value [3]. This one is obtained from:

$$u = \frac{\sum_i \alpha_i s_i}{\sum_i \alpha_i} \tag{1}$$

where α_i is the degree of activation of the ith rule and s_i is the output singleton (see Figure 7).

4.5. RULE BASE

The rule base was written based on the user experience. Two approaches can be followed to write the rules. Write each rule attending several input variables and/or goals, or write each rule attending a unique input variable and/or goal. In the first approach, the behaviour's fusion occurs when the rule is written. In this manner, it avoids a further behaviour combination. Each rule inevitably falls on the desired reaction. It is well suited when the interaction between input variables/goals is important, as it happens with RSC. This motivated us to follow this approach. With

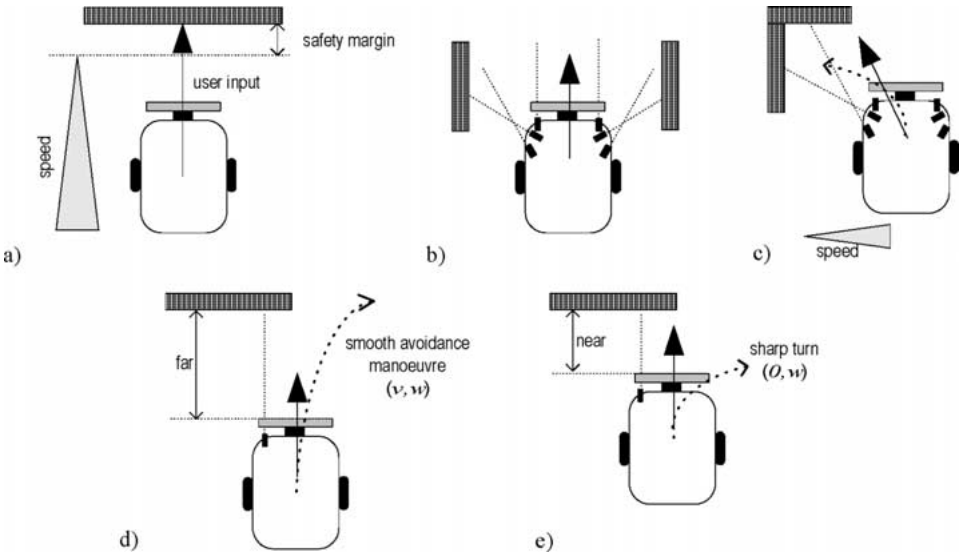


Figure 8. Examples of environment configurations. Solid arrow represents user's input direction.

the second approach, the rules are more easily written. However, it is more difficult to obtain a coherent reaction. Each behaviour output must be further combined. This makes it more sensitive to inference and defuzzification methods. Usually, the problem is to cope with contradictory objectives (authors in [8] present methods to overcome this problem).

5. Intelligent Obstacle Avoidance

Figure 8 presents some environment configurations that were very useful to define the set of rules. It was firstly defined that the controller should perform smooth trajectories. For instance, if obstacles are at a medium distance, the wheelchair has to perform a slightly rotation in order to contour the obstacle. If, otherwise, obstacles are close, then the wheelchair should perform a more accentuated rotation (see Figures 8(d) and (e)). These effects are obtained combining linear and angular velocities.

For guidance, five voice commands are used: *forward*, *left*, *right*, *turnleft*, *turnright*. *Left/Right* expresses the wish to follow in a direction around 30°. *Turnleft/turnright* rotate continuously until a *stop* command is given. Let see some possible scenarios:

- *No obstacles* – the controller follows user commands but with imposed limits for speed (linear and angular);
- *Obstacles in Front (both sides)* – if the user's command is to go straight, the controller reduces gradually linear velocity and stops the wheelchair when the distance to the obstacle is smaller than the safety margin. Angular ve-

locity remains zero. The controller is not supposed to perform any obstacle avoidance manoeuvre because the user wants to go straight and there is no passage (Figure 8(a)). This case can be, for instance, a desk-approach. If the user's command is to go left/right, the wheelchair turns 30° left/right and then follows straight.

- *Obstacles in front Left/Right (just one side)* – if the user's command is to go straight but there are obstacles in front left or front right, but not both, then the wheelchair has to perform a manoeuvre to contour the obstacle and then to follow straight. The avoidance manoeuvres should be the smoothest possible. If obstacles are far, the wheelchair turns smoothly and if the obstacles are close the wheelchair turn is sharper (Figures 8(d) and (e)).
- *Left/Right side Obstacles* – if the obstacles are at the left or right side of the wheelchair but not in front, and the user wants to follow straight, then the wheelchair goes straight. If the user commands the wheelchair to go in the obstacles direction, the controller follows the user input but reduces gradually the speed until a safety margin is overtaken (Figure 8(c)).

The following rules constitute part of the knowledge base of the IOA behaviour and express how the system has to react:

Rule Base

1. If θ_{goal} FR and $d0$ N and $d1$ N and $d2$ N and $d3$ N Then v S and w Z
2. If θ_{goal} FR and $d0$ F and $d1$ F and $d2$ F and $d3$ F Then v M and w Z
3. If θ_{goal} FR and $d0$ N and $d1$ F and $d2$ F and $d3$ N and $d5$ F and $d7$ F Then v Z and w RFS
4. If θ_{goal} FR and $d0$ N and $d1$ F and $d2$ F and $d3$ N and $d5$ and $d7$ N Then v S and w RS
5. If θ_{goal} FR and $d0$ F and $d1$ N and $d2$ N and $d3$ F and $d4$ F and $d6$ F Then v Z and w LFS
6. If θ_{goal} FR and $d0$ F and $d1$ N and $d2$ N and $d3$ F and $d4$ N and $d6$ N Then v Z and w LS
7. If θ_{goal} L and $d0$ N and $d1$ N and $d4$ F and $d6$ F Then v Z and w LFS
8. If θ_{goal} L and $d0$ F and $d1$ F and $d4$ F and $d6$ F Then v S and w LS
9. If θ_{goal} R and $d0$ N and $d1$ N and $d5$ F and $d7$ F Then v Z and w RFS
10. If θ_{goal} R and $d0$ F and $d1$ F and $d5$ F and $d7$ F Then v S and w RS

6. Experimental Results

RSC was tested on the RobChair prototype. The experiments were performed at the ISR (Institute of Systems and Robotics) laboratories. First experiments were performed without RSC assistance. These experiments were very useful to realise why it is so difficult to steer the wheelchair based on low-level voice commands. Figure 9(a) shows a path travelled by the wheelchair. Each small circle in the figure points out a voice command emitted by the user. The goal was to travel from point A to point B. The area of the environment in these experiments is approximately 4.5×6 m. Without RSC assistance the majority of attempts in performing this travel were unsuccessful (even considering that they were performed by an experienced

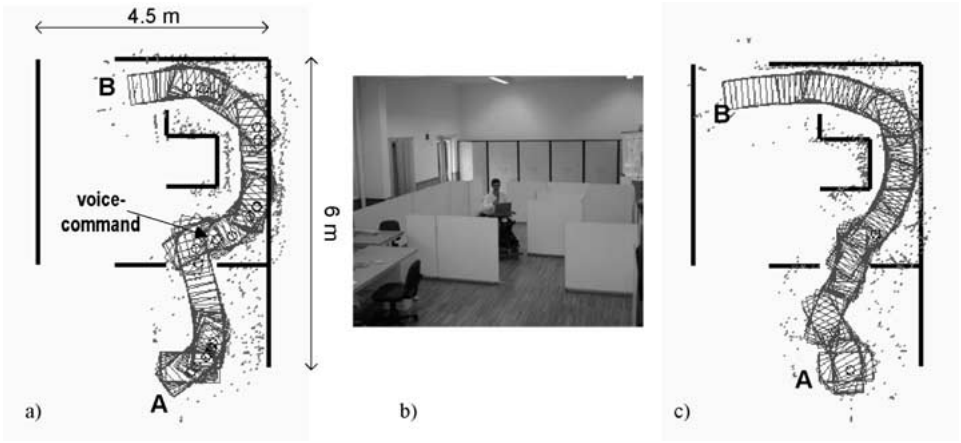


Figure 9. Real lab experiments with RobChair prototype. In these figures the representations of the results were constructed from the real sensory and actuation raw data collected during the navigation experiments, and were grabbed from the own GUI of the wheelchair, exactly as they are. Small circles represent the positions where the user provided voice commands. (a) Voice-steering without RSC assistance (15 commands were emitted); (b) Picture of the environment; (c) Voice-steering with RSC assistance (5 commands were emitted).

user) except the experiment shown in Figure 9(a). A typical unsuccessful attempt is displayed in Figure 10. The first passage (a kind of door) in the setup lab is extremely difficult to perform and therefore leads to failure. Despite the successful travel in Figure 9(a) the user had to emit 15 voice commands and the wheelchair suffered side and frontal collisions. This means that the user had to constantly emit commands, which increased significantly his steering effort, and the travel was unsafe. Another problem is the constant concern and stress of the user. He is constantly worried about eventual collisions and with the possibility of not emitting the command in time. These trials clearly showed that without a navigation assistant, voice-steering is impracticable. There are several reasons that justifies voice-steering difficulty:

- The user does not have a full perception of the space to perform some manoeuvres. Moreover, he is concentrated just with the front part of the wheelchair;
- There are some delays between the voice command issuing and the execution of the command;
- The stress/anxiety leads the user to emit wrong commands. The constant emission of commands also contribute to errors;
- The wheelchair does not always take the user desired direction. This problem is due to the front castors that may deviate the wheelchair from the desired direction.

The RSC controller was developed in order to overcome limitations of a non-assisted voice steering. Figure 9(b) shows a new experiment with RSC assistance. Comparing this experiment with the one of Figure 9(a) we immediately perceive

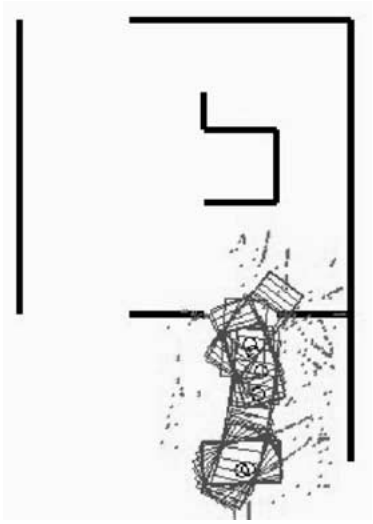
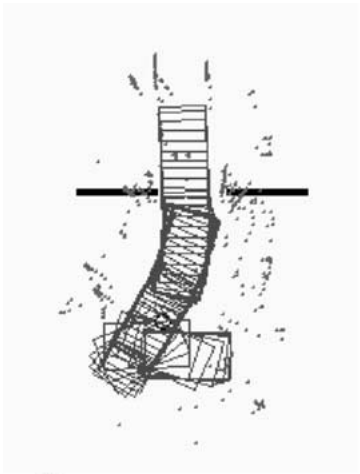


Figure 10. Typical unsuccessful experiment without RSC assistance.



a)

b)

Figure 11. Real lab experiments with RobChair prototype. Door-passage with RSC assistance: (a) Wheelchair trajectory; (b) Picture of the environment.

that the number of voice-commands decreased significantly (to one third), and that manoeuvres are smoother and safe. The user's steering became much more relaxed. The user knows that, even if a command is not given in time, or a wrong command is issued, the wheelchair will provide collision avoidance or obstacle avoidance depending on the situation. Despite of the poor IR information, the wheelchair trajectories are smooth and provide a reasonable degree of security. Wheelchair movements are slower when obstacles are near, and are faster in the absence of obstacles, therefore increasing user's confidence.

These results are encouraging, but it is clear to us that sonars and Laser Range Finder have to be used in order to improve obstacle detection, reduce the risk of collisions, and to allow local planning of trajectories. There are many situations in which local planning is indispensable. One of these has already begun to be studied: a door-passage. The IOA behaviour can pass a door in certain conditions, as illustrated in Figure 11. The passage is successfully performed because the wheelchair is forward oriented to the door. The IOA perceives the door just as a free path between two obstacles. If however, the approach to the door is oblique, the IOA is unable to pass the door. This task requires the identification and localisation of the door, and the planning of the wheelchair trajectory.

7. Conclusions and Future Developments

This paper presents a reactive shared control suited for Human-Oriented Mobile Robots. Currently the Robchair can navigate in dynamic environments in the presence of humans, commanded by simple low-level voice or joystick instructions. A pure reactive layer, enabling obstacle avoidance and contour-following, currently supports the navigation. An efficient map-building method was already developed [1] and is being integrated in Robchair to improve local navigation, improving safety and further reducing the effort of the user. Furthermore, research on trajectory control and planning, with application in nonholonomic robotic systems is being carried out to enhance the navigation capabilities of Robchair, and mobile robots in general.

References

1. Cruz, J. and Nunes, U.: Generating local maps for mobile robots, in: *Controlo2000: 4th Portuguese Conf. on Automatic Control*, Guimarães, Portugal, 2000, pp. 412–417.
2. Dragon Voice Tools: `http://www.dragonsys.com`.
3. Goodridge, S. and Luo, C.: Fuzzy behaviour for reactive control of an autonomous mobile robot: Marge, in: *IEEE Conf. on Robotics and Automation*, Vol. 2, San Diego, CA, 1994, pp. 1622–1627.
4. Mizumoto, M.: Fuzzy controls under product-sum gravity methods and new fuzzy control methods, in: *Fuzzy Control Systems*, CRC Press, 1994, pp. 276–294.
5. Nunes, U., Pires, G., and Coelho, P.: Assistive navigation control architecture, in: *Systems and Control: Theory and Applications*, WSES Press, 2000, pp. 38–43.
6. Pires, G., Araújo, R., Nunes, U., and Almeida, A. T.: Robchair: A powered wheelchair using a behaviour-based navigation, in: *5th IEEE Internat. Workshop on Advanced Motion Control (AMC'98)*, Coimbra, Portugal, 1998, pp. 536–541.
7. Tzafestas, S. (Guest Editor): Research on Autonomous Robotic Wheelchairs in Europe, *Reinventing the Wheelchair, Special Issue of IEEE Robotics Automat. Mag.* **7**(1) (2001).
8. Yen, J. and Pfluger, N.: A fuzzy logic based extension to Payton and Rosenblatt's command fusion method for mobile robot navigation, *IEEE Trans. Systems Man Cybernet.* (1995), 971–978.