Case-based reasoning emulation of persons for wheelchair navigation

Jose Manuel Peula, Cristina Urdiales, Ignacio Herrero, Manuel Fernandez-Carmona, Francisco Sandoval

[a] *Ingeniería de Sistemas Integrados Group, University of Málaga, Campus de Teatinos s/n, Málaga, 29071, Andalucia, Spain*

Abstract

Objective: Testing is a key stage in system development, particularly in systems such as a wheelchair, in which the final user is typically a disabled person. These systems have stringent safety requirements, requiring major testing with many different individuals. The best would be to have the wheelchair tested by many different end users, as each disability affects driving skills in a different way. Unfortunately, from a practical point of view it is difficult to engage end users as beta testers. Hence, testing often relies on simulations. Naturally, these simulations need to be as realistic as possible to make the system robust and safe before real tests can be accomplished. This work presents a tool to automatically test wheelchairs through realistic emulation of different wheelchair users.

Methods and materials: Our approach is based on extracting meaningful data from real users driving a power wheelchair autonomously. This data is then used to train a case-based reasoning (CBR) system that captures the specifics of the driver via learning. The resulting case-base is then used to emulate the driving behavior of that specific person in more complex situations or when a new assistive algorithm needs to be tested. CBR returns user's motion commands appropriate for each specific situation to add the human component to shared control systems.

Results: The proposed system has been used to emulate several power wheelchair users presenting different disabilities. Data to create this emulation was obtained from previous wheelchair navigation experiments with 35 volunteer in-patients presenting different degrees of disability. CBR was trained with a limited number of scenarios for each volunteer. Results proved that: (i) emulated and real users returned similar paths in the same scenario (maximum and mean path deviations are equal to 23 and 10 cm, respectively) and similar efficiency; (ii) we established the generality of our approach taking a new path not present in the training traces; (iii) the emulated user is more realistic – path and efficiency are less homogeneous and smooth – than potential field approaches; and (iv) the system adequately emulates in-patients – maximum and mean path deviations are equal to 19 and 8.3 cm approximately and efficiencies are similar – with specific disabilities (apraxia and dementia) obtaining different behaviors during emulation for each of the in-patients, as expected.

Conclusions: The proposed system adequately emulates the driving behavior of people with different disabilities in indoor scenarios. This approach is suitable to emulate real users' driving behaviors for early testing stages of assistive navigation systems.

Keywords: Case-based reasoning, Behavior learning, Behavior prediction, User emulation, Power wheelchair

---

1. Introduction

Advances in medicine have increased life expectancy leading to a larger elderly population [1]. It is important for these people to maintain their quality of life, independence and autonomy as long as possible [2]. Mobility is a key factor in elderly people's quality of life [3] and, indeed, many of them normally use assistive devices in activities of daily living (ADL) [4] in order to enhance their autonomy [5].

Wheelchairs are a primary mobility aid for the elderly [6], and have been widely studied [7–9], especially: (i) system development [8], (ii) user performance evaluation [10,11], and (iii) wheelchair simulation and testing [12–17]. In a system such as a wheelchair, simulations and tests are particularly important, as its functionality and safety have to be checked starting in the early stages of development. These simulations and tests should be done with a representative group of the target population, as the impact of disability on driving skills changes significantly from one person to another. However, it is not always possible to involve such a group, and hence tests are usually performed by a reduced group of volunteers or by analytical algorithms that simulate dummy users.

A system to emulate target population behavior could be very useful for wheelchair control testing. This system would allow for more complete and realistic test/simulation stages than those of analytical simulations or a reduced number of volunteers. The reason for using emulation instead of simulation is that simulation mimics the outward appearance of a process, but emulation mimics the cause/process. Thus, we only need to know the input/output patterns – in our case, user's motion commands for each specific situation –, but not the internal mechanism of the original system – in our case, a person.

This paper proposes a system that extracts driving habits and skills from data gathered in real tests in order to emulate different kinds of users. Specifically, we use case-based reasoning (CBR) to learn how a given user reacts to a situation when driving a wheelchair. Although emulated users in our tests are people with disabilities, this work does not focus on understanding why a given person drives a wheelchair one way or another depending on his/her disability but just on mimicking users' driving through imitation learning. Our system operates at the reactive level, where driving can be explained by a limited set of cases that, when combined, can build more complex emergent behaviors [18–22]. Thus, the system could generate complex paths with relatively few training sessions. However, these training sessions should be significant and cover most of the situations that an emulation will face. Otherwise the user's behavior would not be correctly modeled or emulated. The main novelties of the proposed approach are that: (i) it is not necessary to model how a real person drives a wheelchair or moves, as the proposed system learns from data extracted from real people when driving a wheelchair and (ii) the system can generate complex paths with relatively little training.

The proposed system would automatize simulations and make them more realistic – interesting for developers – and it would make the system more robust and adapted to final users' needs – interesting for users. Thus, the proposed method could be a new part of the test stage prior to real tests with in-patients. The paper is structured as follows. Section 2 describes the method used to learn how a person drives by imitation at the reactive level. Section 3 shows the results and experiments. Data used in the experiments for CBR training was gathered from real traces performed by inpatients at Fondazione Santa Lucia (FSL) hospital. In this section we: (i) compare results of emulated in-patients with real in-patients (Section 3.2); (ii) evaluate the generality of the case-based approach (Sections 3.3 and 3.4), including a comparison of results of emulated in-patients with results of a potential field approach (PFA) (Section 3.4); and (iii) compare results of emulated in-patients with those of real in-patients for specific disabilities (apraxia and dementia) (Section 3.5). Results obtained were presented to medical doctors, who agreed that they were coherent with what they expected. Finally, Section 4 sums up our conclusions.

2. Development of the system

2.1. Biomimetic motion characterization

In order to reproduce a person's motions, it is first necessary to determine which algorithms can be used by a mobile robot to autonomously calculate and follow a path in any given dynamic environment. This problem is generally known in the literature as navigation.

Navigation systems are either deliberative, reactive or hybrid [23]. Deliberative control is the oldest one and it follows a sense-plan-act scheme (e.g. [24]), meaning that the mobile robot perceives its environment, reasons about it, and then moves. Deliberation requires a reliable model of the environment to reason about it and, consequently, is quite sensitive to model errors. Alternatively, reactive robotic controls relate sensor readings and actions to sense/action pairs [25]. Reactive systems are typically less affected by errors and do not require explicit models of the environment. Furthermore, they usually only deal with the local information

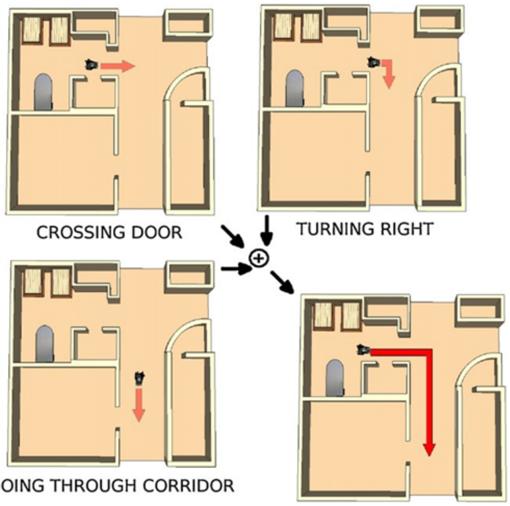


Figure 1: Creation of an emergent behavior (going from room to corridor) from simpler reactive ones (crossing door, turning right and going through corridor).

captured at a time instant. However, reactive systems may fall into local traps and are usually less efficient than deliberative ones. Consequently, both approaches are currently combined into hybrid systems – more scalable than purely reactive ones –, where low level control operates in a reactive way, whereas high level systems tend to be deliberative [26].

A taxonomy of navigation systems, known as the navigation hierarchy, is proposed in [27]. Its main advantage is that it is applicable to both animals and robots and it relies on progressively complex navigation behaviors, starting with reactive and moving up to deliberative navigation. Each behavior includes previous ones and is associated with more complex entities. However, to this date, most work on biomimetic navigation not based on world modeling has focused on behavior-based lower levels of the hierarchy [28–30]. This could be explained by reactive systems being more appealing to intuitively model complex real behaviors in a simpler way than high level algorithms. Furthermore, human behavior, especially in dynamic environments, is not as predictable as it should be to properly model its specifics.

Different authors [18–22] propose that a complex behavior can be modeled through the combination of simple ones. Following this approach, in this work a complex behavior – such as going from room to corridor – is divided into simpler tasks or basic behaviors – crossing door, turning right, following corridor, etc. These basic behaviors correspond to decisions made by the user in distinctive situations, whereas the emerging complex behavior – sum of the basic behaviors – corresponds to the resulting path in a specific environment and circumstance (Fig. 1). As commented, reactive systems are based on coupling sensor readings and commands to obtain a sense/action pair. Thus, these basic behaviors can be implemented as reactive behaviors by coupling a user's movement – a joystick command – to each situation – a laser sensor and goal. A robot could make the action/sensor association via an analytical, reactive algorithm, such as PFA [31]. PFA basically relies on modeling obstacles as repulsors and goals as attractors to create a vector field that returns a motion vector at each point.

Thus, PFA provides a simple and efficient tool for autonomous motion for the wheelchair. However, an analytical algorithm differs from human driving behavior as humans have habits that are influenced by age, gender, medical condition, experience, etc. Since condensation of all these factors within an analytical algorithm is far from simple, our work is based on learning this association via CBR. Since PFA is one of the best known approaches to reactive motion, it is used in this work for baseline comparison with our

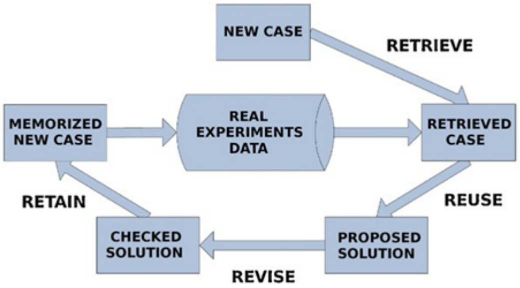


Figure 2: Typical CBR cycle.

learning approach.

Our basic idea is to extract users' low level behaviors when driving a wheelchair from available real traces gathered in previous experiments. Available traces include information such as sensors readings, odometry, position or user joystick movement for each trajectory point. From this data, relevant information – sensor readings, goal and joystick movement – is stored as a case. Then, a CBR learning technique is used to create and manage resulting behaviors, as explained in the next section.

2.2. CBR for navigation behavior modeling

CBR [32] is a reasoning, learning and adaptation technique to solve problems by retrieving and adapting past experiences [33,34]. CBR is mostly derived from the theory of the Dynamic Memory [35], which introduces indexing as the key to use experience in understanding.

A CBR system cycle to solve a new problem consists of four steps (Fig. 2), known as 4R format (retrieve, reuse, revise and retain): (i) retrieve the stored case[1] or cases that are the most similar to the new current case; (ii) reuse the retrieved solution, and adapt it to the new solution; (iii) revise the results of the proposed solution; and (iv) retain from the new experience. In the CBR cycle the input problem is first characterized by means of a number of significant parameters, which, along with the problem solution, comprise a case. Definition of a correct problem instance is of key importance in this kind of system. The input instance is matched against all known cases, so

---

[1] Some authors think that the most adaptable case (the one requiring the least adaptation effort) should be retrieved. In our case, case adaptation is not used, so the most similar stored case is retrieved.

that the most similar one in the case-base is retrieved. After this stage, the retrieved case is compared with the input situation to adapt it if necessary. Finally, if the case is adapted, it is evaluated and stored for future use. In our case, however, only a 2R format – retrieve and reuse – is used in the CBR cycle – revise and retain steps are not used – in order to avoid modifications of the patients' behavior patterns. However, if it is considered interesting or necessary to use, revise and retain stages, PFA could be used to adapt cases to new input situations.

CBR has been used in many different areas [36], especially in the medical field. CBR has been applied for more than 20 years [37] in different fields [38–41], such as image analysis [42,43], psychiatry [37], diagnosis and instruction [44], as well as assistance of elderly and disabled persons [45].

CBR has also been used in robotics, more specifically in robot navigation typically for high level planning, behavior selection and transition [46] and selection of behavioral parameters [47]. CBR in navigation has been used for global path planning in static environments [48,49], where cases absorb the structure of the environment. Other navigation methods based on CBR focus on non-pure reactive navigation [50,51], but they basically rely on accumulating experience over a time window while navigating in a given environment to obtain an emergent global goal-seeking behavior. Global planning in dynamic environments based on a topological map of an a-priori known environment has also been developed [52], but new opportunities can not be discovered when the environment changes unless the topological map is regularly reorganized.

The authors proposed the use of CBR to model reactive navigation behavior via learning by imitation [53]. The key idea was to represent a pure local behavior using just a few parameters. Cases created included the robot on-board sensor readings – including heading to take holonomicity[2] into account – and output direction movement; the case-base thus implicitly contains information on the specifics of robot kinematics and sensor modeling. In this paper, a similar CBR approach is used to emulate the behavior of a real user in a purely reactive fashion. Basically, the case-base absorbs the specifics of the user's way of driving at the local level and combines them to produce complete paths, driving like the real user would. The following sections present the most relevant aspects of the proposed approach.

---

[2] The ability of the robot to move instantaneously in any direction

2.2.1. CBR case-base description

In our system CBR is used as a reactive navigation layer for the wheelchair, allowing us to emulate real in-patients. To do so, CBR learns from real in-patient traces obtained from different real tests [54]. Each case-base stores information about sensor readings, goal and user motion command in each given situation. Our case-base implicitly contains information on the specifics of robot kinematics and sensor modeling [19], and information about the behavior of a real user while driving the wheelchair.

The steps to model information from an experiment into a casebase are (Fig. 3):

1. Data from the robot (sensor readings, goal) and the subject (motion commands) are recorded while subjects are following the test paths.
2. Stored data locally associated with each local point is turned into – reactive – cases. Those cases are stored into a case-base. Casebase, case format and memory storage are explained in Sections 2.2.2 and 2.2.3 respectively.
3. Finally, a clustering algorithm is applied to the case-base. Cases are merged into a single one to avoid excessively large case-bases and to reduce oscillations provoked by different commands associated with similar sensor readings. This is explained in more detail in Section 2.2.4.

2.2.2. Case-base definition

The main concern of navigation is simply to reach a goal point in a safe way. Consequently, the simplest problem instance for a navigation algorithm consists of goal and obstacles, while the output is basically which direction to follow to safely reach the goal. However, it is not simple to understand how each given person processes navigation. This paper aims at capturing purely reactive behaviors, meaning that we do not try to understand how a person thinks, but rather to model his/her behavior as a black-box. Locally only sensor readings representing surrounding obstacles and the goal to reach are required to model a situation. Thus, the input vector (14 components) comprises the wheelchair sensor readings (12 laser samples) and goal location (x and y position) with respect to the wheelchair, whereas the output vector (2 components) corresponds to the joystick command (rotational and translational commands). These input–output vectors are associated into pairs to compose a case (Fig. 4). Hence, the case-base stores what the user does in a specific situation. Fig. 4 shows an example of a real input/output CBR vector and the corresponding situation. Finally, when an input vector (formed by sensor readings and goal) is stored in the CBR case-base, each component of this vector (sensor readings and x and y goal values) is categorized into ranges corresponding to very near/near/medium/far/no, as proposed by the authors [55]. The idea of this categorization is to imitate what a human does, namely to take into account an obstacle depending on its distance.

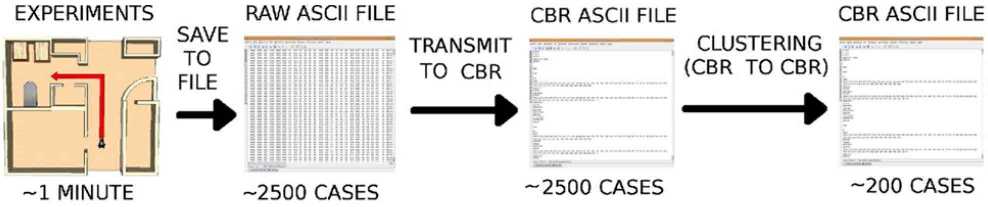


Figure 3: Data flow from experiments to final CBR case-base.

2.2.3. Case-base storage

Main memory organization in CBR systems [32] can be summarized in two general approaches: flat memories and hierarchical memories.

- Flat memories retrieve the set of cases that match best with the input case. Moreover, adding new cases to memory is cheap. However, retrieval time may become very high.
- Hierarchical memories, such as shared feature networks, prioritized discrimination networks/trees or redundant discrimination networks/trees allow more efficient retrieval, but keeping the hierarchical structure in optimal conditions requires a significant overhead in respect to case library organization.

In our approach, CBR case-bases are not very large (around 200 cases after clustering), so a flat memory organization has been selected (Fig. 5).

2.2.4. Clustering

After creating a CBR case-base, clustering is used to reduce the number of stored cases to a representative set, formed by cluster prototypes (CP). A CP is defined as the case resulting from clustering a set of cases into a single one, and represents the user's average motion command when facing similar situations. Thus, clustering removes noise and variability in solutions to similar situations. However, note should be made that clustering does not remove non-efficient or dangerous motion commands if the user provides them while driving the wheelchair, when data used in CBR is gathered.

In our case, data is stored every 10–20 ms, approximately. Hence consecutive cases may be very similar, as sensor readings, goal and motion commands do not change that fast, and local situations may be repeated. Thus, stored information can be reduced using clustering, without substantial modification of the original data.

The clustering algorithm chosen is one of the simplest. First, a case not associated with any CP is chosen as temporal seed ($\vec{S_n}$) and associated with a new CP (CP_n). After choosing the seed ($\vec{S_n}$), all cases similar to ($\vec{S_n}$) are looked for in the case-base. A tested case ($\vec{TC}$) is associated with (CP_n), which corresponds to the seed ($\vec{S_n}$), when the distance (D_n) between ($\vec{S_n}$) and $\vec{TC}$ is under a threshold value. If the distance (D_n) (Eq. (1)) is under the threshold value, the tested case ($\vec{TC}$) is marked as belonging to (CP_n) and will not be used in any other CP. The distance threshold is set heuristically, although it is quite stable against different persons and robot sensors.

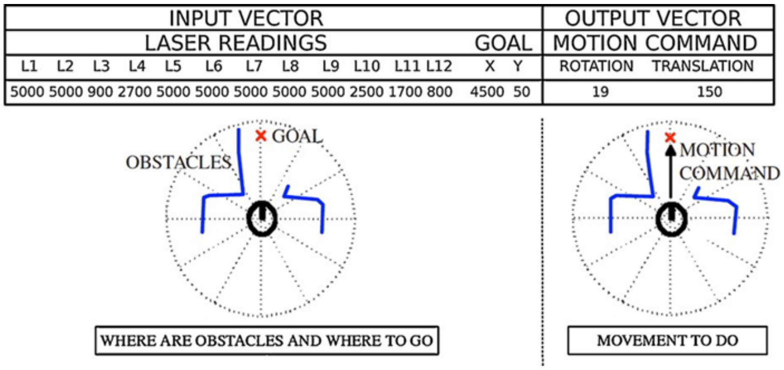
The value of the distance threshold is approximately between 2 x $range_{near}$ and 5 × $range_{near}$, $range_{near}$ being the value used in the categorization. We chose this value because it makes all cases within CPs similar to each other in one range or less (very near/near/medium/far/no) in at least nine out of 12 onboard sensors. Threshold changes within the proposed range only provoke minor differences in the resulting clusters that had no major impact on the experiment results. We checked many experiments to confirm that this value is appropriate for indoor environments like the ones in our experiments; outdoor experiments would probably need a different approach. Finally, after associating cases with (CP_n), these cases are averaged (Eq. (2)) and this averaged vector (CP_n) is saved in the final CBR case-base.

$$D_n = \sqrt{\sum_{i=1}^{N_L} (\vec{S}_n(i) - \vec{S}C(i))^2} \tag{1}$$

$$\vec{CP}_n = \frac{\sum_{i=1}^{N_{cases}} (\vec{TC}_i)}{N_{cases}} \tag{2}$$

N_L being the length (number of components) of vectors ($\vec{S_n}$) and $\vec{TC}$

(they have the same length) and Ncases the number of cases (after browsing all case-bases) associated with the cluster prototype (CP_n).



INPUT VECTOR														OUTPUT VECTOR	
LASER READINGS												GOAL		MOTION COMMAND	
L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	X	Y	ROTATION	TRANSLATION
5000	5000	900	2700	5000	5000	5000	5000	5000	2500	1700	800	4500	50	19	150

Figure 4: CBR case structure is formed by the input vector (12 laser samples in mm, goal position in mm) and the output vector (rotational and translational commands in mrad/s and mm/s respectively). The CBR case shows the values corresponding to the situation represented.

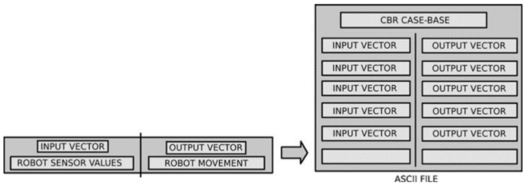


Figure 5: CBR case storage is organized as a flat memory.

2.2.5. CBR in standalone mode

After clustering, the clustered case-base feeds the CBR server when it works in standalone mode. In this mode, when an input vector (sensor readings plus goal) arrives, the CBR server looks for the most similar case available (in the retrieval step) and returns it to the client. The most similar case is the one whose Euclidean distance to the input vector is the lowest. The retrieved case includes the motion command provided by the subject in the situation the most similar to the one defined by the input vector. This motion command is what the emulated in-patient assumes that the user would have done in the current situation. Finally, the retrieved case is directly sent to the platform motors.

One of the main features of CBR is that it always retrieves a case, despite its similarity to the input instance. This means that sometimes the retrieved case is not that similar to the input vector. Usually, CBR provides an adaptation stage when the distance between the input instance and the retrieved case is larger than a threshold. In our case, though, this option is disabled in order to preserve the integrity of the original data. Consequently, it is important to have enough cases in the case-base to cover all the kinds of situations that the emulation could face to avoid the need for adaptation. Prior works [55] have verified that purely reactive navigation can be modeled with a reduced amount of data. In this paper, users are modeled with just a single path of approximately 12 m, having verified that navigation was correctly achieved in typical indoor situations including door crossing, follow corridor, turn left and right, etc., as shown in Section 3. It must be noted, though, that in future situations adaptation or learning from new situations could be necessary to model more complex behaviors or environments.

2.3. Efficiency

Efficiency is a metric applied to each motion command (user, PFA or CBR). It is used to evaluate how good or bad a motion command is, for example to evaluate the user's performance. It is important to take into account that no global factor can be used to rank a purely reactive case. Hence, global efficiency (η) is evaluated as the average of three factors (Fig. 6): smoothness (η_{sm}), safety (η_{sf}) and directness (η_{di}), each of them ranging from 0 to 1. η_{sm} reflects how sharply the wheelchair changes direction when driving (sharp changes mean low η_{sm}). η_{sf} reflects that it is better to keep away from obstacles (the closer the obstacles, the lower the η_{sf}). η_{di} tries to evaluate if the user approaches the goal in a straight direction, leading to shorter paths and higher η_{di} or not. The importance of each of these factors in the global efficiency is controlled by three constants (C_{sm}, C_{di}, C_{sf}, corresponding to smoothness, directness and safety respectively). In our case, these three constants are equal to 1 – all factors have the same importance.

Smoothness (η_{sm}) is locally evaluated as the angle (α_{smooth}) between the current direction of the robot and the output motion vector. It is important to take into account that many robots are non-holonomic[3]. Thus, it is better

---

[3] Non-holonomic robots are the ones that cannot instantaneously move in any direction,

to change heading as little as possible to avoid slippage and oscillations. Eq. (3) shows how η_{sm} is calculated, α_{smooth} being the angle difference between the current heading and the command vector. Thus, low α_{smooth} – smooth trajectory – means high η_{sm} – high efficiency – and vice versa.

$$\eta_{sm} = e^{-C_{sm}\cdot|\alpha_{smooth}|} \begin{cases} \alpha_{smooth} = \pi/2 \Rightarrow \eta_{sm} \approx 0(Min.efficiency) \\ \alpha_{smooth} = 0 \Rightarrow \eta_{sm} = 1(Max.efficiency) \end{cases} \tag{3}$$

Directness (η_{di}) is locally measured in terms of the angle (α_{direc}) comprising the output motion vector and the direction towards the next goal. Obviously, the shortest way to reach that goal is to make α_{direc} approximately 0, making η_{di} the maximum. Eq. (4) shows how η_{di} is calculated, α_{direc} being the angle between the output motion vector and the direction towards the next goal.

$$\eta_{di} = e^{-C_{di}\cdot|\alpha_{direc}|} \begin{cases} \alpha_{direc} = \pi/2 \Rightarrow \eta_{di} \approx 0(Min.efficiency) \\ \alpha_{direc} = 0 \Rightarrow \eta_{di} = 1(Max.efficiency) \end{cases} \tag{4}$$

Finally, safety (η_{sf}), is evaluated in terms of the angle (α_{safe}) between the output motion vector and the direction to the closest obstacle. Thus, when the wheelchair moves towards the nearest obstacle, α_{safe} and η_{sf} decrease – the trajectory is not safe. Eq. (5) shows how η_{sf} is calculated, α_{safe} being the angle between the output motion vector and the closest obstacle. Thus, if motion vector points to the closest obstacle ($\alpha_{safe} \approx 0$) η_{sf} is minimum ($\eta_{sf} \approx 0$).

$$\eta_{sf} = 1 - e^{-C_{sf}\cdot|\alpha_{safe}|} \begin{cases} \alpha_{safe} = 0 \Rightarrow \eta_{sf} \approx 0(Min.efficiency) \\ \alpha_{safe} = \pi/2 \Rightarrow \eta_{sf} = 1(Max.efficiency) \end{cases} \tag{5}$$

3. Experiments and results

This section presents several tests and results emulating how a given person with a specific pathology drives a wheelchair. These tests have been

---

such as a car.

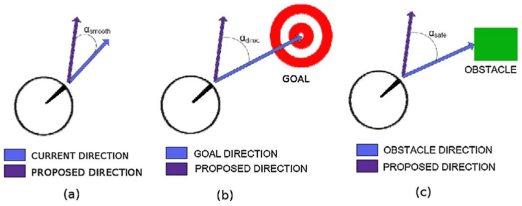


Figure 6: Efficiency factors: (a) smoothness, (b) directness and (c) safety.

chosen among all those available because they are representative of the different activities of daily living (ADL). As discussed, all training and testing is based on data from experiments conducted at FSL in Rome. All experiments were performed by volunteer in-patients and approved by the FSL ethical committee. Tests included up to 35 in-patients presenting different degrees of disability, either physical, cognitive or both, evaluated by the caregivers via different indexes like mini-mental state examination (MMSE)[4] [56] or Barthel[5] index [57].

Tests were performed in an indoor hospital environment. Following the caregivers' advice, persons were assigned a common task: entering and exiting a room from/into a corridor. Experiments follow the layout in Fig. 7. In the first layout (Fig. 7a) the in-patient has to go out of the room, turn right and then follow the corridor for 8 m. The second one (Fig. 7b) is the reverse path, so the in-patient has to go through the corridor and then turn left to come into the room. It must be noted that paths are not symmetrical, as it is easier to move out of the room, while facing the door, and turn right whenever the user decides to, than to choose exactly when to turn left to face the door and cross it.

These scenarios are simple but representative, and in order to compare emulated paths with real ones, the same scenarios have to be used. Furthermore, it is necessary to keep in mind that no adaptation of the CBR cases was allowed during the experiments. In order to extrapolate the results to new, different situations (for example, an environment with moving obstacles) it

---

[4] The MMSE or Folstein test is a brief 30-point questionnaire test commonly used for complaints of memory problems to estimate the severity of cognitive impairment or when a diagnosis of dementia is being considered.

[5] The Barthel index is a scale used to measure performance in basic ADL. It uses ten variables describing ADL and mobility. The higher the index is, the greater the degree of independence.

would be necessary to use adaptation or to learn from new situations.

During recorded experiments, in-patients repeated each path several times (normally 2–3 training sessions). The case-base of the emulated in-patient is created only using data from one of the training sessions. Thus results from the emulation are compared with real results from the corresponding training sessions. If both results are similar, it means that the emulated in-patient works fine. The similarity between the emulated and the real in-patient is measured in two ways:

1. The distance between the real path and the emulated one. If the difference between the real and emulated path is small, it is assumed that the emulated in-patient is doing well. As the whole path is 10 m long, and localization in real tests was only based on the wheelchair odometry, a difference of about 1–2% (10–20 cm) is considered to be a good result.
2. The efficiency of the real and emulated in-patients. Although both paths (emulated and real) could be very similar, it is necessary to see if their efficiencies are similar too (variation and waveform). Similar efficiencies mean that motion commands are similar as well in terms of efficiency, and therefore, that the real in-patient has been correctly emulated.

However, it is not only interesting to check if an emulated person behaves similarly to the real one, but also to check that different emulated persons drive differently[6] as well, because different in-patients behave differently depending on their residual skills. Fig. 8 shows how most in-patients diverge in their paths. Although there are differences in straight paths, the most visible ones occur at the turning point to either face the door or enter the corridor. It is at this point that most variations in emulated in-patients are expected.

3.1. Case-base: autonomous robot based on PFA

In order to prove that the emulated in-patients do not drive like analytical navigation algorithms, this first test is used for comparison with analytical

---

[6] It is important to take into account that the goal of this paper is not to relate disabilities with driving tendencies, as that would be a complex medical goal.

algorithms. In this test the robot follows the trajectory in Fig. 7a using PFA as its navigation algorithm, without human intervention.

PFA [31] is one of the most used and best known analytical algorithms in reactive navigation. It is widely used due to its simplicity and easy implementation.

This algorithm models the robot/wheelchair as a free particle moving in a potential field, where obstacles have the same charge as the robot (thus exerting a repulsion force $\vec{F}_{rep}$ on the robot), whereas the goal has an opposite charge (exerting an attraction force $\vec{F}_{attract}$ on the robot). Thus the robot/wheelchair tends to go to the goal, avoiding the obstacles. Fig. 9 shows an example of PFA forces (where $\vec{F}_{obstacle} = \vec{F}_{rep}$ rep and $\vec{F}_{goal} = \vec{F}_{attract}$) and Eq. (6) shows the general equation of the net force F that the robot perceives from the environment.

$$\vec{F}(x) = \vec{F}_{attract}(x) + \vec{F}_{rep}(x) \tag{6}$$

Fig. 10a shows the robot's path, based on PFA, and Fig. 10b its efficiency at every point of the path[7]. It can be observed that the emerging path is smooth and fairly straight, and efficiency is about 95% on average, with a minimum of about 80% when crossing the door. Again, this fact can be explained by oscillations caused by pure PFA when crossing narrow spaces, a phenomenon widely documented in related literature.

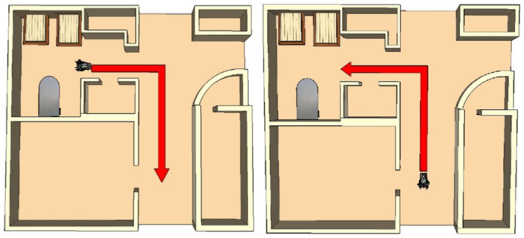
It is interesting to note that, given the departure and arrival locations, and if sensor errors are dismissed, the resulting analytical path given by PFA must be the same no matter how many times the robot repeats the test. Obviously, in real environments this is not always the case, as errors do exist. However, using a laser range sensor and paths as short as the ones used in our experiments, errors are not significant. Thus, it is assumed that PFA returns a canonical, purely reactive path for an initial/departure point pair in our experiments.

3.2. Test 1: room to corridor path and vice versa

This section presents a first set of experiments to test the emulated inpatients. In these tests, data from real users was used. As commented, each user repeated the path in different training sessions (between 2 and 3).

---

[7] In the paths, samples are not equidistant, so there are parts with a greater concentration of points.



(a) (b)

Figure 7: Paths of the experiments: from room to corridor (a) and from corridor to room (b).

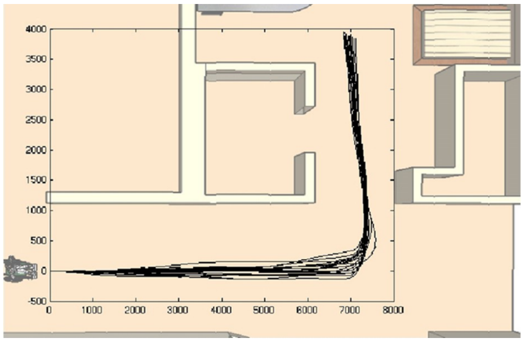


Figure 8: Variation of the paths for different patients and repetitions.

Thus, each time, data from one of the training sessions of a given in-patient is chosen to fill a casebase. Then, in-patients are emulated and results are compared with results of the corresponding real in-patient training session. Similar paths and efficiencies mean that the in-patient systematic driving features are correctly gathered by the system for the trained path.

	Room to corridor test			Corridor to room test		
	TS. 1	TS. 2	TS. 3	TS. 1	TS. 2	TS. 3
Max Deviation (cm)	16.5	16.3	15.8	23	19.5	20
Mean Deviation (cm)	7.02	7.04	7.14	10.23	10	8.99

Table 1: Differences between the emulated and the real paths in the room to corridor test – and vice versa – for a given in-patient in three different training sessions (TS).

Fig. 11a shows the real path (grey) corresponding to a training session, and the emulated one (black) for the same in-patient and trained path – room to corridor. Fig. 11d shows the same comparison in the corridor to

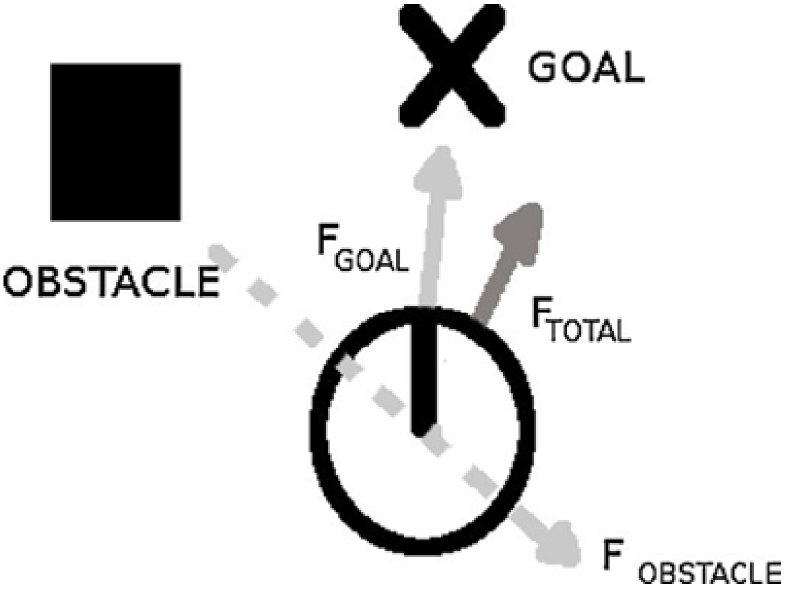


Figure 9: Example of forces over robot using PFA.

room path. It can be observed that real and emulated paths are quite similar, although there are some minor differences. Table 1 summarizes results – maximum and mean deviations – in different training sessions for a given in-patient in each of the tests – room to corridor and vice versa. In the case of the room to corridor test, the maximum deviation between the original and the emulated path is equal to 16.5 cm, while the average deviation is equal to 7.02 cm. In the case of the corridor to room test, the maximum deviation is equal to 23 cm and the average deviation is equal to 10.23 cm. In an 11-m long route, these deviations are considered non-significant. These differences may be caused by nonsystematic driving behaviors that correspond to non-repetitive action/reaction pairs that do not appear so often in a path. Hence, they are not captured by CBR. In some cases, if the case-base is trained with a large enough number of paths, these errors might be reduced, but there will always be some differences due to the unpredictable nature of human beings.

In order to prove that the emulated in-patients and real inpatients navigate similarly, Fig. 11 also presents the efficiencies of the real in-patients versus emulated ones. As efficiencies in both paths are similar, it can be concluded that both motion commands are similar as well, and that:

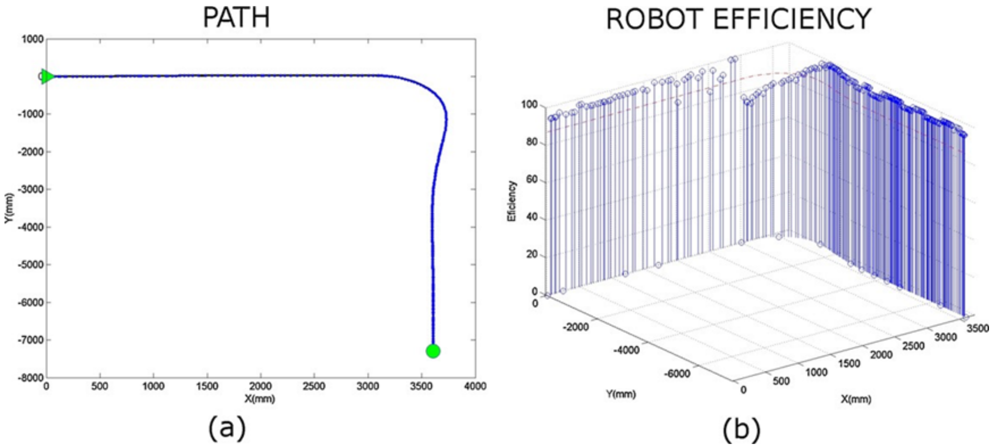


Figure 10: Path (a) and efficiency (b) of an autonomous robot based on PFA. (x, y) is the position of the robot in millimeters and z (in b) is the motion command efficiency at each position.

- Overall, both efficiencies are similar throughout most of the path – average efficiency is equal to 76.03% and 74.39% in the room to corridor experiment and values are similar in the corridor to room experiment, although there are larger differences at some points. This happens when a person is able to produce some commands, yet fails to provide them occasionally. These specific errors are filtered during the case-base clustering. Despite this, motion produced by the emulated in-patient and the real one are roughly equal.

- The efficiencies from real in-patients are more random than those for emulated in-patients. Specifically, the standard deviation for the real in-patient doubles the emulated one – 22.35% versus 11.25% in the room to corridor experiment and similar values in the corridor to room experiment. This could happen because of the clustering stage, which keeps a bounded case-base and grants faster access to the cases. This means the CBR returns the same output vectors to very similar input vectors. Therefore, the efficiency in emulated in-patients is more homogeneous than the efficiency from real in-patients. This could be changed by making our CBR system more sensitive to differences in the input, for example by reducing the clustering threshold and increasing the number of cases in the case-base. However this could make the platform unstable and reduce the time response from the CBR system

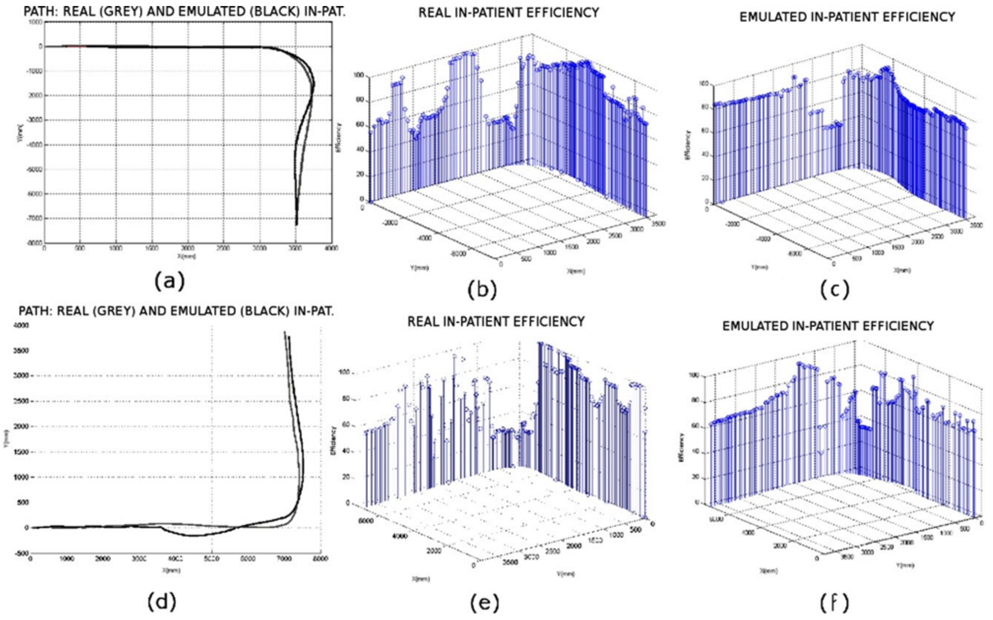


Figure 11: Path of the emulated and real in-patient, real in-patient efficiency and emulated in-patient efficiency for room to corridor test (a, b and c respectively) and corridor to room test (d, e and f respectively). (x, y) is the position of the robot in millimeters and z (in b, c, e and f) is the motion command efficiency at each position.

– the larger the number of cases in the case-base, the slower the time response from the CBR system.

3.3. Test 2: corridor-to-room path based on room-to-corridor data

This section repeats the experiments with the emulated inpatient for the second environment, but using the first experiment's case-base. Basically, this experiment tries to show what the emulated in-patient would do in order to move from a corridor to a room when the CBR case-base has only learned how to move from a room to a corridor.

The objective of this experiment is to prove that CBR only relies on the information stored in the case-base and may require some training to solve new situations. Although coming out of a room into a corridor or vice versa might look like a very similar task to an observer, it must be noted that the emulated in-patient has only learned to turn right and now it is asked to turn left (according to the previous paths). Furthermore, in the first case the turning point is not as important an issue as in the second case, where that decision is critical to be able to enter the door.

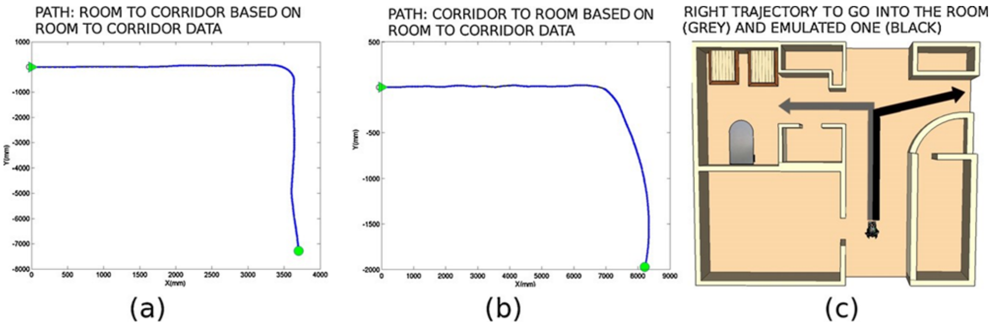


Figure 12: Path of the emulated in-patient for: (a) room to corridor test with data based on the same test, (b) corridor to room test with data based on room to corridor test and (c) corridor to room test with data based on the same test (grey), superimposed with an emulated in-patient for corridor to room test with data based on room to corridor test (black).

Fig. 12a shows the learnt path. Fig. 12b shows the path accomplished by the emulated in-patient when using data from the room to corridor test in the corridor to room environment. It can be seen that the emulated in-patient is able to move through the corridor. However it is not able to come into the room, as the CBR case-base has no information about how to turn left. Fig. 12c shows the correct path that the emulated in-patient should follow (grey), and the real path the emulated in-patient follows (black). As expected, the emulated in-patient only knows how to follow the corridor and turn right, and this is what the emulated in-patient does, instead of going into the room.

This test is an extreme case chosen to show that CBR only knows what it has been trained to do. Anyway, this problem is easy to solve, as CBR only needs to be trained with an in-patient path in which he/she comes into a room to the left, or turns left facing a narrow corridor or similar.

3.4. Test 3: room–corridor–room path

This section emulates the behavior of an in-patient in a new route that has not been tried in FSL tests. This is particularly useful for designers and caregivers to have an estimation of how a person presenting a particular profile would behave in a complex or somewhat hazardous situation or in a given specific environment. The experiment (Fig. 13a) consists in making the emulated in-patient leave a room (3 m), turn, head for the corridor along a straight path (8 m) and then enter into a different room (3 m) using data from corridor navigation and door crossing.

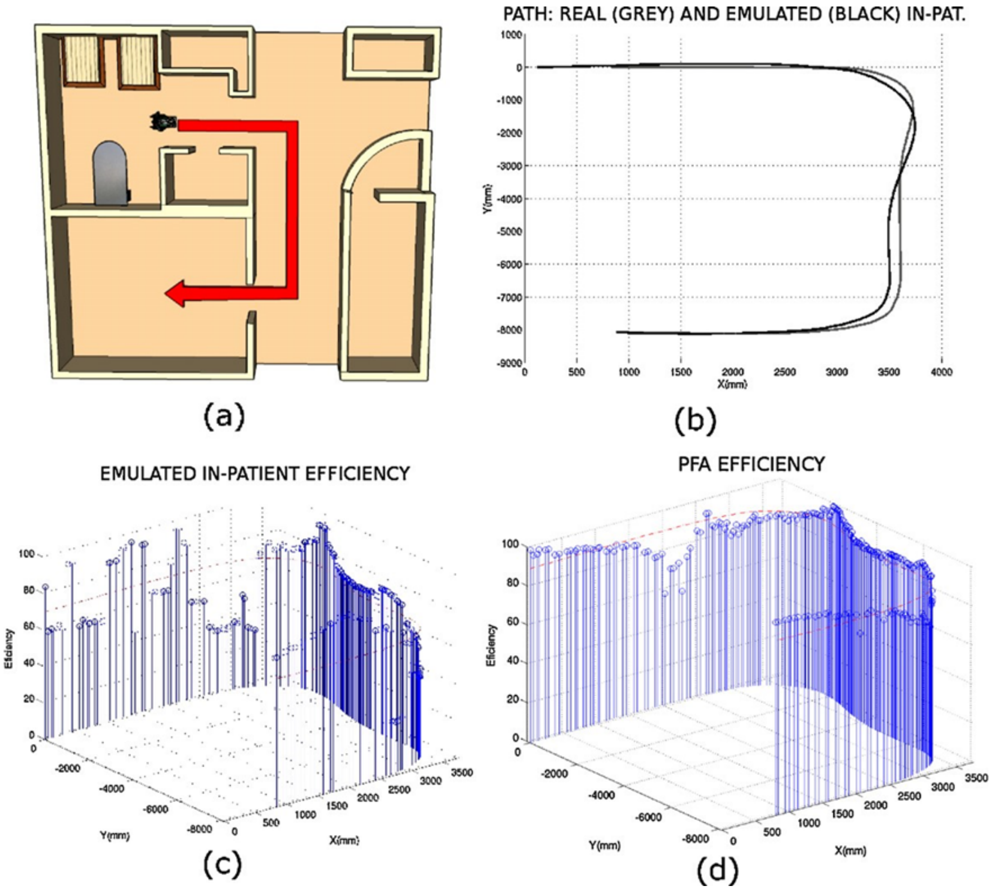


Figure 13: Path of experiment (a), PFA (grey) and emulated in-patient (black) paths (b), emulated in-patient efficiency (c) and PFA efficiency (d).

As commented, in this case there was no real experiment with which to compare results, so the experiment was repeated using PFA for comparison purposes. Fig. 13b shows the path of the emulated in-patient (black) and the PFA (grey). It can be observed that the PFA path is smoother than the emulated in-patient path and that the main differences between emulated in-patient and PFA are the turning point after crossing the door and entering the room, as expected. The maximum deviation between the two paths is equal to 21.0 cm, and the average deviation is equal to 10.07 cm. Fig. 13c shows the efficiency of the emulated in-patient. This efficiency is similar to the one from the real in-patient (in other experiments): the average efficiency is equal to 70.42% and the typical deviation is 17.60%. As in previous sections, the deviation is lower than in the real in-patient paths, as expected. Fig. 13d shows the efficiency of the PFA. It shows that variation is very low and efficiency very high (compared with the real and emulated in-patients). Therefore, the emulated in-patient behaves more similarly to a real in-patient than to an analytical algorithm such as PFA.

It must be noted, though, that similarity between emulated inpatient and PFA in this experiment is mostly due to a restricted maneuvering space, as doors are close to each other and there are few path alternatives for the user. In a larger space, it is to be expected that the user will present a larger variability depending on his/her condition and habits.

3.5. Emulated in-patient with specific disabilities

This section emulates the room-to-corridor experiment with two specific in-patients. In these experiments, most in-patient paths were similar. However, some in-patients with higher disability indexes were not able to follow the initial path correctly, so they returned along paths that were remarkably different from the average. Fig. 14a shows how an in-patient with apraxia drifts away from the usual route (especially, when he has to turn after going out of the room). Specifically, the maximum deviation in his path is 0.5 m from the typical one. Fig. 14b shows how an inpatient with dementia drifts away from the prefixed route – the maximum deviation is 0.46 m, as in the case of the in-patient with apraxia.

This experiment emulates two of the in-patients, whose paths were different from the average path to demonstrate that the emulated in-patient preserves specific behavior of different in-patients and not only the general behavior of the average in-patient. This is particularly important in this case, as there is no such thing as an average disability. Thus, every person may present a different behavior, which is captured via learning by the proposed system.

3.5.1. Test 4: emulated in-patient with apraxia

This section emulates an in-patient who suffers from apraxia. Apraxia is a neurological disorder characterized by the loss of the ability to execute or carry out learned purposeful movements, despite having the will and physical ability to perform the movements. Specifically, this in-patient has an apraxia index of 3 (0 being the worst result and 10 the best result).

Fig. 15a shows the path of the real (grey) and the emulated inpatient (black). It can be observed that the emulated in-patient path is very similar to the real one. The maximum difference between the emulated and the real in-patient is 17.5 cm and the average deviation is 8.3 cm. The efficiency of the real in-patient (Fig. 15b) and the emulated in-patient (Fig. 15c) are also shown. Both efficiencies are very similar – average efficiencies are equal to 61.11% and 62.00% – although, as commented before, the standard deviation of the real in-patient – 21.57% – is almost double that of the emulated in-patient – 12.99%.

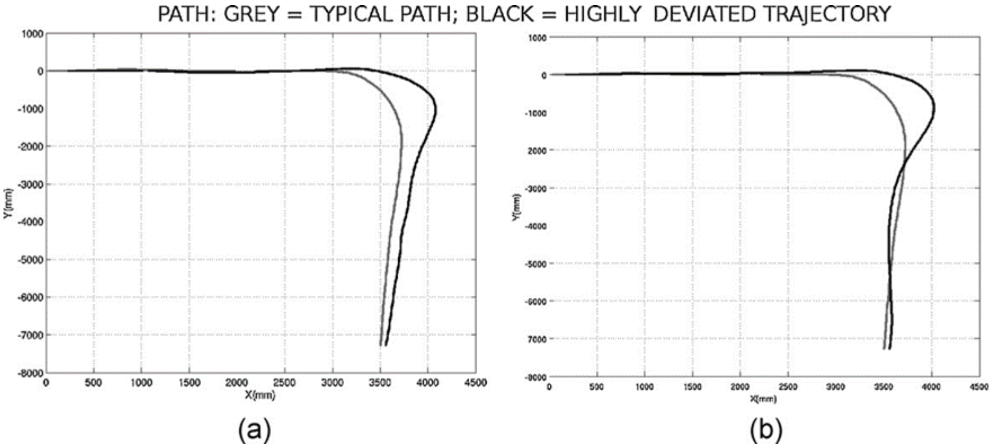


Figure 14: Path deviations with respect to the typical one in two in-patients with higher disabilities: apraxia (a) and dementia (b).

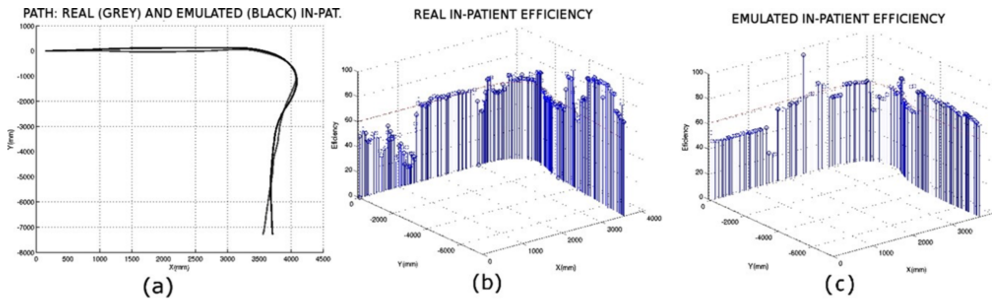


Figure 15: Path of the emulated (black) and real (grey) in-patient (a), real in-patient efficiency (b) and emulated in-patient efficiency (c). (x, y) is the position of the robot in millimeters and z (in b and c) is the motion command efficiency at each position.

3.5.2. Test 5: emulated in-patient with dementia

This section emulates an in-patient who suffers from dementia. Dementia is characterized by progressive decline in cognitive function due to damage or disease in the brain beyond what might be expected from normal aging. Concretely, this in-patient has an MMSE index of 19 (the threshold usually accepted for dementia is 24).

Fig. 16a shows the path of the real (grey) and the emulated inpatient (black). It can be observed that the emulated in-patient path is similar to the real one, although a bit worse than in the previous experiment. Specifically, there is a continuous offset between the real patient and the emulated one of about 13–15 cm. The maximum difference between the emulated and the real in-patient is 19 cm and the average deviation is 8.01 cm. The figure also shows the efficiency for the real in-patient (Fig. 16b) and the emulated inpatient (Fig. 16c). Both efficiencies are similar, although worse than in previous experiments. Specifically, the average efficiency for the emulated and real in-patient is equal to 59.60% and 70.26% respectively, and the standard deviation is equal to 7.99% and 20.78% respectively.

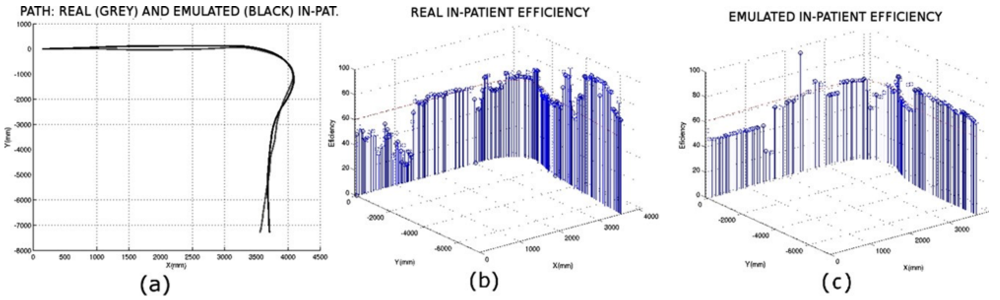


Figure 16: Path of the emulated (black) and the real (grey) in-patient (a), real in-patient efficiency (b) and emulated in-patient efficiency (c). (x, y) is the position of the robot in millimeters and z (in b and c) is the motion command efficiency at each position.

4. Conclusion and future work

This paper presents a CBR-based system to emulate wheelchair driving behaviors via learning. Specifically, the system has been tested with persons with disabilities, either physical, cognitive or both – usually the target population for a robotic wheelchair. This work is motivated by the difficulties in performing real tests with people in early stages of assistive hardware development. The simulation of a person with disabilities is not easy because human behaviors depend on a large number of variables. Emulation, though, is feasible as long as inputs and outputs can be correctly characterized in simple terms, as in reactive control systems. Specifically, wheelchair navigation has been characterized in terms of reactive sense/action pairs. Pairs for a specific user profile have been acquired via CBR learning from data available from previous tests with in-patients at FSL; in these tests in-patients drove a wheelchair in a structured indoor environment. Then, CBR is trained using this data and each person is associated with a casebase that can be used to emulate his/her driving behavior to test new systems and/or algorithms.

Thus, tests performed focus on indoor navigation in structured environments, such as the real in-patient tests. First of all, we emulate in-patients in trained trajectories (Section 3.2). The results obtained are very similar to the real ones (the maximum difference is equal to 20 cm in path and 3–4% in efficiency), even though emulated paths are, in general, smoother than the real ones. Then, we check that our approach correctly emulates a user's behavior only if data and situations learned are representative of the emulated situations (Section 3.3). However, as long as all required maneuvers are trained, we can emulate new paths in situations not present in tests at FSL (Section 3.4). In this test, an emulated in-patient was compared with PFA, since a comparison with real in-patients was not possible. An emulated in-patient turned out to be more similar to a real in-patient than to PFA. Finally, tests at FSL proved that users presenting different disabilities generate different paths. Thus, Section 3.5 shows that the emulated in-patients, based on users presenting different disabilities, also generate different behaviors and paths similar to the real ones; the maximum and mean path deviations are equal to 20 cm and 10 cm, respectively. As commented, thus far the emulated in-patients have only been tested for indoor corridor navigation and door crossing. Hence, future work will focus on extending this study to more complex environments and situations. It would also be interesting to cooperate with the medical staff at FSL to understand the meaning of the differences we observed in different in-patient case-bases by evaluating their condition and disability profile.

In brief, the proposed system could be used as an easy and rapid method to test new assistive algorithms and technology, and could be of interest for researchers, developers, users and caregivers. Specifically, it has proved to be a useful tool to emulate wheelchair driving behaviors in persons with disabilities. In our tests, the emulated in-patient turned out to be more natural and realistic than an analytical algorithm. However, it is less random than a real inpatient and requires the processing of prior data, representative of the emulated situations obtained from real experiments. However, once this data is available, it can be reused for the emulation of persons with a given disability in different environments. When dealing with different environments, determining when two situations are similar could be a problem, although this could be solved by an approximation or situation classification [58,59].

Acknowledgements

This work has been partially supported by the Spanish Ministerio de Educacion y Ciencia (TIN2005-01359 and TEC2008-06734), by FEDER funds (TEC2006-11689-C01/TCM), by the Junta de Andalucia (TIC-3991) and by the European Union in the VI Framework Program (FP6-2005-IST-5. STREP no. 045088). Authors would like to thank PhD. M.D. Roberta Annicchiarico, M.D. Alessia Federici, Jose David Fernandez-Campozo, FSL staff and all volunteer in-patients for their time and help and Sauer Medica for donating the power wheelchair used in our tests (CARMEN).

References

[1] Camarinha-Matos LM, Afsarmanesh H. A multi-agent based infrastructure to support virtual communities in elderly care. International Journal of Networking and Virtual Organisations 2004;2(3):246–66.

[2] Lothian K, Philp I. Care of older people: maintaining the dignity and autonomy of older people in the healthcare setting. BMJ 2001;322(7287):668–70.
[3] Demaerschalk B, Miley M, Kiernan T, Bobrow B, Corday D, Wellik K, et al. Stroke telemedicine. In: Mayo clinic proceedings, Mayo clinic, vol. 84. 2009. p. 53–64.
[4] Gelderblom G, de Witte L. The assessment of assistive technology outcomes effects and costs. Technology and Disability 2002;14(3):91–4.
[5] Cortés U, Annicchiarico R, Campana F, Vàzquez-Salceda J, Urdiales C, Ca namero L, et al. Intelligenza artificiale in medicina: progetto di una piattaforma mobile inserita in un ambiente intelligente per lassistenza ai disabili e agli anziani. Recenti Progressi in Medicina 2004;95(4):190–5.
[6] Agree E, Freedman V. Incorporating assistive devices into community-based long-term care: an analysis of the potential for substitution and supplementation. Journal of Aging and Health 2000;12(3):426–50.
[7] Montesano L, Diaz M, Bhaskar S, Minguez J. Towards an intelligent

wheelchair system for users with cerebral palsy. IEEE Transactions on Neural Systems and Rehabilitation Engineering 2010;18(2):193–202.

[8] Reis L, Braga R, Sousa M, Moreira A. Intellwheels mmi: a flexible interface for an intelligent wheelchair. In: RoboCup 2009: Robot Soccer World Cup XIII. Lecture notes in computer science, vol. 5949. Berlin, Heidelberg: Springer; 2010. p. 296–307.

[9] Urdiales C. Carmen (collaborative autonomous robot for mobility enhancement). Master's thesis. Spain: Technical University of Catalonia; 2010.

[10] Zeng Q, Burdet E, Rebsamen B, Teo CL. Evaluation of the collaborative wheelchair assistant system. In: IEEE 10th international conference on rehabilitation robotics, ICORR 2007. New Jersey, USA: IEEE Service Center; 2007. p. 601–8.

[11] Zeng Q, Burdet E, Teo C. Evaluation of a collaborative wheelchair system in cerebral palsy and traumatic brain injury users. Neurorehabilitation and Neural Repair 2009;23(5):494–504.

[12] Sonar A, Burdick K, Begin R, Resch E, Thompson E, Thacher E, et al. Development of a virtual reality-based power wheel chair simulator. In: Gu J, Liu PX, editors. IEEE International conference on mechatronics and automation, vol. 1. 2005. p. 222–9.

[13] Buxbaum L, Palermo M, Mastrogiovanni D, Read M, Rosenberg-Pitonyak E, Rizzo A, et al. Assessment of spatial neglect with a virtual wheelchair navigation task. In: Burdea G, editor. International workshop on virtual rehabilitation. 2006. p. 94–9.

[14] Grant M, Harrison C, Conway B. Wheelchair simulation. In: Cambridge workshop series on universal access and assistive technology. Springer-Verlag; 2004. p. 101–109.

[15] Dicianno B, Sibenaller S, Kimmich C, Cooper R, Pyo J. Joystick use for virtual power wheelchair driving in individuals with tremor: pilot study. The Journal of Rehabilitation Research and Development 2009;46(2):269–76.

[16] Braga R, Malheiro P, Reis L. Development of a realistic simulator for robotic intelligent wheelchairs in a hospital environment. In: RoboCup 2009: Robot Soccer World Cup XIII. Lecture notes in computer science, vol. 5949. Berlin, Heidelberg: Springer; 2010. p. 23–34.

[17] Dvorznak MJ, Cooper RA, O'Connor TJ, Boninger ML, Fitzgerald SG. Kinematic comparison of hybrid ii test dummy to wheelchair user. Medical Engineering & Physics 2001;23(4):239–47.

[18] Mataric MJ. Interaction and intelligent behavior. Master's thesis. EECS: MIT; 1994.

[19] Herrero I, Urdiales C, Peula JM, Sanchez-Tato I, Sandoval F. A guided learning strategy for vision based navigation of 4-legged robots. AI Communications 2006;19(2):127–36.

[20] Cucker F, Smale S. Emergent behavior in flocks. IEEE Transactions on Automatic Control 2007;52(5):852–62.

[21] Mogul JC. Emergent (mis)behavior vs. complex software systems. SIGOPS Operating Systems Review 2006;40(4):293–304.

[22] Mataric MJ. Behaviour-based control: examples from navigation learning and group behaviour. Journal of Experimental & Theoretical Artificial Intelligence 1997;9(2–3):323–36.

[23] Arkin RC. Integrating behavioral perceptual and world knowledge in reactive navigation. Robotics and Autonomous Systems 1990;6(1–2):105–22.

[24] Hu H, Brady M. A parallel processing architecture for sensor-based control of intelligent mobile robots. Robotics and Autonomous Systems 1996;17(4):235–57.

[25] Brooks RA. Intelligence without representation. Artificial Intelligence 1991;47(1–3):139–59.

[26] Lyons DM, Hendriks AJ. Reactive planning. In: Shapiro SC, editor. Encyclopedia of artificial intelligence. 2nd ed. New York, USA: John Wiley & Sons, Inc.; 1992. p. 1171–81.

[27] Franz MO, Mallot HA. Biomimetic robot navigation. Robotics and Autonomous Systems 2000;30(1–2):133–53.

[28] Codognet P. Behaviors for virtual creatures by constraint-based adaptive search. In: Laird J, van Lent M, editors. Working notes of the AAAI spring symposium on artificial intelligence and interactive entertainment. CA, USA: AAAI Press/The MIT Press; 2001. p. 25–30.

[29] Jelonek J, Komosinski M. Biologically-inspired visual-motor coordination model in a navigation problem. In: Gabrys B, Howlett R, Jain L, editors. Knowledge-based intelligent information and engineering systems. Lecture notes in computer science, vol. 4253. Berlin, Heidelberg: Springer; 2006. p. 341–8.

[30] Wei R, Austin D, Mahony R. Biomimetic application of desert ant visual navigation for mobile robot docking with weighted landmarks. International Journal of Intelligent Systems Technologies and Applications 2005;1(1/2): 174–90.

[31] Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. The International Journal of Robotics Research 1986;5(1):90–8.

[32] Aamodt A, Plaza E. Case-based reasoning: foundational issues method-

ological variations and system approaches. AI Communications 1994;7(1): 39–59.
[33] Schank RC, Abelson RP. In: L. Erlbaum Associates, editor. Scripts plans goals and understanding: an inquiry into human knowledge structures (artificial intelligence series). 1st ed. New Jersey, USA: Psychology Press; 1977.
[34] Aamodt A. Explanation-driven case-based reasoning. In: Wess S, Althoff K-D, Richter M, editors. Topics in case-based reasoning. Lecture notes in computer science, vol. 837. Berlin, Heidelberg, Germany: Springer; 1994. p. 274–88.
[35] Schank RC. In: Schank RC, editor. Dynamic Memory: a theory of reminding and learning in computers and people. 1st ed. New York, USA: Cambridge University Press; 1983.
[36] Kolodner JL. An introduction to case-based reasoning. Artificial Intelligence Review 1992;6(1):3–34.
[37] Kolodner JL, Kolodner RM. Using experience in clinical problem solving: introduction and framework. IEEE Transactions on Systems Man and Cybernetics 1987;17(3):420–31.
[38] Bichindaritz I, Marling C. Case-based reasoning in the health sciences: what's next? Artificial Intelligence in Medicine, Special Issue on Case-based Reasoning in the Health Sciences 2006;36(2):127–35.
[39] Bichindaritz I. Case-based reasoning in the health sciences: why it matters for the health sciences and for cbr. In: Althoff K-D, Bergmann R, Minor M, Hanft A, editors. Advances in case-based reasoning. Lecture notes in computer science, vol. 5. Berlin, Heidelberg, Germany: Springer; 2008. p. 1–17.
[40] Holt A, Bichindaritz I, Schmidt R, Perner P. Medical applications in case-based reasoning. The Knowledge Engineering Review 2005;20(3):289–92.
[41] Nilsson M, Sollenborn M. Advancements and trends in medical case-based reasoning: an overview of systems and system development. In: Russell I, editor. Proceedings of the 17th international FLAIRS conference, Special track on case-based reasoning. CA, USA: AAAI Press; 2004. p. 178–83.
[42] Perner P, Jänichen S, Perner H. Case-based object recognition for airborne fungi recognition. Artificial Intelligence in Medicine 2006;36(2): 137–57.
[43] Perner P. An architecture for a cbr image segmentation system. In: Althoff K-D, Bergmann R, Branting L, editors. Case-based reasoning research and development. Lecture notes in computer science, vol. 1650. Berlin, Heidelberg, London, UK: Springer; 1999. p. 525–34.

[44] Fenstermacher KD. Cadi: an intelligent, multimedia tutor for cardiac auscultation. In: Buchanan B, Musen M, Szolovits P, editors. Proceedings of the thirteenth national conference on artificial intelligence. AAAI '96, vol. 2. CA, USA: AAAI Press/The MIT Press; 1996. p. 1387.
[45] Davis G, Wiratunga N, Taylor B, Craw S. Matching smarthouse technology to needs of the elderly and disabled. In: McGinty L, editor. Proceedings of the 5th international conference on case based reasoning workshop. 2003. p. 29–36.
[46] Liu H, Iba H. A layered control architecture for humanoid robot. In: Mukhopadhyay S, Gupta GS, editors. Proceedings of international conference on autonomous robots and agents. Palmerston North, NZ: Massey University; 2004. p. 434–9.
[47] Lee J, Likhachev M, Arkin R. Selection of behavioral parameters: integration of discontinuous switching via case-based reasoning with continuous adaptation via learning momentum. In: Hamel W, Maciejewski A, editors. IEEE international conference on robotics and automation. Proceedings. ICRA '02, vol. 2. 2002. p. 1275–81.
[48] Branting LK, Aha DW. Stratified case-based reasoning: reusing hierarchical problem solving episodes. In: Mellish CS, editor. Proceedings of the 14th international joint conference on artificial intelligence. IJCAI, vol. 1. CA, USA: Morgan Kaufmann Publishers Inc.; 1995. p. 384–90.
[49] Kruusmaa M. Global navigation in dynamic environments using case-based reasoning. Autonomous Robots 2003;14(1):71–91.
[50] Likhachev M, Arkin R. Spatio-temporal case-based reasoning for behavioral selection. In: Kwon W, Lee BH, editors. IEEE international conference on robotics and automation, 2001. Proceedings. ICRA '01, vol. 2. 2001. p. 1627–34.
[51] Ram A, Santamaria J. A multistrategy case-based and reinforcement learning approach to self-improving reactive control systems for autonomous robotic navigation. In: Michalski RS, Tecuci G, editors. Proceedings of the second international workshop on multistrategy learning. 1993.
[52] Haigh K, Veloso M. Route planning by analogy. In: Veloso M, Aamodt A, editors. Case-based reasoning research and development. Lecture notes in computer science, vol. 1010. Berlin, Heidelberg: Springer; 1995. p. 169–80.
[53] Urdiales C, Perez EJ, Sandoval F. A time stamp control strategy for cbr based reactive navigation in dynamic environments with priorities. In: Zhong N, Bradshaw J, Pal S, Talia D, Liu J, Cercone N, editors. IEEE/WIC/ACM international conference on intelligent agent technology, 2004. Proceedings.

IAT 2004. 2004. p. 58–64.
[54] Sanchez-Tato I, del Toro JC, Perez EJ, Urdiales C, Cortes U, Annicchiaricco R, et al. Collaborative control for personalized mobility assistance to persons with physical/cognitive disabilities. In: Grant E, Henderson T, editors. Proceedings of the IEEE/RSJ international conference on intelligent robots and systems. 2007.
[55] Urdiales C, Perez E, Vázquez-Salceda J, Sànchez-Marrè M, Sandoval F. A purely reactive navigation scheme for dynamic environments using case-based reasoning. Autonomous Robots 2006;21:65–78.
[56] Crum R, Anthony J, Bassett S, Folstein M. Population-based norms for the minimental state examination by age and educational level. JAMA: The Journal of the American Medical Association 1993;269(18):2386–91.
[57] Mahoney FI, Barthel DW. Funcional evaluation: the barthel index. Maryland State Medical Journal 1965;14:61–5.
[58] Minguez J, Osuna J, Montano L. A “divide and conquer” strategy based on situations to achieve reactive collision avoidance in troublesome scenarios. In: Tarn TJ, Fukuda T, editors. IEEE international conference on robotics and automation, 2004. Proceedings. ICRA’04, vol. 4. 2004. p. 3855–62.
[59] Peinado G, Urdiales C, Peula J, Fdez-Carmona M, Annicchiaricco R, Sandoval F, et al. Navigation skills based profiling for collaborative wheelchair control. In: Li Z, editor. IEEE international conference on robotics and automation (ICRA). 2011. p. 2229–34.