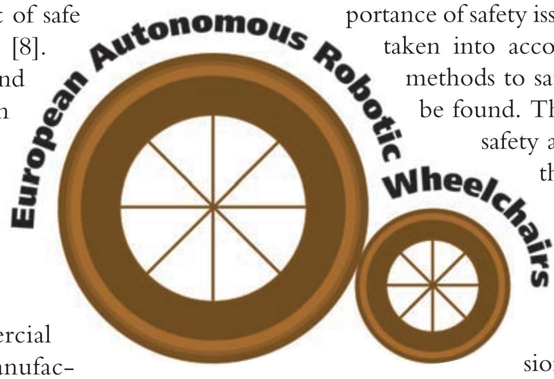
Lankenau, A., Röfer, T. (2001). A Safe and Versatile Mobility Assistant. In: IEEE Robotics and Automation Magazine 7, No. 1, March 2001. 29-37.

A Versatile and Safe Mobility Assistant

The Bremen Autonomous Wheelchair Implements Obstacle Avoidance Plus Driving and Routing Assistance in a Shared-Control System

by AXEL LANKENAU and THOMAS RÖFER

The *Bremen Autonomous Wheelchair* project covers a variety of research issues that include *spatial cognition*, *safe systems*, and *mobility assistance for the handicapped*. Emphasizing one or the other of these cornerstones provides various views of the wheelchair *Rolland* (Fig. 1). Within the framework of the priority program "Spatial Cognition" of the Deutsche Forschungsgemeinschaft, the main topics are navigation, landmark recognition, and space representation [7]. Moreover, the wheelchair serves as a case study for the development of safe systems based on formal methods [8]. The third determinant of the Rolland project is the aspect of rehabilitation [13]; i.e., to develop a support tool that can help handicapped and elder people to control a wheelchair. These seemingly divergent topics complement each other to build a robust and safe wheelchair system.



Rolland is based on the commercial power wheelchair *Genius 1.522* manufactured by the German company Meyra. The wheelchair is a nonholonomic vehicle that is driven by its front axle and steered by its rear axle. The human operator can control the system with a joystick. The wheelchair is equipped with a standard PC (Pentium 233, 64 MB RAM) and sensors to perceive its environment. The principal idea behind such an assistive system is to wiretap the connection between the joystick and the motor. If the human operator issues a command that may lead to a collision with an obstacle, the wheelchair changes the dangerous target command into a safe one.

This article shows how the tasks of reliably detecting obstacles in the environment and safely avoiding these obstructions are solved in the Bremen Autonomous Wheelchair. In addition, it deals with the problems that arise due to the fact that, in contrast to most research robots, the wheelchair is a shared-control system; i.e., it is jointly controlled by the software modules and the human operator. This is primarily important for the obstacle avoidance skill as well as for the mobility assistant.

Safety Layer

Being able to ensure safe traveling of an assistive power wheelchair requires several preconditions to be met. First, the importance of safety issues has to be accepted and rigorously taken into account during system design. Second, methods to satisfy the safety requirements have to be found. This section motivates the meaning of safety aspects in service robots and presents the main components of the safety layer of the Bremen Autonomous Wheelchair—a local obstacle map that represents measurements of range sensors, and the way this representation is used to ensure a collision-free operation of the system.

Service Robots as Safety-Critical Systems

When considering systems engineering tasks in the context of avionics or railway interlocking systems, the use of formal methods in the design process is becoming state of the art. Due to the fact that a malfunction of such applications may cause severe harm to human beings or result in other catastrophic consequences, these systems are referred to as *safety-critical* [11,17]. As the potential damage robot actions can cause to human beings increases with the spatial proximity in which man and machine operate, service and especially rehabilitation robots have to be classified as safety-critical systems, too.

To ensure a dependable behavior, the system architecture of the Bremen Autonomous Wheelchair was developed with the help of formal methods, as is described in [8, 9, 10, 13]. As a result of a fault-tree-based hazard analysis [10], 31 safety require-

 1070-9932/01/$10.00©2001IEEE

ments were specified [9]. Such safety requirements have to hold during operation time. If one of them is violated, the correct behavior of the system cannot be guaranteed any longer.

A static strategy is unsuitable to acquire readings for a map that is employed for collision avoidance.

As an example, consider the requirement *distances to obstacles will never be overestimated*. Note that the notion "overestimation" includes the nonrecognition of an object. The wheelchair perceives its environment with the help of a ring of ultrasonic sensors (i.e., range detectors). To implement the given requirement, it is insufficient to consider only the current sonar readings because such a sensor system does not give a complete snapshot of the environment. Instead, it delivers pairs of readings in a sequence over a longer period of time (approximately 0.5 s). Thus, if the wheelchair moves at maximum speed (84 cm/s), the vehicle could cover a distance of 42 cm before all measurements are taken. To avoid phases of "blind" traveling, recent measurements are temporarily stored in the so-called *local obstacle map*.

Local Obstacle Map

The local obstacle map or *occupancy grid* [5] is a quadratic array of cells, each of which encodes—among other things —the presence of an obstacle at the corresponding position in the environment of the wheelchair (Fig. 3). Due to computational considerations, the position of the wheelchair relative to the map is static with respect to translational movements and dynamic with respect to rotational changes. This means that the wheelchair's $(x\ /\ y)$-position (i.e., the middle of the front axle), constantly remains in the center of the map, whereas the orientation of the wheelchair relative to the map changes during operation. As a consequence of the wheelchair's static position at the map center, the representation of the environment (i.e., the positions of obstacles) has to be shifted according to the current movement of the vehicle that can be determined from the wheelchair's dead-reckoning system. Since the map is represented as an array in memory, the shifts in x and y direction are computationally cheap. In contrast to that, a rotation would be rather time consuming and prone to discretization errors. Therefore, instead of rotating the map, the wheelchair's current rotational deviation to the map is continuously updated. As the map is relatively small in comparison to the wheelchair's speed, odometry errors do not have a negative impact on the consistency of the readings in the map. Erroneous entries will leave the map shortly after they were measured.

Fig. 1. The Bremen Autonomous Wheelchair "Rolland" is a standard product enhanced with sonar sensors and a PC.

To be able to stop in time using the map, its extent from the center to any border must be at least the sum of the following distances:

- The wheelchair's extent from the map's center. When driving forward, this is the wheelchair's length in front of the driving axle (i.e., 40 cm), otherwise it is the length of the back (i.e., 80 cm).
- A minimum range between the obstacles and the wheelchair. The sonar sensors cannot measure ranges shorter than 15 cm. Therefore, the vehicle will stop in time, keeping this distance to travel before a collision occurs.
- The overall stopping distance. For the wheelchair's maximum speed of 84 cm/s, this is 53 cm. In backward direction, the vehicle can only travel up to 42 cm/s, so the stopping distance is shorter (i.e., 18 cm).
- The distance the wheelchair can cover until all relevant readings have been inserted into the map. In the worst case, all sensors must have been fired, which takes approximately 0.5 s. Again, it must be distinguished between forward motion (42 cm) and backward motion (24 cm).

So, although the wheelchair's back is longer than its front, the required map size is maximal for forward motion (2×150 cm), because the vehicle's maximum speed is much smaller in backward direction, and thus the backward overall stopping distance is significantly shorter, too. As the map is also used for another application that smooths the acceleration and deceleration of the wheelchair (see below), its size is larger: 402×402 cm^2. In order to be sufficiently precise to allow, for example, driving through doorframes, the map consists of 17956 3×3 cm^2 cells.

As mentioned, the local obstacle map is the "reference" source of information when the safety layer decides whether or not the driving command set by higher-level applications (or the human operator) can be safely executed. This implies that the map has to contain all relevant data at any point in time. Interestingly, this does not mean that *any* obstacle in the surroundings of the wheelchair has to be represented in the map. Instead, only those obstacles that are *relevant* for the wheelchair in a certain situation have to be considered.

Adaptive Sonar Firing Strategy

The Bremen Autonomous Wheelchair perceives its surroundings with 27 ultrasonic sensors. Usually, measurements of different sonars are taken in sequence to reduce the probability of misreadings typical for this kind of sensor [14]. Such sequences are often referred to as *firing algorithms* [2] or *firing strategies*. As pointed out in [14], the use of a static strategy for firing the ultrasonic sensors involves the risk of overlooking relevant obstacles. For instance, the Bremen Autonomous Wheelchair can only fire two of its 27 sensors simultaneously. Thus, due to the sensor arrangement, the wheelchair would acquire a continuous scan of the environment after 14 sonar measurement cycles if it did not move simultaneously. However, things change when the wheelchair is moving. If, for example, the vehicle drives a narrow curve to the right, as depicted in Fig. 2, the readings give only a holey image of the surroundings, because the positions where readings were taken are shifted by the wheelchair's motion. Thus, there is a risk of overlooking obstacles when using such a static firing strategy, not only for the Bremen Autonomous Wheelchair but also for other mobile robots with sonar sensors [e.g., the *Nomad 200* manufactured by Nomadic (used e.g., in [12]) or RWI's *B21* (used, e.g., in [4])].

As such, a static strategy is unsuitable to acquire readings for a map that is employed for collision avoidance, a dynamic strategy that determines the fire sequence from the wheelchair's current movement direction, its steering angle, and the data already represented in the map is used for Rolland. Two mechanisms were implemented to measure the relevant parts of the environment as fast as possible, and to enter the sensor readings into the map.

RELEVANT AREA

The wheelchair uses only the sensors that "see" the areas of the environment that are relevant for the collision avoidance in the current situation; i.e., the areas the wheelchair could reach in the near future (see below). If this area contains previously recognized obstacles, only those cells are interesting that are closer than the nearest known obstacle, because the vehicle already has to stop in front of this closest obstacle, and therefore farther objects are not relevant for stopping in time. In surroundings that are cluttered with obstacles, this further reduces the number of cells to be measured, and thus the number of sonars to be fired. So, the measurement update rate increases when the environment gets narrow. This is a desirable feature for a reactive system.

The basic idea of the proposed obstacle avoidance method is to detour obstacles in a way that is most likely to be acceptable for the operator.

MEASUREMENT AGE

To reach a uniform coverage of measurements in the relevant area, the age of the last measurement is stored for each cell in the map (Fig. 3). Thereby, the sensors can be fired that are orientated toward the cells that have not been measured for the longest period of time. Obstacles are only noticed by the collision avoidance when they were measured at least twice, so as to reduce the effect of so-called cross-talks [3]. To nevertheless ensure fast obstacle recognition, the second detection is forced by setting the age of all cells to "very old" that were measured as occupied for the first time. Thus, these areas are immediately re-measured, either confirming their occupancy or rejecting it. Cells that are "scrolled in" from the borders of the map due to the wheelchair's movement, and thus have never been measured before, are assumed to be "old" and—for safety reasons—are treated as obstacles until they are measured for the first time. During each measurement cycle, all cells age by one until they are re-measured or they reach the "old" state. Note that the "very old" state is

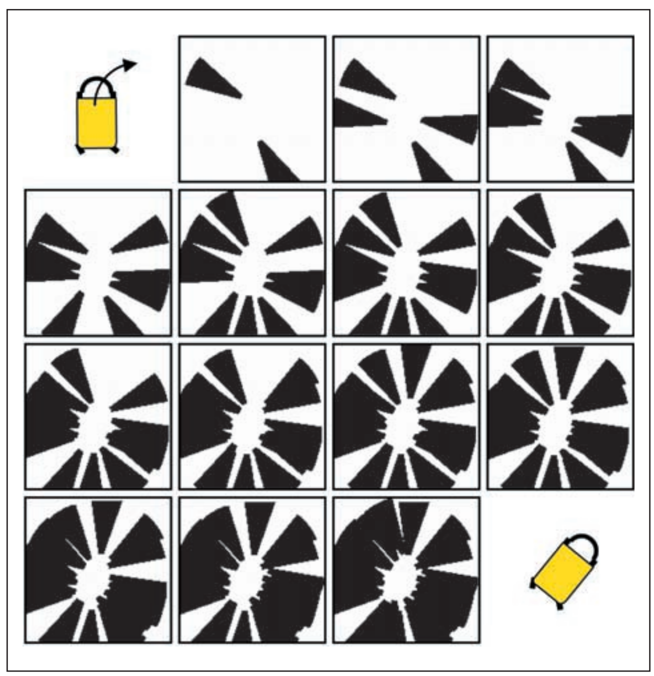


Fig. 2. A static sonar firing strategy. Black color marks the areas measured by the sonar sensors during a complete firing cycle of 14 steps while the wheelchair drives a narrow right curve [14].

reserved for forcing certain measurements; i.e., no cell can reach this state by aging.

RESULT

Figure 4 shows a result of the adaptive firing strategy. As in Fig. 2, the wheelchair follows a narrow curve to the right, but the firing strategy differs heavily: on the one hand, only those sensors are used that look toward the front of the vehicle and toward the outside of the curve; i.e., the areas the wheelchair may visit while driving the curve. On the other hand, this relevant environment is measured very quickly. After six measurement cycles, the spatial coverage is nearly perfect. The hole on the left side is not important at this time, because the wheelchair will not reach this area with its front. Only when the vehicle has turned further will it contact the region with its back; therefore, it is measured then.

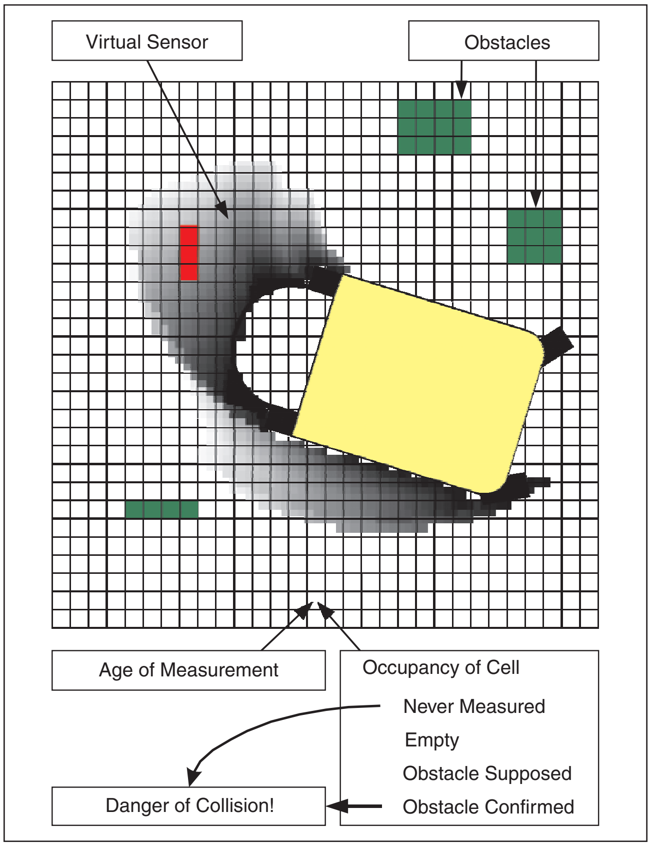


Fig. 3. *The local obstacle map and a virtual sensor. The virtual sensor comprises the cells the wheelchair will "visit" when driving with the steering angle indicated by the back wheels. The darker the cells, the earlier this contact will happen.*

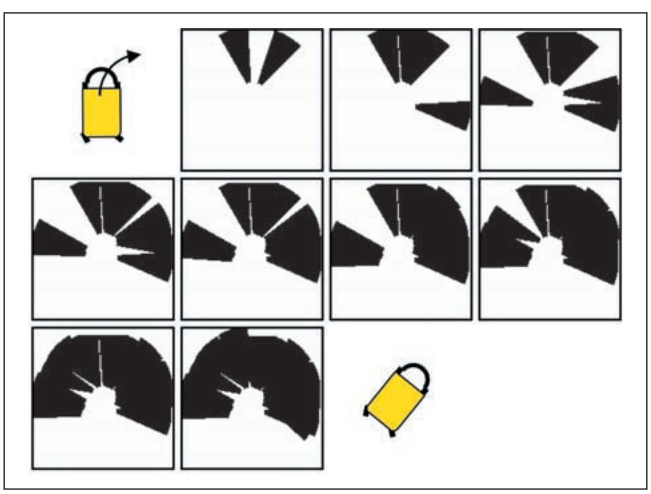


Fig. 4. *The adaptive sonar firing strategy. The relevant parts of the environment are measured faster than in Fig. 2 [14].*

Stopping in Time

To be able to determine when it is necessary to brake the wheelchair, the following things must be taken into account:

- The wheelchair's deceleration and its reaction time. Together with the actual speed, these values determine the vehicle's overall stopping distance.
- The *travel distance* to the closest obstacle. This is the distance the wheelchair can travel with its current driving direction and its actual steering angle before it hits the first obstacle.

Whereas it is a simple operation to calculate the overall stopping distance for a certain speed, the travel distance to the closest obstacle cannot be determined straightforwardly. The wheelchair's motion is nonholonomic and its body has a complex shape. In order to be able to calculate the travel distance to the closest obstacle, so-called *virtual sensors* are used to access the local obstacle map. In order to decide whether an obstacle represented by an entry in the map may potentially cause a collision with the wheelchair, the vehicle's trajectory must be anticipated. This is a rather complex task for a wheelchair with a driving axle and a steering axle, because parts of the body swing out sideways when driving curves. Steering even changes the shape of the system's body, because the turned wheels can stick out sideways. Other approaches try to reduce the computational costs of the anticipation of the trajectory by simplifying the shape of the region that has to be searched for obstacles (e.g., [6]). In contrast, virtual sensors are an exact method (Fig. 3).

For each combination of driving direction, steering angle, and orientation of the wheelchair, the vehicle's movement was precalculated. Using the robot simulator SimRobot [15], the cells of the map that would be visited by the wheelchair were determined. The travel distance to be checked for each direction/angle/orientation-combination is prescribed by the parameters of the wheelchair's braking distance, its response delay to commands, and its minimum safety distance to surrounding objects. In real-time systems, it is a common approach to sacrifice the resource of memory for the sake of saving operation time. As a result of the simulated movements, the algorithm knows in each situation which cells in the local obstacle map have to be checked for objects and how far the wheelchair has to drive to reach a cell. Since this is a means to determine the distance to obstacles that works on an internal representation of the environment, it is called the *virtual sensor* technique.

The shape of a virtual sensor is not only determined by the three variables of driving direction, steering angle, and orientation of the wheelchair but also by various constants that have to be considered during the simulation: the stopping distance, the potential change of the steering angle in time, and quantization problems when mapping the shape of the virtual sensor into the grid of the map. Thus, each virtual sensor anticipates the wheelchair's worst-case movement for a range of rotations and steering angles to retain safety despite the loss of precision due to the use of a discrete map. Using the virtual sensor technique, the safety layer provides a safe interface to the real wheelchair. It is safe in that the continuous update of the local obstacle map and the processing of the travel distance to the closest obstacle ensure that collisions are averted.

Elementary Skills

A hierarchy of elementary skills was implemented on top of the safety layer in order to allow higher-level applications to rely on certain basic features of the wheelchair such as smooth speed control or obstacle avoidance.

Smooth Speed Control

This skill realizes a smooth braking behavior. As mentioned above, the safety layer is able to recognize dangerous objects early enough so that it manages to decelerate the wheelchair to a standstill in time even from maximum speed. Since a smooth braking maneuver takes more time than an emergency braking, the speed control has to intervene much earlier than the collision avoidance does. To decide whether or not the deceleration is to be activated, the virtual sensors already used by the safety layer are employed to determine the distance to the closest obstacle. Similar to other approaches (e.g., [1]), the module sets the target speed to a value half as large as required to stop in time. This way, the wheelchair's locomotion becomes rather smooth and comfortable.

This behavior is useful for all users of a wheelchair because they can always choose the maximum speed with the joystick and can concentrate on steering, while the vehicle independently selects an adequate velocity with respect to the current obstacle situation. Also, the speed control facilitates the realization of other applications: Since a sonar reading only provides the information that there is an obstacle in a certain distance within the opening angle of the sensor, objects that have been detected early are represented relatively large in the local obstacle map. Due to safety (i.e., worst-case) considerations, no heuristics can be employed in order to tailor down the size of detected obstacles. When the wheelchair approaches such an obstacle, its representation in the map also comes closer; i.e., the distance returned by the virtual sensors decreases. If the wheelchair drives at full speed, this approach happens rather fast. As a consequence, a possible decision to intervene with an emergency brake depends on the old and possibly overestimated size of the obstacle but not on its real size. If the wheelchair drives slowly, more sensors are fired per driven distance. So, the update rate for the relevant area increases, resulting in a more precise representation of the real world in the map. Thus, other application modules that are responsible for narrow-space maneuvering (e.g., the obstacle avoidance skill presented in the following section) profit from the smooth speed control.

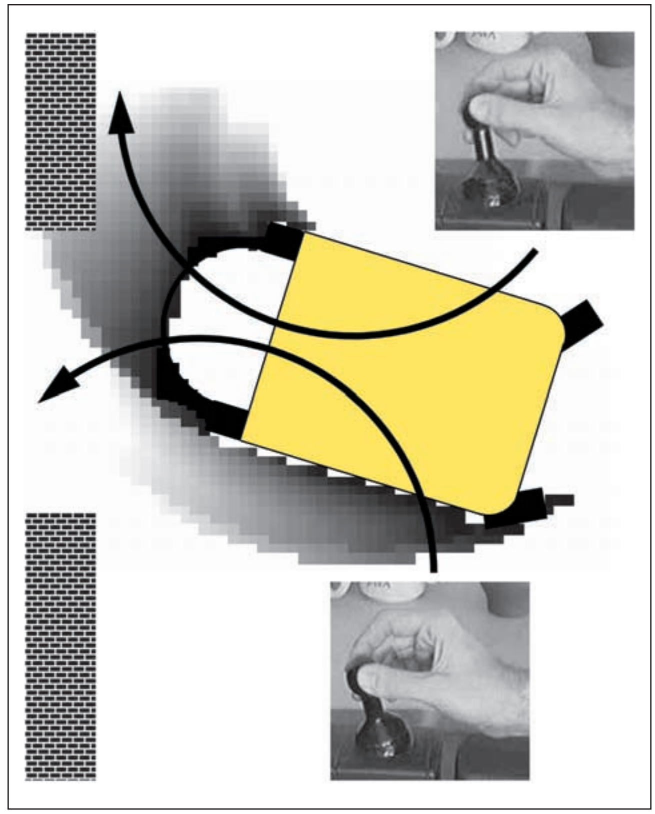


Fig. 5. *Deciding on which side the user wants the obstacle to be passed.*

Obstacle Avoidance

The basic idea of the proposed obstacle avoidance method is to detour obstacles in a way that is most likely to be acceptable for the operator. By taking into account the desired traveling direction in terms of the radius indicated by the operator via the joystick, it is decided whether the wheelchair should steer to the right or to the left. That way, the requirements of the wheelchair as a shared-control system [14] are adequately met.

A typical situation is depicted in Fig. 5. The wheelchair is placed in front of a doorway. If it simply drove forward, it would hit into the right door post. If the operator indicates a

Table 1. Avoid Directions When Driving Forward

Obstacle Position	Indicated Direction		
	Left	None	Right
Front, left	Left	Straight	Right
Front, right	Left	Straight	Right
Rear, left	—	Straight	Left
Rear, right	Right	Straight	—

right turn (as shown in the upper-right picture), the assistive system infers that he or she wants to pass the obstacle (i.e., the right door post) on the right-hand side. Thus, it reinforces its

Depending on the specific needs of the user, various levels of support, so-called assistants, can be activated.

steering angle to the right. If, instead, the operator indicates a left curve (as shown in the lower right picture), the assistive system takes it for granted that the operator wants to pass the obstacle on the left-hand side; i.e., to pass through the doorway.

In a former approach, a rather machine-oriented obstacle avoidance behavior [8] was implemented on the Bremen Autonomous Wheelchair. Similar to the vector field histogram (VFH) method [1], the best steering angle was chosen by searching the local obstacle map presented above for obstacle-free regions. Even though some enhancements in comparison to Borenstein's VFH method were implemented, this approach suffered from the fact that the success of an obstacle avoidance maneuver heavily depended on the time of intervention. In addition, the wheelchair often took control in situations in which the user did not want to avoid an obstacle; e.g., when approaching a wall to maximally exploit the available space for a 180° turn.

Table 2. Avoid Directions When Driving Backward

Obstacle Position	Indicated Direction		
	Left	None	Right
Front, left	—	Straight	—
Front, right	—	Straight	—
Rear, left	Left	Straight	Right
Rear, right	Left	Straight	Right

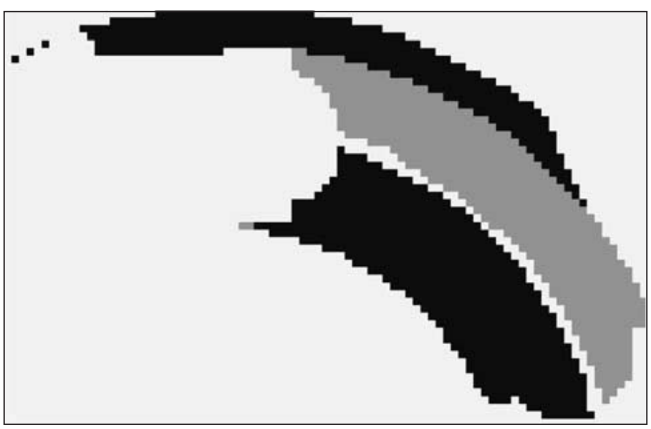


Fig. 6. Decision diagram for obstacle avoidance during a right curve. Objects in the black area are detoured toward the left, those in the gray area toward the right. Obstacles in the white parts are not avoided (the user remains in control).

To overcome these limitations, the obstacle avoidance approach was enhanced: to determine the avoidance maneuver that matches the intention of the human operator best, the direction he or she indicates with the joystick is considered. The module continuously searches the local map for obstacles based on this direction by employing the corresponding virtual sensor. If the wheelchair is already detouring, the direction indicated by the user may deviate from the system's current steering direction. Thus, the obstacle avoidance module always assesses the world from the user's point of view, and this view includes the judgement whether an obstacle is on the left or on the right side of the intended driving direction.

To allow the avoidance module to reconstruct this assessment, for each cell in a virtual sensor it was preprocessed whether it is better to avoid an obstacle at the corresponding real-world position on the left side or on the right side. For instance, if the wheelchair hit the obstacle with its front left, the vehicle would try to avoid it on the right side. In contrast, if a collision was expected on the left side behind the front axle, the wheelchair should also turn to the left. So, the avoidance direction depends on the obstacle's position relative to the center of the wheelchair's front axle. The different cases are listed in Table 1 for forward movements and in Table 2 for backward motion. Using these decision tables, the virtual sensors can be enriched by the appropriate avoidance directions for each cell in the covered area. Figure 6 shows an example of a resulting decision diagram. Note that if the indicated course directly points towards an object, the obstacle avoidance module will not intervene at all. This is because in such a case it is likely that the operator's intention was to move close to that obstacle and not to pass around it; e.g., when docking to a desk.

As mentioned above, in common obstacle avoidance approaches it is a question of parameter optimization and intuition when the robot decides to detour a detected obstacle. The Bremen Autonomous Wheelchair pursues another strategy: in every operation step, the maximum speed v_{mc} that does not lead to a collision with the detected closest obstacle is determined. As a result, reducing the current speed to or below v_{mc} will avoid the collision. v_{mc} is not computed during program execution, but it is read out of a precalculated map, as depicted in Fig. 7. This map represents a lookup table for the function

$$v_{mc}(pos_{obs}, \alpha, dir)$$

where pos_{obs} is the position of the obstacle relative to the wheelchair, α is the current steering angle, and dir encodes whether the wheelchair tries to pass the obstacle on the left or on the right side. The lookup table was calculated in advance by the simulation SimRobot already mentioned above.

The obstacle avoidance skill was successfully tested in several situations; e.g., in narrow corridors and while passing doorways. As an example, Fig. 8 visualizes a forward drive through a 92 cm-wide doorway. Each of the depicted outlines corresponds to the position of the wheelchair in 0.5 s steps. During the experiment, the user only gave a rough driving direction towards the door. The fine navigation was performed by the obstacle avoidance skill. The wheelchair started driving with maximum speed toward the door, slowed down, centered itself between the door-posts, passed the doorway, and reaccelerated. In the run shown in Fig. 8, it additionally avoided the side wall of the room behind the doorway.

Turning Around

In contrast to common omni-directional research robots, the Bremen Autonomous Wheelchair is a kinematically restricted vehicle that cannot perform sideways movements, and, particularly, cannot turn on the spot. Similar to a car driver, the user of the wheelchair has to accomplish subtle shunting maneuvers in order to turn the orientation of the vehicle by 180°. As these maneuvers comprise difficult backward movements, it is quite helpful for the user if the wheelchair is able to turn automatically.

Most daily-life situations in which an automatic turning behavior is required can be solved driving on a simple trajectory that consists of two quarter circles with opposite steering angles and driving directions. To explain the algorithm, the easy task of turning in open space is shown in Fig. 9. The turning maneuver consists of two parts: first, the wheelchair drives on a narrow curve forward to the left [Fig. 9(a)] until it reaches an orientation of about 90° with respect to its original position. The second phase is performed backward in a right curve [Fig. 9(b)]. The wheelchair declares the maneuver to be successful when its orientation changed by 180°.

Whereas turning in open space is not very difficult, matters change in cluttered environments or in corners as shown in Fig. 10. In the initial situation depicted in Fig. 10(a), the wheelchair cannot perform any of the four theoretically available quarter circles. This is because the wall in the back (i.e., left in the figure) prevents the vehicle from driving backward. Driving forward in a narrow left curve is also not possible, because the right-hand back of the vehicle would collide with the wall in the lower part of the figure. So, the wheelchair decides to turn to the forward right, because there is an entrance of a room providing a little bit of maneuvering space. Even if this entrance was not present, the system would choose this direction, because its narrower front always allows to drive a short curve. The obstacle avoidance module beneath the turning behavior pays attention inasmuch as it “softens” the curve in order not to hit into the wall. As a result, the wheelchair follows an almost straight path parallel to the wall. After a while, there is enough space to accomplish a quarter circle backward to the left [Fig. 10(b)]. Finally, turning forward to the right is the successful movement that brings the wheelchair to the target orientation. Note that the intention of the turning behavior is to change the orientation of the wheelchair by 180° with minimum shunting effort. For the convenience of the human user, it is *not* the goal to exactly turn on the spot and to precisely reach the original position.

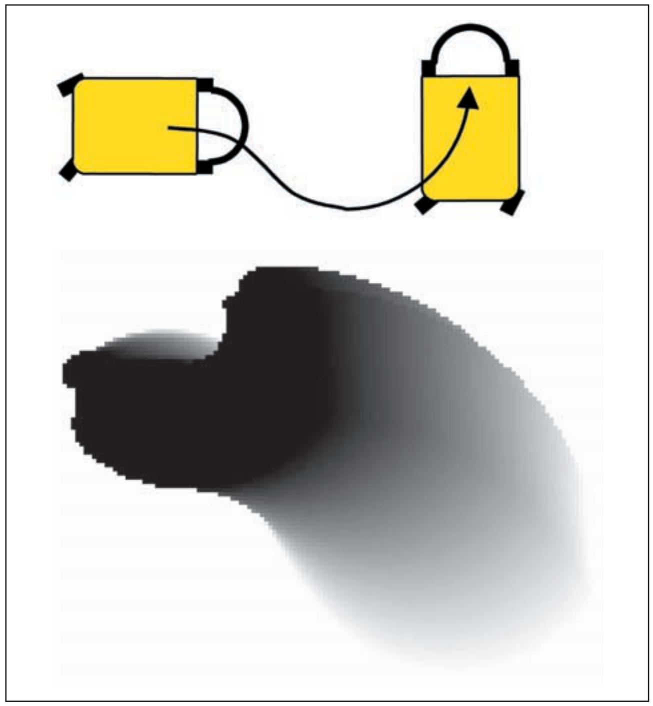


Fig. 7. *Example of an avoid speed map. For each cell, the map contains the highest speed at which the wheelchair will not collide with the cell while changing the steering angle from maximum right to maximum left until it turned by 90° to the left. The speed is depicted in grayscale; the brighter the cell, the faster the wheelchair can drive.*

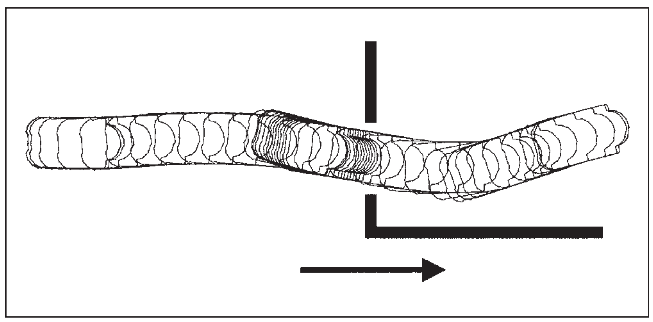


Fig. 8. *Doorway passage while driving forward.*

The Mobility Assistant

The Mobility Assistant builds the framework for the skills and behaviors mentioned so far. Depending on the specific needs of the user, various levels of support, so-called *assistants,* can be activated. The idea behind this modular concept is to help the human operator only with respect to certain actions or maneuvers he or she cannot perform independently any longer. Two such support tools are the *Driving Assistant* and the *Route Assistant.*

Scenario: The Driving Assistant

In their basic configuration, common power wheelchairs are usually controlled with a joystick. The human operator determines the speed and the steering angle by moving the joystick to a certain position. Even though in many wheelchairs the sensitivity of the joystick can be adapted to the user's motor capabilities, it is very hard to travel collision-free through narrow indoor environments such as apartments, shops, or offices. To overcome these problems, and especially to allow elder people and persons with slow reactions the use of a power wheelchair, the driving assistant combines the three elementary skills that were presented in above. As a consequence, the human operator can concentrate on the global navigation task by setting a rough direction where the vehicle should go. The driving assistant is responsible for the local maneuvers.

Being supported by the driving assistant while traveling with the wheelchair feels quite comfortable because the behavior of the vehicle is adapted to the current situation of the environment. This kind of a mobility assistant was successfully tested at the Hannover Trade Fair 1999 where the wheelchair traveled without a single collision for many hours through corridors overcrowded with trade show visitors.

In comparison to the driving assistant, the responsibilities for the global and local navigation tasks are exchanged in the route assistant.

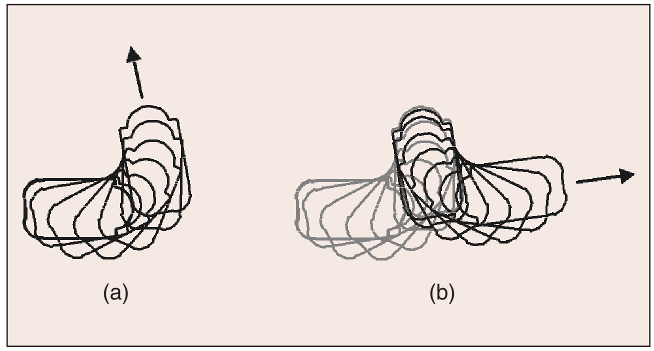


Fig. 9. *Turning around in open space: (a) Turning left forward. (b) Turning right backward.*

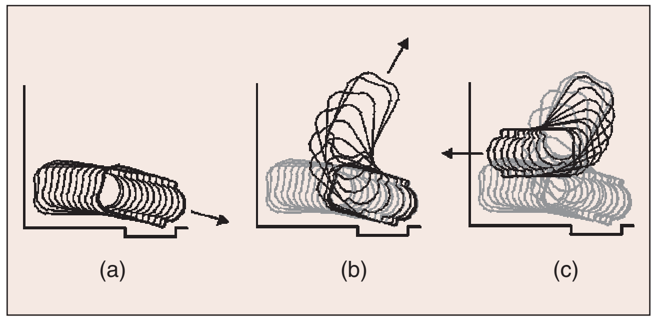


Fig. 10. *Turning around in a corner: (a) Trying to turn right. (b) Turning left backward. (c) Turning right forward.*

Scenario: The Route Assistant

Car navigation systems based on the Global Positioning System (GPS) provide instructions for the driver such as to take a certain junction of a motorway, etc. The backbone of these systems is a self-localization module that cooperates with a simple routing algorithm. The self-localization procedure makes use of the satellite-based GPS information in combination with data drawn from digital maps of the world and the locomotion of the car itself.

When transferring such a navigation system to the wheelchair context, two applications are plausible:

- Direct implementation of the *global* concept used in cars. This would require much more detailed maps because the resolution of the GPS-generated position as such is not precise enough to distinguish between, for example, a sidewalk and a street.
- The idea of giving instructions to the driver at certain decision points is adapted by replacing the global self-localization with a local one.

Whereas the first application cannot be realized at the moment due to the lack of GPS information and digital maps, the second application has already been implemented in a preliminary version on the Bremen Autonomous Wheelchair.

The basic idea of the route assistant is to make use of the capabilities of both controlling entities of the wheelchair: the human operator and the technical system. There is a certain community of people who are able to perform motor tasks such as riding a bike or steering a power wheelchair but who cannot memorize the routes or environments they are using. For example, most amnesic patients are very good in carrying out procedural tasks but fail when they should find the way to their office. In most cases, these people have a small number of distinct routes they are using every day and certain places where they spend their time.

Taking into account these considerations, the route assistant of the Bremen Autonomous Wheelchair provides the following functionality: During a teaching phase, the system explores the routes and places pertinent for the future user(s). If, for example, the wheelchair is used in a rehabilitation center for amnesic patients, the routes to all relevant places in the building could be learned. The route assistant uses its dead-reckoning system to build a so-called route graph that represents the traveled routes. With the help of a generalization algorithm, the routes are stored as sequences of straight line segments that join under certain angles [16]. In the replay mode, a nurse chooses a certain target for the patient in the wheelchair. The patient is independently responsible for controlling the vehicle with respect to local maneuvers such as obstacle avoidance. Similar to a GPS-based navigation system, the larger scale navigation is done by the route assistant by giving instructions where to go at decision points, enabling the patient to travel around on his or her own. This approach does not require additional sensors such as cameras to detect landmarks, because it only requires the wheelchair's odometry data.

Conclusion

The Bremen Autonomous Wheelchair is designed as a mobility assistant for the handicapped. Its software architecture is realized as a hierarchy in which higher modules can rely on the functionality provided by the lower levels, especially by the safety layer that guarantees collision-free motion of the vehicle. During the development of this safety layer, it turned out that common static firing strategies for ultrasonic sensors are inherently unsafe as long as only a few sensors are fired simultaneously. Therefore, a new adaptive strategy has been implemented that delivers a complete coverage of the robot's environment relevant for collision prevention very quickly. The elementary skills for smooth speed control, avoiding obstacles, and turning around build the foundation for the *driving assistant*, an application that eases everyday handling of the wheelchair. A second kind of support tool is the *route assistant*. It helps amnesic patients to navigate in middle-scale environments (e.g., in hospitals).

Acknowledgments

The Deutsche Forschungsgemeinschaft supports this work through the priority program "Spatial Cognition."

Keywords

Safety, rehabilitation, robotics, sensors, mobile robotics, assistance.

References

[1] D.A. Bell, S.P. Levine, Y. Koren, L.A. Jaros, and J. Borenstein, "Design criteria for obstacle avoidance in a shared control system," in *Proc. RESNA '94 Conference*, Nashville, TN, 1994, pp. 17-24.

[2] J. Borenstein, H.R. Everett, and L. Feng, *Where am I? Sensors and Methods for Mobile Robot Positioning*. Ann Arbor: University of Michigan, 1996.

[3] J. Borenstein and Y. Koren, "Error eliminating rapid ultrasonic firing for mobile robot obstacle avoidance," *IEEE Trans. Robot. Automat.*, 11(1): 132-138, 1995.

[4] W. Burgard, D. Fox, and D. Henning, "Fast grid-based position tracking for mobile robots," in G. Brewka, Ch. Habel, and B. Nebel (eds.), *KI-97: Advances in Artificial Intelligence* (Lecture Notes in Artificial Intelligence). Berlin: Springer, 1997, pp. 289-300.

[5] A. Elfes. "Occupancy grids: A stochastic spatial representation for active robot perception." In S.S. Iyengar and A. Elfes (eds.), *Autonomous Mobile Robots*, volume 1. Los Alamitos, CA: IEEE Computer Society Press, 1991, pp. 60-70.

[6] R. Furtwängler, U.D. Hanebeck, and G. Schmidt, "Dynamic control of a mobile manipulator," in *Proc.AMC'98, 5th Int. Workshop on Advanced Motion Control*, Coimbra, Portugal, 1998, pp. 440-445.

[7] B. Krieg-Brückner, Th. Röfer, H.-O. Carmesin, and R. Müller, "A taxonomy of spatial knowledge and its application to the Bremen Autonomous Wheelchair," in Ch. Freksa, Ch. Habel, and K.F. Wender (eds.), *Spatial Cognition* (vol. 1404 of *Lecture Notes in Artificial Intelligence*), Berlin: Springer, 1998, pp. 373-397.

[8] A. Lankenau, O. Meyer, and B. Krieg-Brückner, "Safety in robotics: The Bremen Autonomous Wheelchair," in *Proc. AMC'98, 5th Int. Workshop on Advanced Motion Control*, Coimbra, Portugal, 1998, pp. 524-529.

[9] A. Lankenau and O. Meyer, "Der autonome Rollstuhl als sicheres eingebettetes System.," master's thesis, Fachbereich Mathematik und Informatik, Universität Bremen, 1997.

[10] A. Lankenau and O. Meyer, "Formal methods in robotics: Fault tree based verification," in *Proc. Quality Week Europe*, Brussels, Belgium, 1999.

[11] N. Leveson, *Safeware: System Safety and Computers*. Reading, MA: Addison-Wesley, 1995.

[12] C. Owen and U. Nehmzow, "Middle scale navigation - A case study," in *Spatial Reasoning in Mobile Robots and Animals* (AISB-97 Workshop), Manchester University, 1997, pp. 104-112.

[13] Th. Röfer and A. Lankenau, "Architecture and applications of the Bremen Autonomous Wheelchair," In P.P. Wang (ed.), in *Proc Fourth Joint Conference on Information Systems*, volume 1, Association for Intelligent Machinery, pp. 365-368, 1998.

[14] Th. Röfer and A. Lankenau, "Ensuring safe obstacle avoidance in a shared-control system," in J.M. Fuertes (ed.), *Proc. 7th Int.Conf. Emergent Technologies and Factory Automation*, pp. 1405-1414, 1999.

[15] Th. Röfer, "Strategies for using a simulation in the development of the Bremen Autonomous Wheelchair," in R. Zobel and D. Moeller (eds.), *Simulation-Past, Present and Future*. Society for Computer Simulation International, 1998, pp. 460-464.

[16] T. Röfer, "Route navigation using motion analysis," in *Proc. Conf. Spatial Information Theory '99* (vol. 1661 of *Lecture Notes in Artificial Intelligence*). Berlin: Springer, 1999, pp. 21-36.

[17] N. Storey, *Safety-Critical Computer Systems*. Reading, MA: Addison-Wesley, 1996.

Axel Lankenau studied computer science at Bremen University (Germany). As a research assistant and Ph.D. student at the Bremen Institute of Safe Systems (BISS), he is currently working in the priority program "Spatial Cognition" of the Deutsche Forschungsgemeinschaft (DFG). Primarily, he is interested in the application of formal methods to the design of service robotic systems. He also works on local navigation strategies for mobile systems. He is involved in the Bremen Autonomous Wheelchair project, which aims at the development of an assistive device for handicapped and elder people.

Thomas Röfer studied computer science at the University of Bremen (Germany). He received his master's degree in 1993. Then, he was a Ph.D. student in the graduate study program "Spatial Orientation and the Organization of Actions of Autonomous Systems," funded by the Deutsche Forschungsgemeinschaft and the Freie Hansestadt Bremen. Since 1997, he has worked in the priority program "Spatial Cognition" that was as well funded by the DFG. He finished his Ph.D. thesis, "Panoramic Image Processing and Route Navigation," in 1998. Currently, he is an assistant professor at the Department of Mathematics and Computer Science at the University of Bremen.

Address for Correspondence: Thomas Röfer, Bremer Institut für Sichere Systeme, TZI, FB 3 Mathematik/Informatik, Universität Bremen, Postfach 330440, 28334 Bremen, Germany. Tel.: + 49 421 2184659. Fax: + 49 421 2187585. E-mail: roefer@tzi.de.