Architecture and Applications of the Bremen Autonomous Wheelchair

Thomas Röfer and Axel Lankenau

Bremer Institut für Sichere Systeme, FB3, Universität Bremen, Postfach 330440, D-28334 Bremen
roefer@tzi.de, alone@tzi.de

Abstract

The work described here is motivated by the necessities for safety and multi-functionality in rehabilitation robotics. The *Bremen Autonomous Wheelchair* will be presented as a platform for the realization of rehabilitation applications. This paper mainly focuses on safety issues. Thus, it presents the wheelchair's system architecture that contains a safety layer as its central module. Three applications realized on top of this layer will be outlined.

1 Motivation

Due to the changed distribution of ages in future populations, more and more elder people, in addition to handicapped persons, will be forced to maintain their mobility with the help of rehabilitation technology. As there is a large number of possible disabilities, a wide range of support technologies has to be developed to provide an adequate assistance to the handicapped users, from simple mechanic wheelchairs and manually controlled power wheelchairs to sophisticated semi-autonomous robot systems that may even include robot arms. In order to maintain the user's mental activity, the transition between manual control and autonomy has to be fluent, i.e. people should only get as much support as they really need. This implies that it is not possible to develop *the* semi-autonomous wheelchair that satisfies the needs of all handicapped persons. Instead, a system should be *modular* in terms of software *and* hardware to allow to set-up an adequate vehicle for each particular user.

In recent years, a large number of service robotics applications has been developed, e.g. surveillance robots or autonomous vacuum-cleaners. As these robots often operate in environments in which people live and work, they have to satisfy very strict requirements with regard to dependability. In case of rehabilitation robots, e.g. manipulators, or intelligent wheelchairs, safety considerations become the dominating design issues. Accordingly, both trends—the increasing importance of safety as well as multi-functional platforms—have to be thoroughly taken into account when developing a rehabilitation robot. Considering these leitmotivs, the Bremen Autonomous Wheelchair is developed as a safe system that, due to its modular system architecture, can be adapted to various needs of specific users.

2 The Wheelchair

The basis for the Bremen Autonmous Wheelchair "Rolland" is a commercial power wheelchair manufactured by Meyra (cf. Fig. 1). It is a non-holonomic vehicle that is driven by its front wheels and steered by its back wheels. This model already has two serial ports that allow to set the speed and the steering angle every 32 ms as well as to get back the systems actual state. So far, the wheelchair was equipped with 27 Polaroid sonar sensors mounted around the system and a QNX-driven industrial PC placed behind the seat. The user controls the wheelchair with a joystick.

Figure 1: The Bremen Autonomous Wheelchair.

3 Safety Issues

In order to ensure the correct behavior of the wheelchair, the following approach has been chosen: A hazard analysis was performed that figured out the safety requirements of the system. In addition, a detailed specification of the environment could be derived from this analysis (Lankenau *et al.*, 1998). Safety mechanisms were defined to prevent the system from running into hazardous states. Finally, important safety aspects were verified by model-checking (Lankenau and Meyer, 1997).

Safety-Critical Systems. A system is *safety-critical* if its malfunction is able to cause catastrophic consequences on the environment (Laprie, 1992). Examples are nuclear power plants, railway interlocking systems etc. Designing safety-critical systems necessitates a hazard analysis to figure out the dangerous states the system should avoid, a formal specification of the resulting safety requirements, and some kind of validation technique, e.g. formal verification or systematic testing. One approach to realize a safe system is to create a safe controller hosting the implemented safety mechanisms which ensure that the original system never enters a hazardous state.

In the wheelchair application, it turned out that the known safety notion of Laprie (1992) should be broadened inasmuch as the avoidance of collisions as well as the direct control of locomotion by the user are subsumed.

Hazard Analysis. Lankenau and Meyer (1997) presented a fault-tree based hazard analysis that refines the main hazard "*System failure with regard to safety specification*" by its possible causes. In a fault-tree analysis, relations and dependencies of various faults and failures can be represented. Thus, this hazard analysis technique suits the requirements of problem domains in robotics very well.

As an example, consider the threat of a *collision* as a refinement of the system-failure hazard. A collision can either be an *active collision*, i.e. the wheelchair hits an obstacle, or a *passive collision*, i.e. an obstacle hits the wheelchair. Active collisions will be averted by the Sense&Act-Module (cf. section 4), whereas passive collisions have to be defined away by stating that no object in the environment will actively crash into the wheelchair (*specification of the environment*). There are several possible causes of an active collision, only one of which will be considered here: a *measuring error of the sonar sensors*. Such an error becomes fatal if and only if the measured distance is larger than the real distance to the obstacle *and*, as a result, the distance to the obstacle is overestimated. Assuming that on the one hand, the system has to be able to cope with n consecutive misreadings with values too large, but on the other hand, the $(n+1)^{\text{th}}$ value is guaranteed to be either correct or to small, only the overestimation has to be ruled out.

The remaining hazard (*overestimation of obstacle distance*) cannot be further refined. Thus, a safety mechanism has to be defined that prevents the system from an overestimation. Note that this mechanism can rely on the fact that no more than n consecutive measurements are too large. The safety mechanism is realized in the *Sense&Act-Module*, i.e. the basic layer in the system's architecture.

4 System Architecture

The architecture depicted in figure 2 consists of the wheelchair, a control PC (the *base host*), and optionally several other computers (the *application hosts*). The hosts are connected via an Ethernet wire. Driven by the base host, a real-time capable frame-protocol on a client/server basis has been developed. The various application modules do not necessarily satisfy real-time conditions and—as far as they do not run on the base host—are not restricted to using QNX as operating system.

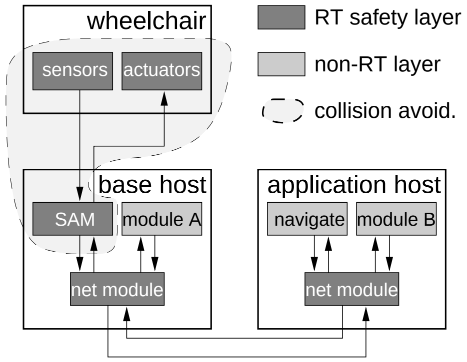

Figure 2: System architecture.

The Sense&Act-Module (SAM) builds the interface to the real wheelchair. It reads the sensor values (sonar measurements, actual speed, and actual steering angle) and forwards the desired values for speed and steering angle that are generated by application modules to the hardware. It only intervenes if these driving commands are not safe, i.e. their execution may cause a collision.

Local Obstacle Map. In order to be able to detect threatening collisions, SAM manages a local obstacle map. This map is a quadratic array of cells, in which each cell encodes the presence of an obstacle at the corresponding position in reality. Similar to other approaches (e.g. by Jörg *et al.*, 1993), the map is scrolled if the wheelchair moves but it is not rotated. Instead, the vehicle's rotation relative to the map is maintained. Thus the map is local in x and y coordinates, but has a semi-global rotation, i.e. its rotation will only shift slowly due to odometry errors, but this does not influence the method described here.

On the one hand, such a map is necessary because the sonar sensors are fired sequentially and the wheelchair can move up to 36 cm during a complete sensor update. On the other hand, it is required to satisfy the safety requirements specified in the hazard analysis. The size of the map is dictated by the assumptions made in the analysis. As it is assumed that every obstacle is seen after $n+1$ sensor updates at the latest, these $n+1$ measurements must fit into the map at the wheelchair's maximum driving speed. Readings older than $n+1$ sensor updates are removed from the map to forget misreadings generated by dynamic objects. As the vehicle travels up to $u = 36$ cm during a sensor update, the map must have an extent of at least $n+1$ times this distance. In addition, the system has an overall stopping distance of $s = 53$ cm and it should keep a minimum distance of $m = 15$ cm from all obstacles. The wheelchair's own extent w has also to be taken into account. So, the minimum extent of the map in one direction is calculated as

$$e = u\,(n+1) + s + m + w \qquad (1)$$

As the sensors used are quite reliable, $n = 1$ has been assumed. This results in an extent of at least 180 cm in each direction. Thus the wheelchair has to maintain a map of 360×360 cm^2. In order to be precise enough to allow the driving through doorframes, the map consists of 14400 3×3 cm^2 cells. Its maintenance—entering new sensor readings, shifting, forgetting old readings—is done in less than 3 ms on a Pentium 133 PC. As the wheelchair accepts drive commands every 32 ms, the map is updated 31 times per second.

Virtual Sensors. In order to decide whether an entry in the map represents a danger for the wheelchair, the system's movement must be taken into account. In a vehicle with a driving axle and a steering axle, this is rather complex 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 may stick out sideways. In contrast to other approaches that try to reduce the computational costs by simplifying the shape of the region that has to be searched for obstacles (e.g. by Furtwängler *et al.*, 1998), an exact method has been used. For each combination of driving direction, steering angle, and rotation, the wheelchair's movement over the distance $s+m$ has been pre-calculated and stored using the robot simulator SimRobot (Röfer, 1998b). As a result, for each of these states it is known which cells in the obstacle map have to be checked and how far the wheelchair has to drive to reach each cell. So, this is a means to determine the distance to obstacles, and because it works on an abstraction of the environment, it is called *virtual sensor* (cf. Fig. 3).

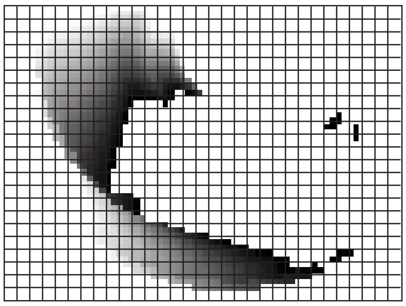

Figure 3: Virtual sensor. Brightness indicates distance.

To speed up searching the map, only the indices (relative addresses) of the relevant cells are stored. In addition, these indices are sorted by the distance the wheelchair has to move to reach the corresponding cell. As a result, the first obstacle found in the map is always the closest to the wheelchair and its distance is immediately known. On a Pentium 133, this takes less than 75 μs. This is more than 50 times faster than a real sonar sensor would need to detect obstacles within a distance of 68 cm!

5 Applications

As discussed in section 1, several levels of support are required to help handicapped persons. Hence, various hierarchically organized modules have been implemented on top of the safe Sense&Act-Module that only brakes the wheelchair to a standstill in time, whenever an obstacle is dangerously close.

SpeedWizard. As SAM only decides whether to stop or not, its interventions are quite abrupt. Therefore, to realize a smooth braking behavior, the *SpeedWizard* manages a second local map that covers a larger area but is less precise. Again, virtual sensors are used to determine the distance to the closest obstacle. Similar to Bell *et al.* (1994), this module sets the target speed to a value half as large as the one SAM would have allowed. This way, the wheelchair's locomotion becomes rather smooth and comfortable. This is useful for all users of a wheelchair because they can always choose the maximum speed with their joystick and can concentrate on steering, while the vehicle selects an adequate velocity with respect to the current obstacle situation.

DoorWizard. Whereas the previous levels confine themselves to reducing the wheelchair's speed, the *DoorWizard* only affects the steering angle. This module realizes a short-distance obstacle avoidance mechanism that facilitates the collision-free passage of doorways. Using virtual sensors, the DoorWizard searches for the steering angle with the largest distance to obstacles that is as close as possible to the direction indicated by the user. This helps persons who are unable to precisely control the wheelchair. Otherwise, they would either crash into the doorjambs (without guidance), or they would get stuck in the doorway (if they only relied on SAM).

Route Navigation. Currently, the highest level of support is the autonomous driving of pre-taught routes in networks of passages. This competence is actually ported from Rolland's predecessor. As described by Krieg-Brückner *et al.* (1998), routes are represented as sequences of *basic behaviors* and *routemarks* that trigger the switching between these behaviors. These combinations are learned during a single teaching drive. After that, the wheelchair is able to travel this route autonomously. This will support users suffering from severe impairments who are currently not able to control a power wheelchair, e.g. blind people.

6 Experimental Results

An analysis of the SpeedWizard was performed by Lankenau *et al.* (1998). The route navigation was described by (Krieg-Brückner *et al.*, 1998; Röfer, 1998a). Hence, only the DoorWizard is analyzed in this paper. Figure 4 visualizes the performance of the DoorWizard with different initial positions and rotations. The wheelchair was always started in a distance of 60 cm from the wall the 92 cm wide doorway was located in. In the analysis, the wheelchair's relative offset to the door and its rotation were varied. If both offset and rotation were zero, the wheelchair stood perpendicularly and centered in front of the door. The data shown in figure 4 was generated using a simulation (Röfer, 1998b). However, spot checks have proven that the real system behaves similar. The result shows that the wheelchair passes the doorway as long as it is oriented towards it. It also reflects the vehicle's asymmetric sensor arrangement. Note that the wheelchair has the typical kinematics of a car driving backwards, and therefore from many starting positions, it is not possible to pass the doorway without shunting.

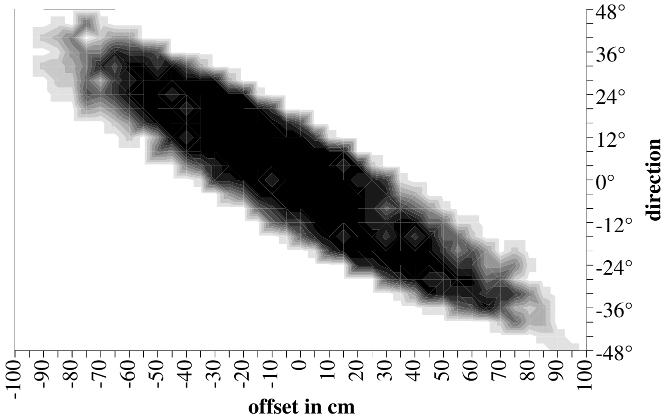

Figure 4: Performance of the DoorWizard. The darker the area, the more often the wheelchair successfully passed the doorway from the corresponding location.

7 Conclusion and Future Work

This paper has presented the architecture of the Bremen Autonomous Wheelchair and has emphasized the need for safety in rehabilitation robotics. In the system discussed, safety with respect to collisions is guaranteed by the Sense&Act-Module. This module uses a local map and the concept of *virtual sensors* to detect threatening collisions very rapidly. First applications of the wheelchair are modules that support users in the control of speed, steering, and even in the following of complete routes.

Future work will include the implementation of shunting and additional behaviors, e.g. turning or docking with tables. In order to broaden the range of possible users, a speech recognition system will be integrated, enabling users to activate basic behaviors by spoken commands, e.g. *straight ahead* or *left*, and to specify target locations for the route navigation.

Acknowledgement

The authors are supported by the Deutsche Forschungsgemeinschaft through the priority program "Spatial Cognition" and by the Freie Hansestadt Bremen.

References

Bell, D. A., Levine, S. P., Koren, Y., Jaros, L. A., and Borenstein, J. (1994). Design criteria for obstacle avoidance in a shared control system. pages 17–24, Nashville.

Furtwängler, R., Hanebeck, U. D., and Schmidt, G. (1998). Dynamic control of a mobile manipulator. In *Proceedings of AMC'98, 5th Int. Workshop on Advanced Motion Control*, pages 440–445, Coimbra, Portugal.

Jörg, K.-W., v. Puttkamer, E., and Richstein, H.-J. (1993). Integration und Fusion heterogener Multisensorinformation zur geometrischen Weltmodellierung für einen Autonomen Mobilen Roboter. In G. Schmidt, editor, *Autonome Mobile Systeme*, volume 9, pages 287–298, Technische Universität München.

Krieg-Brückner, B., Röfer, T., Carmesin, H.-O., and Müller, R. (1998). A taxonomy of spatial knowledge and its application to the bremen autonomous wheelchair. In C. Freksa, C. Habel, and K. F. Wender, editors, *Spatial Cognition*, volume 1404 of *Lecture Notes in Artificial Intelligence*, pages 373–397, Berlin, Heidelberg, New York. Springer.

Lankenau, A. and Meyer, O. (1997). *Der autonome Rollstuhl als sicheres eingebettetes System*. Master's thesis, Universität Bremen.

Lankenau, A., Meyer, O., and Krieg-Brückner, B. (1998). Safety in robotics: The Bremen Autonomous Wheelchair. In *Proceedings of AMC'98, 5th Int. Workshop on Advanced Motion Control*, pages 524–529, Coimbra, Portugal.

Laprie, J. C., editor (1992). *Dependability: Basic Concepts and Terminology*. Springer, Berlin, Heidelberg, New York.

Röfer, T. (1998a). *Panoramic Image Processing and Route Navigation. Ph. D. thesis*. Number 7 in BISS Monographs. Shaker. To appear.

Röfer, T. (1998b). Strategies for using a simulation in the development of the Bremen Autonomous Wheelchair. In R. Zobel and D. Moeller, editors, *Simulation-Past, Present and Future*, pages 460–464. Society for Computer Simulation International.