Chapter 43
Obstacle Handling of the Holonomic-driven Interactive Behavior-operated Shopping Trolley InBOT

Michael Göller, Thilo Kerscher, Johann Marius Zöllner, and Rüdiger Dillmann

43.1 Introduction

The interactive behavior operated trolley (InBOT, Fig. 43.1) is an approach to transfer state of the art robotics technology into human everyday environments. The benefits of the actual technology shall become available for the masses. InBOT addresses several everyday problems and therefore the main task is to ease the daily shopping. Among other possibilities this means to help the customer to find the desired products without extensive search in big supermarkets or to relieve the customer from the burden to push the shopping cart using his own force all the time, especially if the cart is heavily loaded or the customer is elderly or handicapped. InBOT provides four different modes of operation. These are *haptic steering*, *following*, *guiding* and *autonomous*. In all the modes the user is assisted in several ways. Additionally InBOT is able to perform local maneuvers like finding a parking position at a wall or turning to help loading heavy goods. A flexible mode-dependent task-planner rounds out the capabilities of InBOT. It is able to stop and continue or postpone given tasks at runtime in combination with topological navigation [8]. A rough overview over the main components of InBOT is illustrated in Fig. 43.2. Finally, InBOT is equipped with the holonomic mecanum drive because the robot shall not lack the manoeuvrability of conventional shopping carts.

43.1.1 Behavior-based Control

As baseline of all these modes acts a *behavior-based control* that provides a repertoire of behaviors to be used in the course of the four modes. Among the provided

Michael Göller, Thilo Kerscher, Johann Marius Zöllner, and Rüdiger Dillmann
FZI Research Center for Information Technology, Haid-und-Neu-Str. 10-14,
D-76131 Karlsruhe, Germany
e-mail: {goeller,kerscher,zoellner,dillmann}@fzi.de

K.R. Kozłowski (Ed.): Robot Motion and Control 2009, LNCIS 396, pp. 463–472.
springerlink.com

behaviors is for example the capability to localize itself and to find a way to a desired product or the avoidance of obstacles or dynamic objects. The architecture with multiple levels of abstraction is described in [7]. Fig. 43.4 summarizes the levels of control containing the repertoire of basic behaviors. This document focuses on a major feature that is relevant to all modes of control and in all imaginable tasks all the same: safe and reliable navigation without collisions as well as the ability to find effective and smooth paths around obstacles. Especially thinking of a not exactly lightweight robot loaded with several heavy goods makes this necessity obvious.

Fig. 43.1 The interactive behavior operated shopping trolley (InBOT); it is equipped with a mecanum drive for full holonomic movements

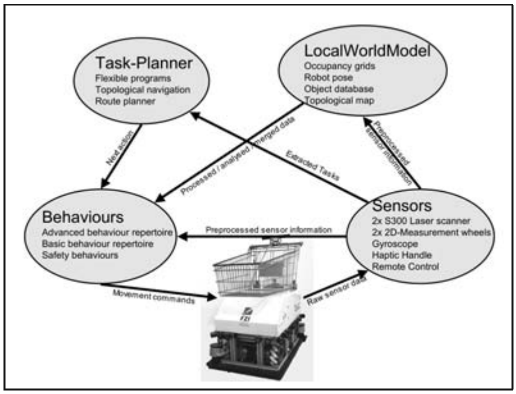

Fig. 43.2 The abilities of InBOT are grouped in 2 repertoires of behaviors with a flexible task-planner on top; the *local world model* collects and preprocesses information acquired by the sensor systems

The main components of the obstacle handling are located in the reactive layer of the behavior-based control. This layer is implemented by a behavior network. Here each behavior module (see Fig. 43.3) operates independently. The generated output of the different behaviors is merged by a fusion behavior. The concept of the behavior network is described in detail in [7]. An important feature of the reactive control concept is that at every time a suitable actor-output is available. The basic reactive behaviors provide a constant stream of commands to control the robot even if deliberative behaviors cannot contribute in time. The use of a special fusion behavior for all reactive behaviors offers a broad approach. Various behaviors with different main objectives can be merged this way.

43.1.2 Related Work and Discrimination

Potential field methods [9] are very popular. Due to restrictions of the field of view, potential field methods are computational highly efficient. But they have some limitations like trapping situations or oscillations [12], so these methods are often

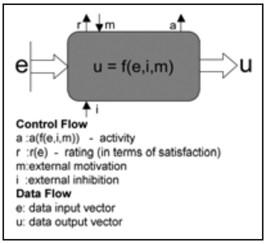

Fig. 43.3 Behavior module, the basic unit of the behavior-based control

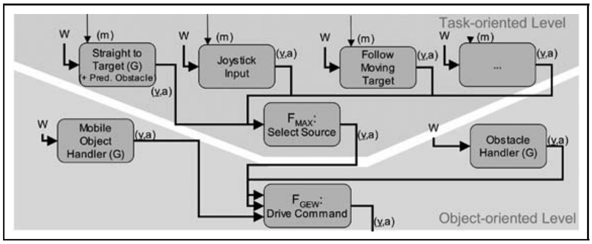

Fig. 43.4 The two levels of control containing the basic behavior repertoire: the task-oriented level and the object-oriented level

enhanced [11, 1]. A comparable approach, the virtual force field, is introduced in [2] and later on leads to the vector field histograms [3, 15]. These are based on the occupancy-/certainty-grid methods [4, 14]. All cells in a defined window generate a repelling force which works on the robot. Generally these methods don't take the robot dynamics into account. In contrast, the dynamic windows method [6, 5] for collision avoidance is specially designed to cope with higher velocities where the robot dynamics must not be omitted. Another approach are the elastic bands [10, 13]. These optimize a given path according to static or dynamic obstacles.

The navigation concept provided in this document proposes a potential field method based on occupancy grids encapsulated in behavior modules. In contrast to usual potential field methods the robot dynamics is taken into account. Therefore a dynamic potential field is computed online all the time. In contrast to [3] one representative repelling vector is generated for every object to save computational power. All cells are weighted by their relevance for the robot; irrelevant cells are of no consequence. Another specialty of the concept provided in this document is the capability of the robot to perform a predictive obstacle analysis prior to the application of the potential field. Here subtargets are generated to move the robot along an efficient path through the obstacles and to avoid for example moving into deadends. Additionally the holonomic character of the mecanum drive has to be taken into account.

43.2 Local World Model

The *local world model* is a layer- and level-independent model of the robot surroundings and inner state. The reactive behaviors responsible for the obstacle avoidance work based on a occupancy grid [4, 14] that is kept in the world model. The surrounding environment is discretized into a grid of cells. Each cell contains a value that represents the probability that the cell is at least partially occupied. These grids merge the information from various sources like two laser-scanners or information received from other robots. The more deliberative behaviors make use of an object database. This database is generated by identifying and tracking objects in the grids.

43.3 Obstacle Handling

The behaviors presented in the following sections are responsible for the handling of surrounding objects. The behaviors responsible for the obstacle avoidance can be separated in two levels of abstraction: point-based and velocity-based. The members of the first group manage the deliberative and predictive avoidance to generate sub-target-points, the members of the second group on the other hand contain the reactive behaviors which generate attracting or repelling velocity vectors.

43.3.1 Predictive Avoidance of Obstacles

If a target location is known, the task is a point-to-point movement instead of e.g. the haptic control of the robot, predictive obstacle handling takes place to overcome the local minima problem of the potential field approaches [1, 12] and to generate more efficient and shorter paths. If the direct path to the target is blocked, the obstacles are scanned for corners. The corners are then ordered by their quality (Q_C) that depends on the corner's distance ($d_{T,C}$) and direction ($\Theta_{T,C}$) compared to the target as well as the width of the free room around it (w_C), taking into account a minimum width (w_{min}) the robot needs because of its physical dimension. Additional constant parameters are used to set a minimum quality (q_{min}) and to limit the quality resulting from a large free space around the corner (w_{max}). The best corner is chosen as new subtarget and temporarily replaces the original target. This way the robot is driving on some kind of dynamic visibility graph (see Fig. 43.5 and 43.6)

$$\begin{aligned} Q_C = & \max^2(q_{min};(1-\frac{\Theta_{T,C}}{2\pi}))\cdot\max(q_{min};(1-\frac{\min(d_{T,C};d_T)}{d_T}))\cdot \\ & \cdot\min(\max(w_C-w_{min};0);w_{max}-w_{min}). \end{aligned} \tag{43.1}$$

43.3.2 Reactive Avoidance of Obstacles

Each one of the reactive behaviors generates one velocity-vector. All these vectors are merged with the vector from one of the task-oriented behaviors and generate some kind of highly dynamic potential field that depends on many parameters like the status (position, velocity, orientation, task) of the robot and environmental information (distance, position, velocity of objects and target). Because of the complexity of the processed information no path planning is performed. The robot calculates the

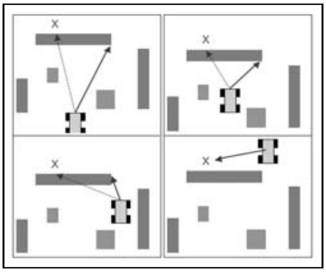

Fig. 43.5 A sequence of subtargets is generated on the way to the target location, forming some kind of dynamic visibility graph

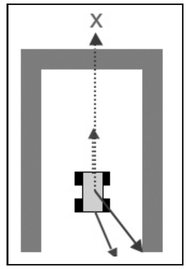

Fig. 43.6 The robot automatically avoids U-shaped obstacles or moves out of them by choosing the best-suited corner as subtarget

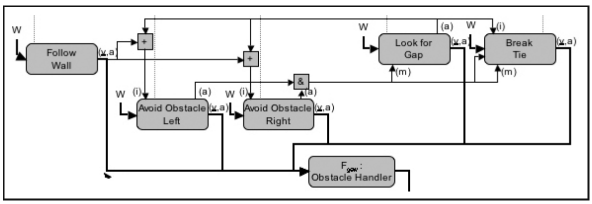

Fig. 43.7 Behavior module group for handling static obstacles

most beneficial movement online instead. The structure of the corresponding group is illustrated in Fig. 43.7. Two behaviors observe the right- and left-hand obstacles. In special cases *Look for Gaps* is motivated to look for passages and *Break Tie* is activated to force a decision if necessary. The behavior for wall following inhibits one of the obstacle avoidance behaviors.

43.3.2.1 Behavior: Avoid Obstacles

This is the central behavior of this group. It gives the robot the ability to avoid static obstacles by letting the robot repel from a special center of gravity (CoG) of the obstacles. Each occupied cell of the occupancy grid belonging to the obstacle is weighted taking into account the relation to the robot, obstacle and target direction. The result is a new repelling velocity vector.

Relevant directions: Due to the holonomic drive there are two relevant directions ($\mathbf{P}_T$) for the robot (see Fig. 43.8 and 43.9). The first one is the actual driving direction, the second one is the direction the robot intends to accelerate in. In the case of a differential drive these two directions are always the same, but in the case of a holonomic drive they can differ. Here both directions have to be checked for obstacles. Therefore four instances of the behavior are needed, one for each side of the two relevant directions.

Obstacle relevance: The visible obstacles, i.e. the occupied cells ($\mathbf{O}_i$) of the certainty grid, are weighted ($W()$) by their distance ($d()$) to the robot and by their

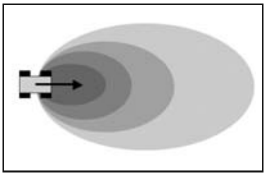

Fig. 43.8 Relevance of obstacles with differential drive: obstacles are only relevant if they are in the front area

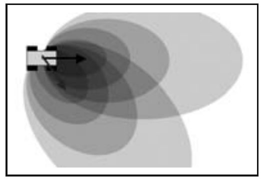

Fig. 43.9 Relevance of obstacles with holonomic drive: additionally the direction it intends to accelerate in is relevant (grey arrow)

angular position ($\theta()$) compared to relevant direction ($\mathbf{P}_T$). Obstacles outside a certain activity-area (d_A,θ_A) are not taken into account. This reduces the influence of obstacles that are not in the (intended) driving direction or just too far away. So driving along walls or through narrow corridors becomes easier and oscillations, a standard problem of potential field methods, can be avoided:

$$W(\mathbf{P}_X) = \max^2(d_A - d(\mathbf{P}_X,\mathbf{P}_R);0) \cdot \max^2(\theta_A - \theta(\mathbf{P}_X,\mathbf{P}_T);0). \tag{43.2}$$

Obstacles CoG: The CoG of the obstacles ($\mathbf{P}_C$) is the weighted sum of the occupied cells in the certainty grid:

$$\mathbf{P}_C = \sum_{i=1}^{M} \mathbf{O}_i \cdot W(\mathbf{O}_i). \tag{43.3}$$

Repelling vector: The repelling vector $\mathbf{u}_{AO}$ is calculated in three steps. First of all the direction is determined as the vector from the obstacle's CoG to the robot centre $\widehat{P}_{R-C}$. Then the length of the vector is calculated by using the weight of the CoG itself and multiplying it by the robot's velocity. This enables the robot to slowly approach an object but generates strong decelerations if the robot approaches fast. Finally the vector is adapted to the parameters (see Fig. 43.3) for the external control of the behavior, the motivation (m), the inhibition (i), and a proportional factor (f) for the overall adjustment. The concept was tested and evaluated using a MATLAB® simulation (see some examples in Fig. 43.10)

$$\mathbf{u}_{AO} = \widehat{P}_{R-C} \cdot W(\mathbf{P}_C) \cdot |\mathbf{v}| \cdot f \cdot m \cdot (1-i), \quad a = \|\mathbf{u}_{AO}\|, \quad r = \|\sum_{i=1}^{M} W(O_i)\|. \tag{43.4}$$

Activity and rating: The activity a of the behavior is the normalized length of the repelling vector. The rating r is the total sum of the weighted cells of the occupancy grid and represents the difficulty of the local environment regarding the amount and the relevance of the local obstacles.

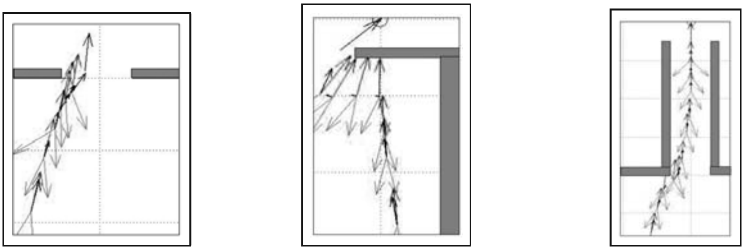

Fig. 43.10 MATLAB® simulation of two *Avoid Obstacle* behaviors (light grey repelling arrows) and *Head For Target* (dark grey attracting arrow); the resulting velocity vector is painted in black

43.3.2.2 Behavior: Look for Gaps

This behavior is activated if the activities of the *Avoid Obstacle* behaviors have an equally strong activity. In this case they cannot decide in which direction the robot shall drive. This can happen if the robot is located centrically in front of a small gap. Here *Look for Gaps* will identify the gap and drive through it without slowing down too much. The output of the behavior is a new velocity vector that points to the middle of the gap. This way the robot is additionally directed to the middle of the gap and the deceleration of the *Avoid Obstacle* behavior is countered.

Quality of gaps: A small area in direction of the robot's (intended) movement is searched for gaps, e.g. for free room that is framed by occupied cells. The width (w_G) of the free area, e.g. the gap, is compared with the minimum (w_{min}) and maximum (w_{max}) width a gap must fulfill. Additionally the gap's quality (Q_G) is influenced by the angular position ($\theta()$) of the gap centre ($\mathbf{C}_G$) compared to the relevant direction ($\mathbf{P}_T$) of the robot:

$$Q_G = \max(w_G - w_{min};0) \cdot \max(w_G - w_{max};0) \cdot \max(\theta_A - \theta(\mathbf{C}_G, \mathbf{P}_T);0). \quad (43.5)$$

The gap with the best quality is chosen to contribute an attracting vector.

Attracting vector: The attracting vector ($\mathbf{u}_G$) points towards the centre of the gap ($\widehat{C}_G$), is weighted and adapted to the parameters f, m (motivation), and i (inhibition):

$$\mathbf{u}_G = \widehat{C}_G \cdot \max(w_G - w_{min};0) \cdot f \cdot m \cdot (1-i), \quad a = \|\mathbf{u}_G\|, \quad r = \|\mathbf{Q}_G\|. \quad (43.6)$$

Activity and rating: The activity (a) of the behavior is the normalized length of the attracting vector. The rating (r) is proportional to the quality of the gap.

43.3.2.3 Behavior: Follow Wall

This behavior is an auxiliary behavior. It does not steer the robot towards a target. It provides an additional functionality that is useful in corridors. A two directional traffic can be implemented this way. It lets the robot keep a constant distance to a wall by inhibiting the corresponding obstacle avoidance behaviors with its activity. The output is a movement vector that corrects the distance to the wall.

Activity and rating: The activity is proportional to the output vector. The rating is proportional to the difference between the requested distance and the actual one.

43.3.2.4 Behavior: Break Tie

This behavior is activated if both activities of *Avoid Obstacle Left* and *Right* are equal for some time. In this case these two are obviously not able to make a decision in which direction the robot shall drive around the obstacle. This behavior generates a velocity vector that points alongside the obstacle. The actual sign of the vector is based on the direction of the target and the heading of the robot. If there is at least a slight preference for one direction this preference is amplified strongly. If there is absolutely no preference a random component is used.

Activity and rating: The activity (a) is proportional to the length and the angle of the new vector. The rating (r) is proportional to the position, angle, and distance off the obstacle.

43.4 Experimental Results

Several experiments were performed to evaluate the functionalities described above. Some of the results are presented in the following. Fig. 43.11 shows the path taken by InBOT when it was commanded to move to a point located behind a deadend while InBOT was located inside the deadend. The behavior responsible for the predictive obstacle handling (Section 43.3.1) leads the robot around the obstacle smoothly by generating appropriate subtargets. Another experiment is depicted in Fig. 43.12. It was performed fifty times. During the resulting 250 obstacle avoidance attempts only two very slight collisions occurred. In both cases it was a rear corner of the robot scratching an obstacle when swinging in too early after passing an obstacle. In this scenario InBOT was guiding the user through a corridor cluttered with several obstacles. In the scene presented in Fig. 43.14 the behavior *Look for Gaps* detects a narrow passage and helps maneuvering InBOT through it. The underlying occupancy map with the indicated velocity vectors from the different behaviors is depicted in Fig. 43.13 (bottom). In another scenario the user is pushing InBOT in the *haptic steering* mode. InBOT assists the user by automatically avoiding obstacles like a parked shopping cart. The underlying occupancy map with the indicated velocity vectors from the different behaviors is depicted in Fig. 43.13 (top).

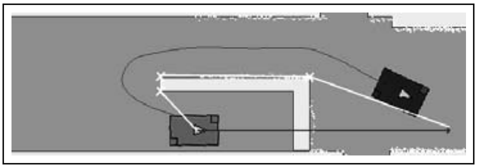

Fig. 43.11 Exemplary path of InBOT when commanded to move through a deadend. The grey robot indicates the starting position and the grey arrow shows the given movement command. The white Xs indicate the corners chosen as subtargets by the predictive obstacle handling. Finally the black line represents the path actually taken by the robot

Fig. 43.12 Exemplary path of InBOT moving through a corridor with several obstacles

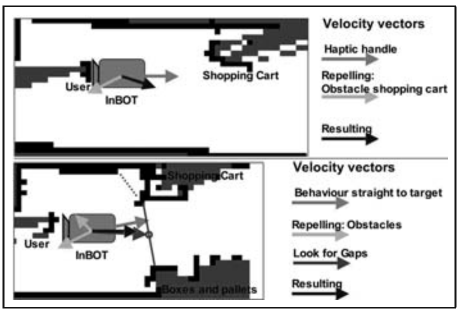

Fig. 43.13 Occupancy grid maps of InBOT with illustrated output of the reactive behaviors

Fig. 43.14 Picture of a scene where InBOT leads the user through a corridor

43.5 Conclusions and Outlook

Several specialized behaviors for a hierarchical behavior-based control were developed for a robot with a holonomic drive system. The ability focused on was the safe and reliable avoidance of all kinds of static obstacles as well as finding efficient paths through cluttered scenes. Occlusions had to be taken into account, therefore the robot had only the information it was able to acquire with its own sensors and therefore was not able to overview the scene. To respect this unpredictability a reactive approach was chosen. Another design feature was the expandability of the control. New modules providing new features, like the avoidance of moving obstacles shall be included in future easily.

In all tests the reactive component was able to avoid collisions with static obstacles reliably. The predictive obstacle handler on the other hand generates efficient paths that are comparable to those generated by visibility graph methods.This way the most prominent shortcoming of potential field methods, the local minima problem, is negated as well. It should be kept in mind that the robot has no global map knowledge and therefore is only able to plan the path in visibility range of the sensors or within a local memorized area. The network character of the control enables us to easily extend the control with new functionalities; either by hooking in new behavior modules which is done straight forward due to the use of fusion behaviors or by recombining present functionalities by activating the corresponding behavior modules. This way it is possible to use the obstacle avoidance functionality to augment the steering functionality of the haptic handle so that the intelligent trolley moves around obstacles while it is being pushed by it's user.

Acknowledgements. This research has been carried out in the CommRob (www.commrob.eu) project and is partially funded by the EU (contract number IST-045441 under the 6th Framework Programme).

References

1. Antich, J.: Extending the potential fields approach to avoid trapping situations. University of the Balearic Islands (2005)
2. Borenstein, J., Koren, Y.: Real-time obstacle avoidance for fast mobile robots in cluttered environments. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 572–577 (1990)
3. Borenstein, J., Koren, Y.: The vector field histogramFast obstacle avoidance for mobile robots. IEEE J. Robotics and Automation 7(3), 278–288 (1991)
4. Elfes, A.: Using occupancy grids for mobile robot perception and navigation. Computer 22(6), 46–57 (1989)
5. Fox, D., Burgard, W., Thrun, S.: Controlling synchro-drive robots with the dynamic window approach to collision avoidance. In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 1996 (1996)
6. Fox, D., Burgard, W., Thrun, S.: The dynamic window approach to collision avoidance. IEEE Robotics and Automation (1997)
7. Göller, M., Kerscher, T., Zöllner, J.M., Dillmann, R.: Behavior Network Control for a Holonomic Mobile Robot in Realistic Environments. In: CLAWAR 2008 (2008)
8. Göller, M., Steinhardt, F., Kerscher, T., Zöllner, J.M., Dillmann, R.: RFID Transponder Barriers as Artificial Landmarks for the Semantic Navigation of Autonomous Robots. In: CLAWAR 2008 (2008)
9. Khatib, O.: Real-time obstacle avoidance for robot manipulator and mobile robots. Internat. J. Robotics Res. 5(1), 90–98 (1986)
10. Khatib, O.: Elastic Bands: Connecting Path Planning and Control. Stanford University (1993)
11. Khatib, M., Chatila, R.: An extended potential field approach for mobile robot sensor-based motions. In: Proc. Int. Conf. on Intelligent Autonomous Systems, IAS 2004 (1995)
12. Koren, Y., Borenstein, J.: Potential field methods and their inherent limitations for mobile robot navigation. In: Proc. IEEE Internat. Conference Robotics and Automation (1991)
13. Sattel, T., Brandt, T.: Ground Vehicle Guidance along Collision-Free Trajectories using Elastic Bands. In: 2005 American Control Conference, Portland, OR, USA, June 8-10 (2005)
14. Thrun, S.: Learning Occupancy Grids with Forward Sensor Models. Carnegie Mellon University (2002)
15. Ulrich, I., Borenstein, J.: VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots. In: Proc. of the 1998 IEEE Int. Conf. on Robotics and Automation Leuven, Belgium (May 1998)