

Flexible Path Planning for Nonholonomic Mobile Robots

Birgit Graf, José Manuel Hostalet Wandosell, Christoph Schaeffer
Fraunhofer Institute Manufacturing Engineering and Automation (IPA)
Nobelstrasse 12, 70569 Stuttgart, Germany
Email: [{btg,jwh,cfs}@ipa.fhg.de](mailto:btg_jwh_cfs@ipa.fhg.de)

Abstract

This work presents a flexible path planning method for nonholonomic mobile robots. An intelligent planner based on a static map of the robot's environment has been developed. It comprises the possibility to set current robot properties on the fly and can therefore be used for many different mobile robots. The generated path is smoothened and eventually modified in reaction to dynamic obstacles or other external forces, using the method of elastic bands. This common algorithm has been extended for robots meeting the restrictions of a Dubin's car (nonholonomic robot that can only move forward).

1. Introduction

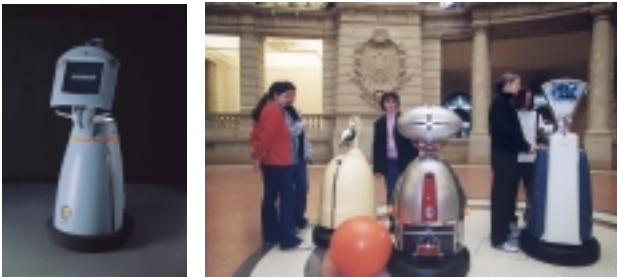


Figure 1. Care-O-bot and museum robots

Care-O-bot (Figure 1) is the prototype of a robotic home care system [12] developed at Fraunhofer IPA. A first mobile platform has been built in 1998. Care-O-bot has already proven its ability to operate safely and reliably in public environments. Three robots based on the same hardware platform have been installed in March 2000 for constant operation in the "Museum für Kommunikation Berlin" where they autonomously move among the visitors, communicate to and interact with them [1][2][14].

Up to now, the robots only navigated towards a given target position. Obstacle avoidance was done by modifying the calculated velocity vector towards an intermediate position lying between the given target and

the current robot position, using the reactive obstacle avoidance algorithm PolarBug [13].

In order to enable more complex operations of the vehicle, an intelligent path planner, including the possibility to reactively modify a path due to dynamic obstacles was needed. Furthermore, to make the algorithm useful for different kinds of robots, it was necessary to include the possibility to set geometric and kinematic properties for the robot on the fly. A new navigation module has been designed and implemented which features path planning and path modification for different nonholonomic mobile robots in real time.

2. Background

A large number of methods for solving the path planning problem based on an environment map for mobile robots are known. They are described in [8]. The following approaches can be differed:

Roadmap methods

The roadmap approach provides a robot with a collection of path segments leading it around static obstacles. This path is calculated by connecting the initial and the goal configuration of the robot with a roadmap that can be built in several ways. The Visibility Graph is built by connecting the initial and goal configuration with the edges of all obstacles in the given map. The Voronoi Diagram leads through the middle of available corridors between obstacles.

Cell Decomposition

Cell decomposition methods divide the robot's free space into several regions, so called cells. The connectivity graph is built by connecting adjacent cells. A channel leading from initial to goal configuration through the graph can then be computed. A path can be chosen as, for example, leading through the midpoints of the intersections of two successive cells.

Potential Field

Potential field methods divide the free space into a fine regular grid and search this grid for a free path. Different potentials are assigned to the cells of the grid whereas

“attractive” potentials are given to cells close to the robot’s goal, “repulsive” potentials are assigned to obstacles. A path is constructed along the most promising direction.

Nonholonomic planners

In a nonholonomic planner, the path is created as a set of maneuvers, which take into account the geometric and kinematic constraints of the robot. Different approaches have been developed using a random planner [9] or nonholonomic graphs.

3. Navigation Module

If the user specifies a target within the known operation area of the robot, the latter must find and follow the best path leading to this position. Hereby “best” does not in any case mean shortest – safety issues might require the path to e.g. not include sharp turns or to keep a maximum distance to obstacles [3]. Furthermore, restricted operation areas have to be considered. Path modifications due to dynamic obstacles or user request are required.

3.1. Realization of the Path Planner

An intelligent path planning system based on a static map of the robot’s environment has been implemented. The planner is based on the algorithm presented in [6].

Firstly, wall segments given in the map are converted into polygons. Overlapping polygons are joined using a clipping algorithm. This step is executed only once, at the initialization of the robot.

The Visibility Graph is a semi-free path (the robot may touch obstacles) created by connecting an initial robot configuration to the goal configuration passing through the vertices of polygonal regions (obstacles).

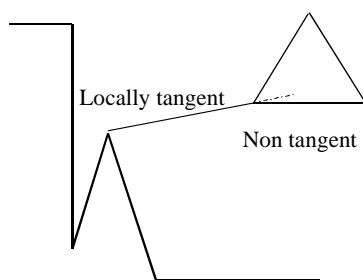


Figure 2. Creation of the Reduced Vertex Visibility Graph

In order to reduce the complexity of the Visibility Graph, the “Reduced Vertex Visibility Graph” (RVVG) is created (Figure 2). The RVVG is defined as the graph whose arcs are “locally tangent”. Locally tangent means, that the previous and the next vertex of a node lie on the same side of the line passing through the corresponding

arc of a polygon, but not necessarily all the polygon. The arc is locally tangent in one given vertex, but might intersect the polygon later. This reduction of the visibility graph permits to apply the path planning method in highly unstructured environments, including non-convex polygons.

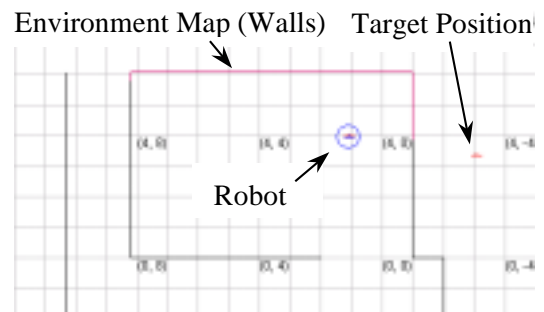
In the RVVG, the shortest path is obtained using the A* algorithm [11]. This shortest path is evaluated to find out whether it can be used as a reference to build up a feasible path for the given mobile robot. If not, the path is discarded and the next shortest path is selected and evaluated. This step is repeated until an appropriate reference path has been found.

At each point of the path, the distance to obstacles on the right and the left side of the path has to be greater than the swept area of the robot. If this condition is not satisfied, the current path is rejected. All arcs corresponding to a rejected graph are not considered for future calculations.

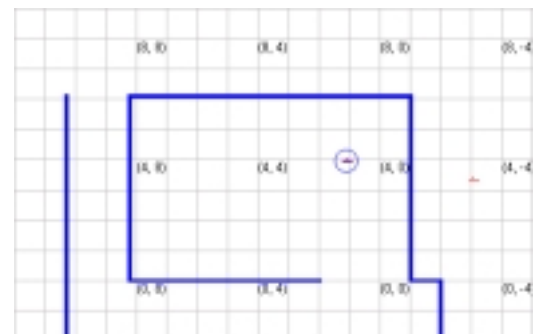
Finally the chosen path will be adjusted to fit the geometric and kinematic properties of the robot. These constraints can be modified or switched off easily for different robot platforms. Robot configurations are placed along the selected path in a way that the robot can move from one configuration to the next avoiding all obstacles given in the map.

The single steps of the described algorithm are displayed in Figure 3.

Initial robot position and map of surroundings



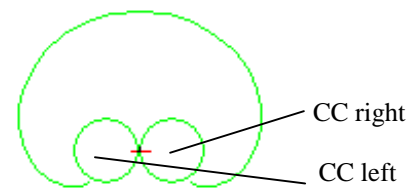
Step 1: Polygon detection



3.2. Reactive Path Modification

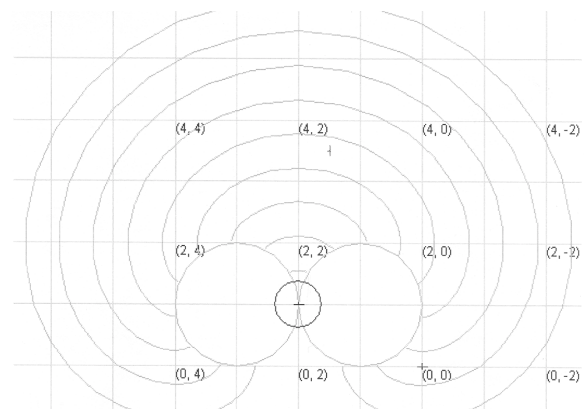
An elastic band is a deformable collision free path. It is represented as a sequence of connected “bubbles”. The bubbles are created by examining the local freedom of the robot at configurations along this path.

An elastic band for a Dubin's car (nonholonomic robot that can only move forward) has been developed.



The bubbles of the elastic band (Figure 4) represent all configurations p , which a robot with the given restrictions can reach from its current configuration q without collision (Figure 5). The radius r of a bubble is defined as the minimum nonholonomic distance from the current configuration to the closest obstacle. An appropriate metric for nonholonomic distance evaluation is used. The nonholonomic distance $B(q, r)$ from a configuration to a point or a segment was solved in [15]:

$$B(q, r) = \{ p \in \mathfrak{R}^N : \|q - p\| < r(q) \}$$



A bubble for a Dubin's car has two so called CC regions (Figure 6). Given a configuration $q(x, y, \theta)$, the

corresponding CC regions are defined by the set of configurations lying inside the circles with a radius the minimum turning radius of the robot, tangent at q to the robot orientation θ . The areas “CC left” and “CC right” are situated on the left and right side of the robot, respectively. These CC regions will be kept into account when generating optimal paths for the robot.

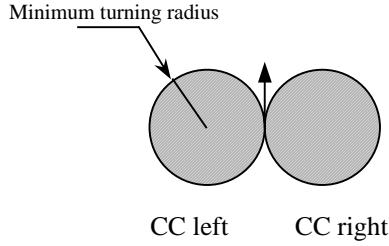


Figure 6. CC regions for a Dubin's car.

To optimize the distance between bubbles, we impose the following restriction: “The opposite CC regions between two bubbles must not overlap”. The force model of the elastic band must be able to handle this restriction (Figure 7).

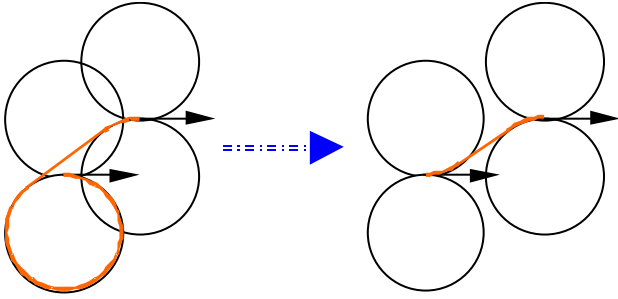


Figure 7. Optimized distance between bubbles.

Each bubble can be modified according to two kinds of forces – external and internal forces. External forces push the path away from obstacles; internal forces remove any possible slack in the path.

External forces

External forces move the bubble band away from obstacles. They are calculated creating a local potential field $\mathcal{O}(x,y)$ as the maximum potential generated by all polygons, or – what is the same – the potential generated by the closest polygon.

$$\mathcal{O}(x,y) = \max \{ \mathcal{O}_i(x,y) \}, i = 1,2,3...$$

For d being the distance to the nearest obstacle, w being the limit specified for obstacles to be considered, the following potentials are calculated for each polygon:

$$\phi(x,y) = \tan\left(\frac{\pi}{2}\left(1 - \frac{d}{w}\right)\right) \quad (d < w)$$

$$\phi(x,y) = 0 \quad (d \geq w)$$

The generated potentials come close to infinite near the obstacles ($d \rightarrow 0$). The potentials become zero when the distance to all obstacles is larger than the limit w .

The external force F_{ext} affecting each bubble is the negative gradient vector of the potential field, evaluated in the origin (x,y) of the bubble.

$$F_{ext} = -\nabla \mathcal{O}(x,y)$$

Internal forces

Internal forces contract the band and push away the opposite CC regions between bubbles. A model with three forces has been proposed successfully (Figure 8).

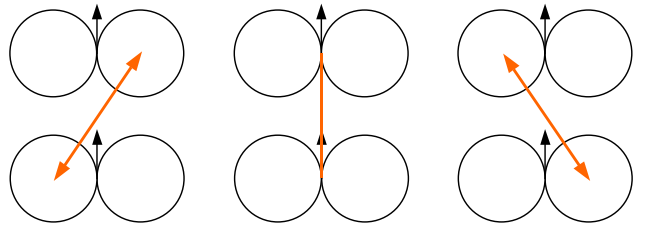


Figure 8. Internal forces between bubbles.

We denote “RL-forces” the forces, which push away the left CC region of the current bubble from the right CC regions of the previous and next bubble. In the same way, we denote “LR-forces” the forces, which push away the right CC region of the current bubble from the left CC regions of the previous and next bubble. We denote “C-forces” the forces of attraction between centers of adjacent bubbles.

With q_L , q and q_R being the center of the left CC region, the bubble and the right CC region respectively, the following equations specify the forces system.

$$LR_i = \frac{q_{Li} - q_{Ri-1}}{|q_{Li} - q_{Ri-1}|} \frac{1}{r_{i-1}} P(q_{Li}, q_{Ri-1}) + \frac{q_{Li} - q_{Ri+1}}{|q_{Li} - q_{Ri+1}|} \frac{1}{r_i} P(q_{Li}, q_{Ri+1})$$

$$RL_i = \frac{q_{Ri} - q_{Li-1}}{|q_{Ri} - q_{Li-1}|} \frac{1}{r_{i-1}} P(q_{Ri}, q_{Li-1}) + \frac{q_{Ri} - q_{Li+1}}{|q_{Ri} - q_{Li+1}|} \frac{1}{r_i} P(q_{Ri}, q_{Li+1})$$

$$C_i = \frac{q_{i-1} - q_i}{r_{i-1}} + \frac{q_{i+1} - q_i}{r_i}$$

$$F_i = K_r \cdot (LR_i + RL_i) + K_c \cdot C_i$$

The resulting internal force affecting each bubble is the sum of its LR, C and RL forces. LR and RL forces are multiplied with a repulsion factor K_r , C forces are multiplied with a contraction factor K_c .

All forces are previously scaled depending on the radius r of the current and previous bubble. This is important to get independence of the size of the bubbles, to avoid bubble migration along the elastic band, and to avoid continuous creation and elimination of a bubble.

LR and RL forces are further scaled by the potential function $P(q_L, q_R)$. This function returns a real number depending on the distance between CC regions. The return value comes close to infinite for CC regions being too close, thereby avoiding overlapping bubbles. The potential comes close to zero for bubbles being further apart than a given maximum distance.

The orientation of a bubble is modified depending on the torque T – computed as the momentum affecting the bubble due to LR and RL forces. The angle modification $\Delta\alpha$ is calculated as this torque scaled by a rotation factor k_r .

$$T_i = LR_i \times (q_{Li} - q_i) + RL_i \times (q_{Ri} - q_i)$$

$$\Delta\alpha_i = k_r T_i$$

Virtual forces

A third kind of forces can be generated by user input, to modify the path followed by the robot. User input here means that a person can “push” or “pull” the robot in a certain direction. The forces applied are read through force/torque sensors attached to the platform and transferred directly to the elastic band algorithm.

Disconnected/Redundant bubbles

The origin of a bubble must always be inside the previous one, except for the first bubble of the band. When the origin of a bubble is outside all other bubbles, it is called “disconnected”.

To find out whether a bubble is disconnected, the radius of the previous bubble r_{i-1} , is compared with the nonholonomic distance between origins $nhd(p_{i-1}, p_i)$. A small margin value ϵ_c is used to avoid creation-elimination loops. When a bubble is disconnected, the shortest path between the current and previous bubble is calculated and a new bubble is created in the middle.

Disconnection condition:

$$nhd(p_{i-1}, p_i) \geq r_{i-1} - \epsilon_c$$

When a bubble is detected between two connected bubbles, it is “redundant” and can therefore be deleted from the band. A small margin value ϵ_o is used to avoid creation-elimination loops.

Redundancy condition:

$$nhd(p_{i-1}, p_{i+1}) \leq r_{i-1} - \epsilon_o$$

Elastic band creation

The elastic band is initialized with a small number of configurations. Then, the band is checked to detect disconnections. When a disconnection is detected, a new bubble is created. This process is repeated until all bubbles are connected. A connected bubble band for a Dubin’s car is displayed in Figure 9.

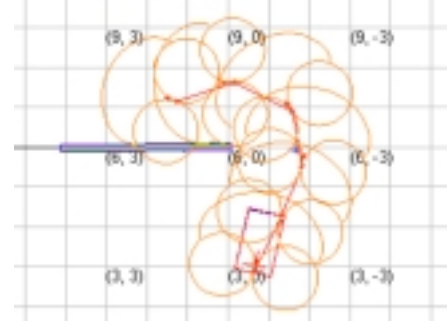


Figure 9. Connected bubble band for a Dubin’s car

Elastic band modification

The elastic band is modified cyclically in three steps:

1. Check for disconnections
2. Check for redundancy
3. Modify configurations

Each bubble is modified according to the given forces, external, internal and virtual:

$$F_{res} = F_{ext} + F_{int} + F_{virt}$$

$$Pos_{new} = Pos_{Old} + \delta F_{res}$$

A bubble might change its position due to the given forces. For the modified bubble configuration, the size of the bubble and the nonholonomic distance to the obstacles must be recalculated.

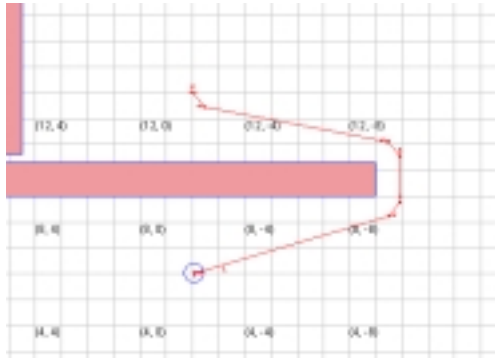
Smoothing

Once the bubble band is set, the path to execute is the concatenation of all bubbles’ centers. A Bézier algorithm is used to track the path smoothly.

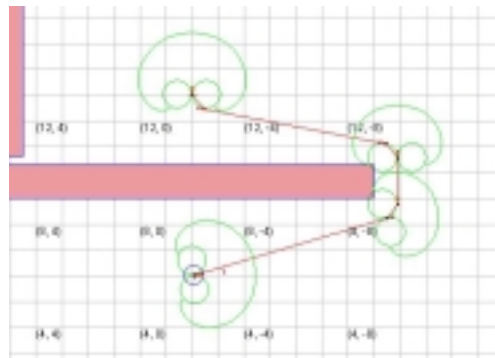
4. Experimental Results

Some experimental results of the elastic band algorithm for a Dubin’s car are displayed in Figure 10, Figure 11, and Figure 12.

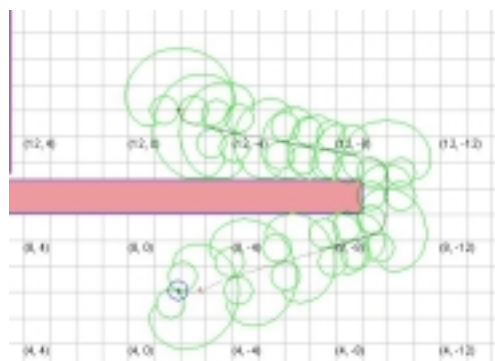
Step 1: The planner returns four configurations.



Step 2: A bubble is created in each configuration



Step 3: The band is covered with connected bubbles



Step 4: The band is optimized

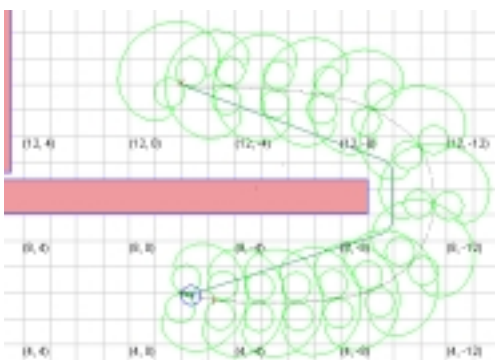
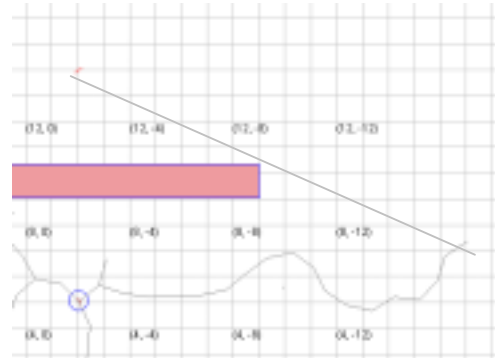
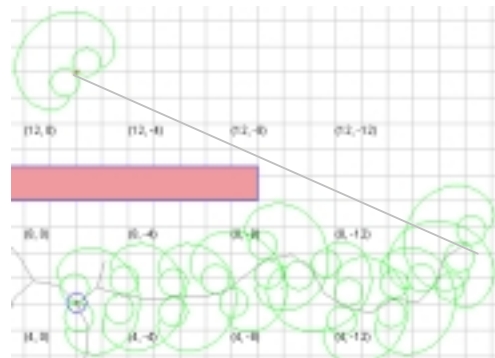


Figure 10. Path planner as described in section 3.1

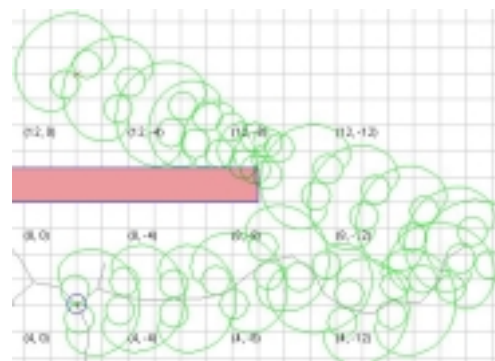
Step 1: Path returned by a random planner



Step 2: A bubble is created in each configuration



Step 3: The band is covered with connected bubbles



Step 4: The band is optimized

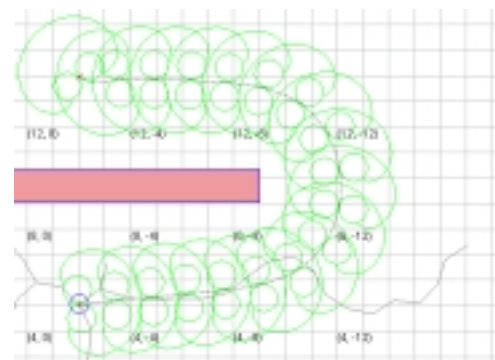


Figure 11. Random planner based on [9]

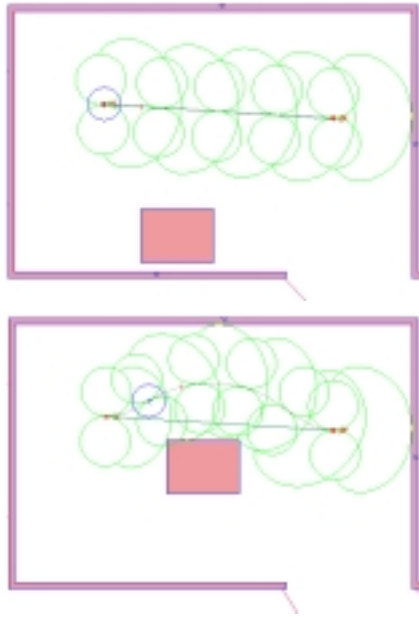


Figure 12. Dynamic obstacle avoidance

5. Conclusion and Outlook

The implementation of a flexible path planning and path modification algorithm has been described. The algorithm can be used on different kinds of robots, geometric and kinematic properties can be activated and deactivated on the fly.



Figure 13. Intelligent walking aid prototype and cleaning robot

The algorithm has successfully been implemented on the intelligent mobile home assistant Care-O-bot, which is also used as an intelligent walking aid [4] (Figure 13). Reactive path modification not only due to dynamic obstacles but also in reaction to user input through force sensors attached to the walking handles while following a path is included.

The dynamic path modification has also been implemented on a cleaning machine, which uses its own path planner but employs the elastic band method to modify this path when detecting dynamic obstacles.

The described force system model is very efficient and the algorithm is very fast. When running the program on a Pentium III processor under Windows NT4, the local modification of the elastic band in the region of view of the robot takes less than 100 milliseconds.

For future applications we are planning to apply the random path planning and dynamic path modification methods on a mobile manipulator.

6. Acknowledgements

Part of this work has been supported by the MORPHA-project [10] funded by the German Ministry of Education and Research (bmb+f) under grant 01IL902G/9.

7. References

- [1] Fraunhofer IPA: "Entertainment Robotics Homepage", <http://www.ipa.fhg.de/300/320/323/EntertainmentHome.php3>, 2000
- [2] Graf, B.; Schraft, R.D.; Neugebauer, J.: "A Mobile Robot Platform for Assistance and Entertainment." In Proceedings of ISR-2000, Montreal, pp. 252-253.
- [3] Graf, B., Haegele, M.: "Dependable Interaction with an Intelligent Home Care Robot". In Proceedings of ICRA-Workshop on Technical Challenge for Dependable Robots in Human Environments, 2001.
- [4] Graf, B.: "Reactive Navigation of an Intelligent Robotic Walking Aid". Accepted to ROMAN-2001.
- [5] Jaouni, H.; Khatib, Maher; Laumond, J.P.: "Elastic Bands For Nonholonomic Car-Like Robots: Algorithms and Combinatorial Issues." In: 3rd International Workshop on the Algorithmic Foundations of Robotics (WAFR'98), Houston, 1998.
- [6] Jiang, Kaichun; Seneviratne, Lakmal D.; Earles, P.P. W. E.: "A shortest Path Based Path Planning Algorithm for Nonholonomic Mobile Robots." In: Journal of Intelligent and Robotic Systems, 24th Ed (1999), pp. 347-366.
- [7] Khatib, Maher; Jaouni, H.; Chatila, R.; Laumond, J.P.: "Dynamic path modification for car-like nonholonomic mobile robots." In: IEEE International Conference on Robotics and Automation, Albuquerque, 1997, pp. 2920-2925.
- [8] Latombe, J.-C.: "Robot Motion Planning", Kluwer Academic Publishers, UK, 1996.
- [9] LaValle, S.M., Kuffner, J.J.: "Rapidly Exploring Random Trees: Progress and Prospects". In proceedings of the Workshop on the Algorithmic Foundation, 2000.
- [10] "MORPHA - intelligente antropomorphe Assistenzsysteme", <http://www.morpha.de>, 2000.
- [11] Morris, John: "Data Structures and Algorithms: Dijkstra's Algorithm", Course Notes PLDS210, The University of Western Australia, 1998, <http://swwww.ee.uwa.edu.au/~plsd210/ds/dijkstra.html>.
- [12] Schaeffer, C.; May, T.: "Care-O-bot: A System for Assisting Elderly or Disabled Persons in Home Environments". In Proceedings of AAATE-99, Düsseldorf, 1999, pp. 340-345.

- [13] Schraft, R.D; Graf, B.; Traub, A.; John, D.: „PolarBug – ein effizienter Algorithmus zur reaktiven Hindernisumfahrung“. In Proceedings of AMS 2000.
- [14] Schraft, R.D.; Graf, B.; Traub, A.; John, D.: „A Mobile Robot Platform for Assistance and Entertainment“. In Industrial Robot Journal, Vol. 28, 2001, pp. 29-34.
- [15] Vendittelli, Marilena; Laumond, Jean-Paul; Nissoux, Carole: “Obstacle Distance for Car-Like Robots”. In IEEE Transactions on Robotics and Automation, Vol. 15, N°4, August 1999.
- [16] David M. Mount: ANN Approximate nearest neighbors” Department of computer Science University of Maryland. 1998
- [17] Sean Quinlan; Oussama Khatib: “Elastic bands: Connecting Path Planning and Control”. ”. In IEEE Transactions on Robotics and Automation. 1993.