BEHAVIOR-BASED OBSTACLE DETECTION AND AVOIDANCE SYSTEM FOR THE OMNIDIRECTIONAL WALL-CLIMBING ROBOT CROMSCI

M. JUNG, D. SCHMIDT* and K. BERNS
Robotics Research Lab, University of Kaiserslautern,
Kaiserslautern, Germany
**E-mail: dschmidt@cs.uni-kl.de*
http://agrosy.cs.uni-kl.de/cromsci

Large vertical concrete structures are still a great challenge for autonomous climbing robots, which should be able to perform different service tasks like inspection or coating of the rough surface. With the behavior-based obstacle avoidance system of our robot Cromsci this paper presents a step towards such an autonomous climbing system. Main problems arise from a limited payload for environmental sensor systems. Therefore a special sensor setup and internal representation of the environment has to be found. In this paper we will show the overall structure of our climbing robot Cromsci using negative pressure adhesion and omnidirectional wheels for locomotion and focus on its sensor systems and the behavior-based components for obstacle avoidance.

Keywords: Climbing Robot, Behavior-Based, Obstacle Avoidance, Laser Scanner, Local Mapping

1. Motivation and State of the Art

Mobile service robots nowadays are used in many applications. Especially in hazardous and for humans unreachable environments they are applied very often to enhance human's capacity to act. One of these tasks is the inspection of large concrete buildings like dams or bridge pylons which normally utilizes complex access devices like gondolas or special cranes. Climbing robots can be used to inspect concrete buildings area-wide and without endangering human technicians.

In this context our climbing robot project focusses on the development of a (semi-)autonomous robot called Cromsci[a] (see fig. 1) which should be

---

[a] Climbing RObot with Multiple Sucking Chambers for Inspection tasks

able to perform the presented task.[1] In contrast to other climbing robots using negative pressure adhesion[2–4] our system is equipped with an adhesion system of multiple suction chambers combined with three omnidirectional wheels. The pressure inside each of the seven vacuum chambers is close-loop controlled to adjust the desired total downforce of the robot. The omnidirectional drive consists of three unsprung steerable and driven wheels with a maximum speed of $9.63 m/min$ for high maneuverability to reach all places on the building and for fast and continuous motion on the plane. Integrated load cells measure forces and torques at the contact point between wheel and wall. In the last years the research was focussed on the development of a robust control mechanism for safe adhesion. This includes the close-loop pressure controller, special inflatable sealings in a shape of spokes[5] but also a close-loop motion control to reduce wheel slip and wear.[6]

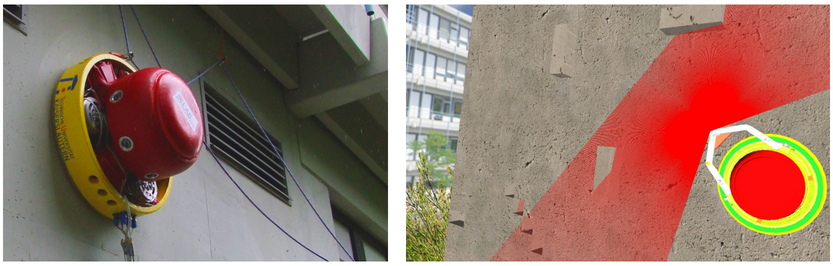


Fig. 1. Robot prototype CROMSCI driving at a concrete wall secured by a rope and supplied via umbilical cord (left) and 3d-simulation with visualization of scan lines (right)

Based on the finished robotic system we are now able to add methods for evading obstacles automatically to allow a (semi-)autonomous motion. In contrast to other mobile robots using planar laser range sensors[7,8] our omnidirectional climbing robot has special requirements concerning an obstacle avoidance system. At first the complete obstacle avoidance system is very safety critical because an unrecognized hole could lead to a loss of adhesion and a total damage of the robotic system. Unfortunately the number of environmental sensors is low due to limited payload and mounting space which reduces the perception capabilities. At second the robot has to build up an abstract local map which has to be robust against odometry errors to memorize object positions close to the vehicle. Another interesting point arises from the omnidirectional movement which allows different kinds of evasion strategies, but also needs to work 360° all around the robot.

2. Perception of the Environment

Before any obstacle avoidance can be executed, one has to receive information about the environment and objects within a certain range around the robot. Figure 2 shows the overall structure of the obstacle avoidance system including obstacle detection, local mapping, localization and the safety behaviors for obstacle avoidance.

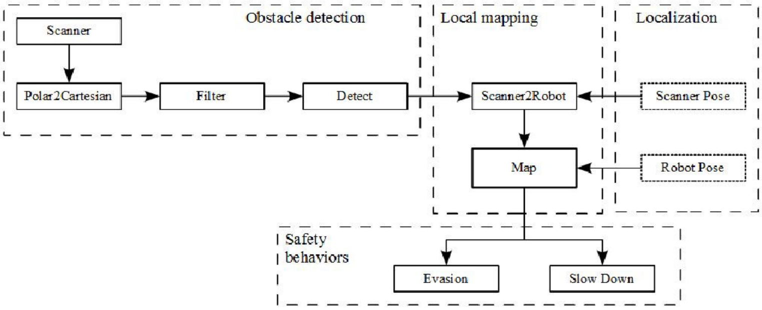


Fig. 2. Structure of the obstacle avoidance system

2.1. *Obstacle detection*

In vertical environments like concrete walls robots only have to deal with static obstacles which can be classified in two major types: Positive obstacles like superstructures and negative ones like deep cracks or holes. This distinction is caused by a requirement concerning the manipulator arm, because it is necessary to prevent a collision with positive obstacles but it is obviously no problem if the arm reaches over negative ones.

Considering weight restrictions CROMSCI uses only a pitched lightweighted laser scanner by Hokuyo attached to the manipulator arm ahead to scan the surface in front (see fig. 1, right). The obstacle detection is performed by finding special edges in sensor distance data by comparing the indexed distance values x_i in scanner coordinates with the reference distance x_{ref} and offset x_{offset}. This leads to points of interest (POI) regarding eq. 1 which are given to a mapping algorithm to store the obstacle positions in a local memory whereas missing obstacles which indicate free space can be used to delete phantom obstacles.

$$POI(x_i) = \begin{cases} x_i & \text{, if } x_i < x_{\text{ref}} - x_{\text{offset}} \text{ (positive obstacle)} \\ x_{ref} & \text{, if } x_i > x_{\text{ref}} + x_{\text{offset}} \text{ (negative obstacle)} \\ 0 & \text{, else (no obstacle, free space)} \end{cases} \tag{1}$$

2.2. *Local Mapping*

Based on the fact that obstacles can only be detected in a certain distance they have to be stored within a local map used as a short-term memory. The map has to cover at least the area between the robot and the current detection distance and has to be as abstract as possible for higher robustness against odometry errors which occur because of a big wheel slip.

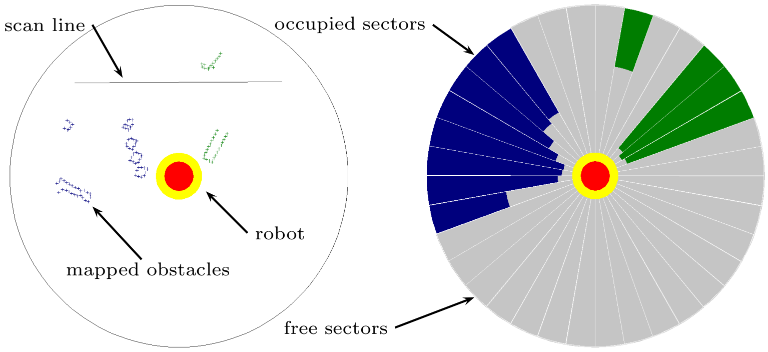


Fig. 3. Generated local obstacle map (left) with current scan line above the robot. Resulting sector map (right) for obstacle avoidance with sectors containing negative obstacles on the left and sectors with positive ones on the right side of the robot

As shown in figure 3 a circular map with a diameter of 6m containing obstacle edges is generated. This diameter results from the far-sight of the laser scanner. For obstacle avoidance these obstacle positions are transformed to a sector view which is used for reactive motion behaviors. Outside lying objects will be forgotten.

3. Behavior-based Obstacle Avoidance

Based on the map data different safety behaviors are supplied with obstacle information to either slow the robot down to prevent a collision or to let it evade the obstacle by turning away. If the information indicates an obstacle the behaviors become active and affect other basic control behaviors which are responsible for robot movement. All behaviors follow the design of the iB2C architecture[9] in which beside incoming informations and outgoing control data also additional signals are used to influence the activity of other behaviors and to signal the current behavior state. Each behavior has three state values: the *stimulation* s points out how active the behavior is allowed to be, the *activity* a shows the real amount of action and the *target rating* r how satisfied the behavior is in the current situation.

Two different reflexive behaviors are used for collision avoidance. The first one is an autonomous deceleration based on detected obstacles within a certain range. The deceleration ends up in a total stop of the motion in the observed direction when reaching a minimal distance. As described above, there is a special treatment of motions of the manipulator arm, because it only has to be considered while moving towards positive obstacles. In total there exists several instances of the basic **Slow Down Behavior** which are responsible for different movement directions to allow motion in obstacle-free directions. Each behavior receives the minimal distance value d_{min} of the sectors in the specific movement direction. Then they calculate activation $a_{slowdown}$ ranging from 0 (no stop) to 1 (full stop) as shown in equation 2 using a far distance limit d_{limit}:

$$a_{slowdown} = \begin{cases} sin\left(\left(\frac{\pi}{2} \cdot \frac{d_{min}}{d_{limit}}\right) + \pi\right) + 1 & \text{, if } d_{min} < d_{limit} \\ 0 & \text{, else} \end{cases} \tag{2}$$

Additionally an autonomous **Evasion Behavior** has been built up to turn away the robot to a more safely direction. The reaction range is important to leave enough space for maneuvers. The evasion behavior calculates a weighted difference between sectors on the left and on the right to get the best and easiest way. To achieve this, two weights w for the left and right side of the robot are calculated pointing out the amount of obstacles on this side. The weight value for each side is determined as shown in equation 3 by summing up values of all n sectors ranging from frontal $(i = n - 1)$ to most-left respective most-right sector $(i = 0)$. d_i is the closest distance value of sector i whereas d_{limit} is a far distance limit. Two aspects have to mentioned here: The sector position has an influence on the weight (sectors in front have a higher weight than those at the side of the robot) and a sector is only taken into account, if the measured distance d_i is smaller than the limit d_{limit}.

$$w = \sum_{i=0}^{n-1} \begin{cases} \frac{i+1}{2 \cdot n} \cdot \left(1 - \frac{d_i}{d_{limit}}\right) & \text{, if } d_i < d_{limit} \\ 0 & \text{, else} \end{cases} \tag{3}$$

After the calculation of the two weights w both are amplified or weakened depending on the closest measured value on the specific side. The side with the lower weight (more free space) is used for avoiding, the activity $a_{evasion}$ of the evasion behavior depends on the weight w of the blocked side and takes effect on robot turning.

The overall structure of the behavior network is very important for the control process. In normal remote control mode the motion behaviors of drive and tool center point are triggered by user commands (e.g. via joystick) and generate motions. For obstacle avoidance these basic motion behaviors are influenced by activating or inhibition. The evasion behavior for example activates the turning behaviors depending on its own activity, the behaviors for straight motion are inhibited by the corresponding slow down behavior. By this a complex total robot behavior can be achieved as shown in the experiments.

4. Experiments and Test Results

The experiments are performed with the real robot CROMSCI on the ground and show different behaviors, the robot trajectory and the local obstacle map. For each behavior all state values *stimulation*, *activity* and *target rating* (see section 3) are plotted. In fig. 4 the automatic deceleration of the robot has been tested. The robot was simply set to drive forward with half speed. Therefore the stimulation of *Drive Forward* is set to $s = 0.5$. **Slow Down Forward** is activated ($s = 1.0$) but not active ($a = 0.0$). The target rating $r = 0.0$ points out that it is satisfied at the beginning of the experiment. While the robot drives forward starting at position $(0, 0)$, the distance to the obstacles decreases and the *Slow Down Forward* behavior becomes 'unhappy' and inhibits the forward driving by its own activity. One can see in the robot trajectory that the robot slows down until it performs a full stop to prevent a collision. In this situation one can free the robot only by driving backwards or turning away.

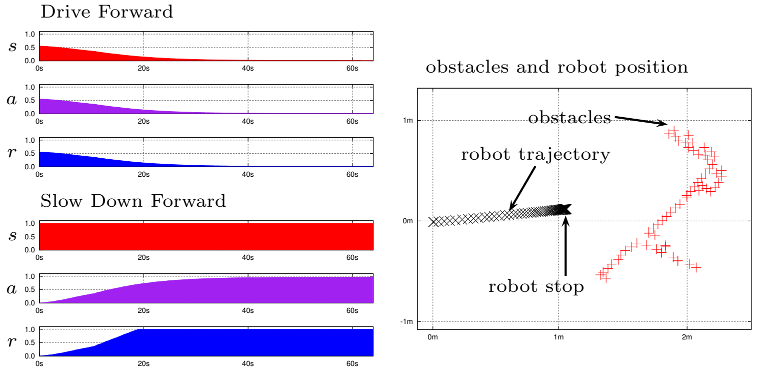
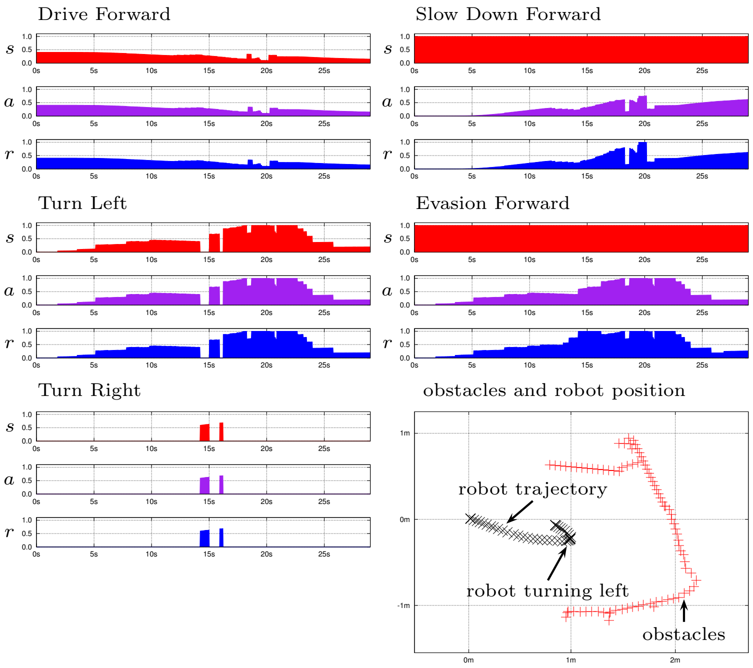


Fig. 4. The slow down behavior inhibits the forward driving by its own activity, the robot slows down and stops (with *stimulation* s, *activity* a and *target rating* r)

In the second experiment a more complex situation is given and more behaviors are involved. For a much better robot reaction the **Evasion Forward** behavior has been set up as presented in section 3. Figure 5 shows the robot driving into a dead end (bottom right) and the basic motion behaviors for driving forward and turning to the left and right. One can see the interaction of the behaviors as the robot detects first obstacles. The activity of *Evasion Forward* increases and stimulates the *Turn Left* behavior depending on the distances of obstacles on the left and right side of the robot. In the meantime the *Slow Down Forward* behavior becomes active because of the decreasing obstacle distance and slows down *Drive Forward*. During the time from $15s$ to $23s$ the robot nearly stays at the same position and turns to the left until the way in driving direction is free again. For a short period of time around $15s$ the robot was undecided which turning direction is the best, but he managed this situation, turned left and did not start shaking from left to right.



Drive Forward

Slow Down Forward

Turn Left

Evasion Forward

Turn Right

obstacles and robot position

Fig. 5. Robot behavior while driving into a dead end: Forward driving is reduced by the slow down behavior, the evasion behavior automatically triggers a turning of the robot by its own activity. As a result the robot leaves the dead end and continues driving

5. Conclusion and Outlook

We presented a new control system for obstacle avoidance for a wall-climbing robot. By this the robot is able to avoid critical situations by turning away from obstacles and holes and to automatically slow down, if it gets to close. The real-world experiments on the ground are very promising, unfortunately the system can not be used while the robot is driving up a wall because of a too high wheel slip and bad odometry. To enhance the obstacle avoidance an additional localization system will be developed to get more reliable information about the robot pose.

References

1. C. Hillenbrand, D. Schmidt and K. Berns, Cromsci - development of a climbing robot with negative pressure adhesion for inspections, in *Industrial Robot Volume 35 Issue 3*, (Emerald Group Publishing Ltd., May 13 2008).
2. H. Zhang, J. Zhang and G. Zong et al., Sky cleaner 3 - a real pneumatic climbing robot for glass-wall cleaning, in *IEEE Robotics And Automation Magazine*, March 2006.
3. D. Longo and G. Muscato, The alicia3 climbing robot, in *IEEE Robotics And Automation Magazine*, March 2006.
4. B. Luk, L. Liu and A. Collie, Climbing service robots for improving safety in building maintenance industry, in *Bioinspiration and Robotics: Walking and Climbing Robots*, September 2007 pp. 127–146.
5. C. Hillenbrand, D. Schmidt, K. Berns, T. Leichner, T. Gastauer and B. Sauer, Development of a sealing system for a climbing robot with negative pressure adhesion, in *10th International Conference on Climbing and Walking Robots (CLAWAR)*, July 16-18 2007.
6. C. Marx, D. Schmidt, C. Hillenbrand and K. Berns, Force and traction controlled propulsion of the omnidirectional climbing robot cromsci, in *12th International Conference on Climbing and Walking Robots (CLAWAR)*, September 9-11 2009.
7. R. Philippsen and R. Siegwart, Smooth and efficient obstacle avoidance for a tour guide robot, in *IEEE International Conference on Robototics and Automation (ICRA)*, 2003.
8. A. Aboshosha and A. Zell, Robust mapping and path planning for indoor robots based on sensor integration of sonar and a 2d laser range finder, in *IEEE 7th International Conference on Intelligent Engineering Systems (INES)*, 2003.
9. M. Proetzsch, T. Luksch and K. Berns, The behaviour-based control architecture iB2C for complex robotic systems, in *Proceedings of the 30th Annual German Conference on Artificial Intelligence (KI)*, (Osnabrück, Germany, 2007).