Obstacle avoidance for wheeled mobile robotic systems

Obstacle Avoidance for Wheeled Mobile Robotic Systems (Literature Exploration)

J.J.A.M. Keij
Report No. DCT 2003.10

Eindhoven, 17 February 2003

Engineering Thesis Committee:

prof.dr. H. Nijmeijer (chairman)
dr.ir. H.A. van Essen (coach)
dr.ir. P.F. Lambrechts
dr.ir. W.J. Witteman
dr.ir. M.J.G. van de Molengraft

Technische Universiteit Eindhoven
Department of Mechanical Engineering
Dynamics and Control Group

Contents

1 Literature Exploration

1.1 Introduction

In this reference exploration report an introductional overview will be given in the field of wheeled mobile robotics, more specific into the area of motion planning, trajectory generation and obstacle avoidance. Please note that this is a draft report and for internal use only.

Although trying to be as complete as possible, the current field of wheeled mobile robotics has grown too big to give a complete comprehensive overview. This report should be placed in the perspective of the current project on obstacle avoidance for wheeled mobile robotic systems at the Dynamics and Control group at the Eindhoven University of Technology. Although this reference exploration is not solely based on the available robot, the focus will be on this class of systems, the neighboring classes will be addressed more briefly where necessary.

In the early 1970's robots where already able to walk, drive and live completely autonomous in cooperation with humans; well at least in science fiction they were. Although fiction is most of the time a good drive for science, autonomous natural acting robots are still a giant leap away for humans to realize, though especially on the field of robotic motion planning great advances are made by scientists recently.

Wheeled mobile robots can increasingly become important in different fields in the near future. Robots could help humans in places less suitable for humans, like deep-sea exploration, hazardous waste sites, damaged nuclear reactors and deep space exploration.

In this project a small mobile robot is available. This mobile robotic system, called The BellyBot, is described in more detail in Van den Berg [2]. This robot uses two stepper motors (differential drive robot) for propulsion and two position sensors for axis position and orientation reconstruction. The robot is powered and controlled by a external computer, power supply and data acquisition unit (TUDacs). This incorporates the main limitation to this robot; the robot has a relatively limited range because of the cable between the robot and control and power unit. The stepper motors have the big advantage of great dead-reckoning capabilities.

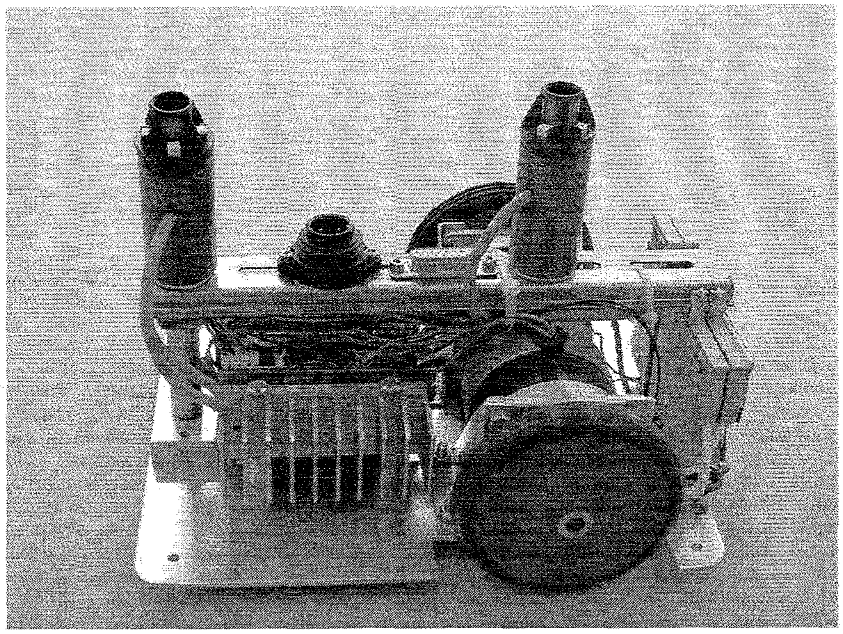


Figure 1.1: The available robotic system (BellyBot)

1.2 Approach

This literature exploration examines the status quo in the area of obstacle avoidance algorithms for wheeled mobile robotic systems. In this report first a short introduction about general robotic motion planning will be given. This will be followed by a more detailed historic overview of trajectory generation of wheeled mobile robotic systems and obstacle avoidance. Finally some examples are presented of implemented robotic systems.

2 Robotic Motion Planning

In general a motion plan consists of the kinematic trajectory for the system as well as the actuator forces that move the system along the trajectory. The actuator forces can be sometimes obtained from the given kinematics in a particularly simple way. In other instances we use the kinematic description of the system because the dynamic equations are difficult to derive. There are also cases when we employ the kinematic model of the system to abstract the details of the actuation scheme. In literature the term dynamic motion planning is used when the actuator forces are part of the computed motion plan and kinematic motion planning is used when only the kinematic trajectory for the system is computed.

Motion planning can furthermore be separated into two categories; explicit motion planning if the motion plan is computed before the motion is executed, and implicit motion planning if the trajectory and the actuator forces are computed while the system moves. Please note that if we want to optimize the performance of the motion or guarantee certain properties of the trajectory in general an explicit motion plan has to be used. If it only matters that a desired configuration is reached, implicit schemes are considered to be sufficient. The division into explicit and implicit schemes for motion planning also applies to trajectory generation in robotics. In most cases, explicit schemes are used for kinematic motion planning while implicit schemes are usually employed for dynamic motion planning.

2.1 Implicit methods

Implicit schemes only use the information about the state of the robot and the environment to compute how to move and can be interpreted as feedback mechanisms. They are very attractive from a computational point of view since no processing is required prior to the motion. The simplest scheme, corresponding to the final position control in biological systems, is to make the set-point for the joint controllers equal to the desired final position in the joint space and let the error between the current position and the set-point drive the robot. A modification of this scheme where the velocity during the motion is appropriately shaped is often provided on industrial robots as one of the possible modes of motion, but it is hardly useful for large amplitude motions. One of the reasons is that the shape of the trajectory in the taskspace depends on the location of the start and the end-configuration within the joint space. If obstacles are present in the workspace of a robot, it is difficult to predict whether the robot will avoid them or not. Another possibility is to define a potential function with the equilibrium point at the goal configuration. The actuators of the robot are programmed to generate the force dictated by the potential field, driving the robot towards the goal configuration. This scheme is much more flexible than the final position control since the potential field that guides the motion can be chosen. It is also very easy to implement obstacle avoidance by assigning a repulsive potential to each obstacle. This method has evolved in a method, in which the range of the repulsive potential is limited, so that only the obstacles that are close to the robot will affect the motion. The robot thus only needs to know local information about the environment. If the potential is defined in the joint space, the problem of kinematic redundancy can be resolved as well. The main drawback of the potential function method is that there may exist local minima that can trap the robot. Rimon and Koditschek [36] demonstrated that a potential (navigation) function can be constructed which has a global minimum and for which all other equilibrium points are saddle-points (unstable equilibria) that lie in a set of measurement zero. However, constructing such a navigation function requires complete knowledge of the space topology and many advantages of the original potential function method are lost. Another deficiency of potential fields is that the generated trajectories are usually far from being of minimal length. Finally, it is difficult to take various constraints posed by the task into account such as velocity limits or nonholonomic constraints.

2.2 Explicit methods

To compute a trajectory, explicit methods (also referred to as open-loop schemes) require knowledge of the global properties of the space. The advantage of such schemes is that task requirements can be taken into account during the planning process. The approach is also attractive from the control point of view: once the trajectory of the system is planned, the system can be linearised along this trajectory and methods from linear control theory can be used to control its motion. One possible subclass are roadmap methods, which construct a set of curves, called roadmap, that sufficiently connect the space. A path between two arbitrary points is found by choosing a curve on the roadmap and connecting each of the two points to this curve with a simple arc. Instead, cell decomposition methods divide the configuration space into non-overlapping cells and construct a connectivity graph expressing the neighborhood relations between the cells. The cells are chosen so that a path between any two points in the cell is easily found. To find a trajectory between two points in the configuration space, a corridor is first identified by finding a path between two points in the connectivity graph. Subsequently, a path in the configuration space is obtained by appropriately connecting the cells that form the corridor. The most general versions of roadmap and cell decomposition methods work for cases in which obstacles in the configuration space can be described as semi-algebraic sets. However, most practical implementations assume that the obstacles and the robot can be described as polygons. At the price of considerably increased complexity, it is also possible to extend some of the approaches to cases in which the obstacles in the environment move and sensors provide their position. A common feature of all the motion planning schemes described in this section is that they are based on discrete algorithms. In one way or another the configuration space is discretized and represented by a graph. Subsequently, trajectory planning is reduced to finding a path in this graph. These methods are purely kinematic: they only generate a trajectory in the configuration space, while the dynamics of the robot and the possible constraints on the actuator forces are not taken into account. To obtain a trajectory in the actuator space a separate mechanism must be employed. From the point of view of hierarchical organisation, they therefore assume separate planning at each of the three levels: task space, joint space and actuator space. Zefran [39] illustrates some more classifications.

3 Obstacle Avoidance

Trajectory generators for wheeled mobile robots can be interpreted as evolutions of the motion planners of general robotic systems. Though guiding mobile robots is a little different than industrial robots for example. That is the reason that this chapter will give a historic overview of the evolution of trajectory planners for wheeled mobile robots. Of course the separation into implicit and explicit methods still hold for the methods addressed in this chapter but the evolution of these methods are more thought to be more important than the rough separation into these two groups, therefore an historic overview is chosen to present some of the methods found in literature, more or less relevant to the robot in this project.

3.1 Historic overview of obstacle avoidance

In most of the methods that are presented in this overview of path planners (or trajectory generators) assumptions and/or simplifications are made. Mainly the robot is assumed to be modelled as a point capable of holonomic motion located at x in a two-dimensional environment, for simplicity although most techniques described here extend to higher dimensional spaces. Obstacles block the robot's sensors as well as its motion. In this chapter we will not discuss those in specific unless these are necessary to understand that specific method, more details can be found in the references.

According to Sharon Laubach (Jet Propulsion Lab, NASA) [27] the field of motion planning can be split by three guiding philosophies: *classical path planning*, *heuristic planning* and *"complete and correct" sensor-based path planning*. Each of these philosophies will now be discussed along with the advantages and disadvantages of each approach. The first group in this section is explicit, the second and third group implicit as stated in chapter 1. In the following overview the path planner, generates a trajectory for the robot in the *free space* which describes the allowable regions for robot traversal: the 2D environment minus the (interiors of) the obstacles.

3.1.1 Classical motion planners

Classical motion planners assume that full knowledge of the geometry of the robot's environment is known a priori. This can be seen as a serious disadvantage if this class, but on the other hand the classical planners have the useful properties of *correctness* and *completeness*.

A path is *correct* if it lies wholly within the free-space, and if the the goal is reachable, connecting the initial position with the goal. This property is quite common for most known path-planners, otherwise they will be instinctively be called *incorrect*. The second property of *completeness* is a highly desirable virtue: the planner generates a path if one is possible and halt otherwise in finite time. Latombe describes classical planners in some detail in his book, *Robot Motion Planning* [26], in which he splits the classical planners into three major categories: roadmap algorithms, cell decomposition methods and potential field approaches. The first two categories seek to create maps or channels for robot navigation.

Roadmap algorithms

Examples of roadmap algorithms include the visibility and reduced visibility (also known as tangent) graphs as described in Nilsson [32] and Latombe [26]. Other examples of roadmap techniques are Voronoi graphs [14] and Canny's silhouette method [8]. Each of these schemes constructs a set of one-dimensional curves which encapsulate the topology of the free space, and serve as a system of so called "freeways", complete with methods for entry and exit, to enable the robot to traverse from start to goal.

As illustrated in figure 3.1 this method constructs a roadmap for the complete known world surrounding the robot and far beyond. The major differences in roadmap theories lays in the local planner which selects a collision free path within the free space given by the previous constructed roadmap.

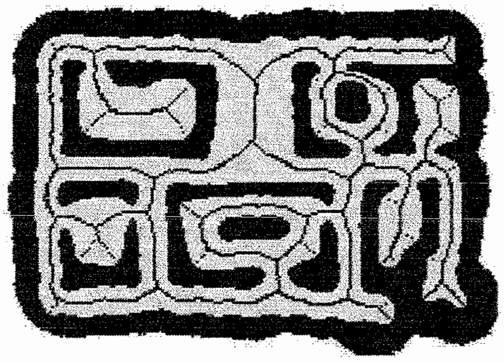
This is analog to an exploration of an urban area. The first algorithm *draws* a map of all the roads in town. The second *local planner* algorithm selects a path within this map to drive from a starting point to some desired point.

Cell decomposition algorithms

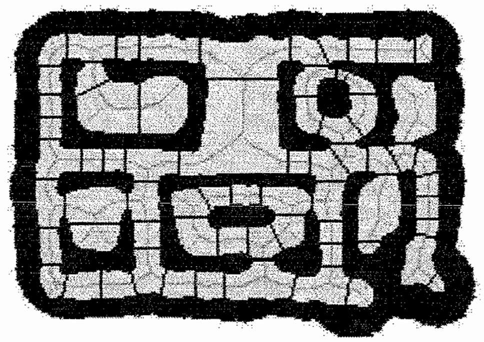
Cell decomposition algorithms come in two varieties: those which break the free-space into exact polygonal decomposition and approximate techniques which overlay a regular grid (with possible local adaptations in resolution) on the entire world model. An example of the exact variety more precisely the *trapezoidal* decompositions is illustrated in figure 3.2(a). The approximate or grid methods include applications of A^* and *quadtree* decompositions, which are also described in Latombes *Robot Motion Planning* [26].

Although the concept of cell decomposition is quite simple, the implementation is more difficult. This group of algorithms all split the space into two parts; obstacle space and free traversable space. The exact variety of this group separate the space in trapezoids or polygons. The free space is constructed by combining all the separate free subspaces.

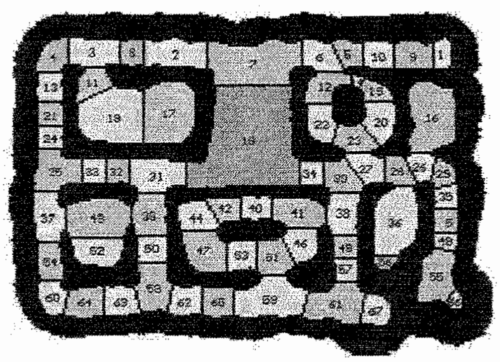
In the approximate cell decomposition algorithms the space is discretised into small cells. If an obstacle in situated within a cell, the complete cell is *disabled* for traversal and thus the complete cell is added to the obstacle subspace as illustrated in figure 3.2(b). Just like the roadmap concept this group of algorithms construct an obstacle free subspace wherein the robot will be able to traverse without encountering an obstacle. Cell decomposition algorithms do not produce *hard-to-follow* one-dimensional curves but give save corridors between obstacles. This property puts less pressure on tracking controllers, but do not completely solve the path generation problem.



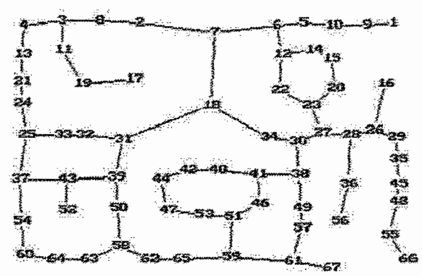
(a) Voronoi graph



(b) Critical lines graph



(c) Regions mapping



(d) Topological graph

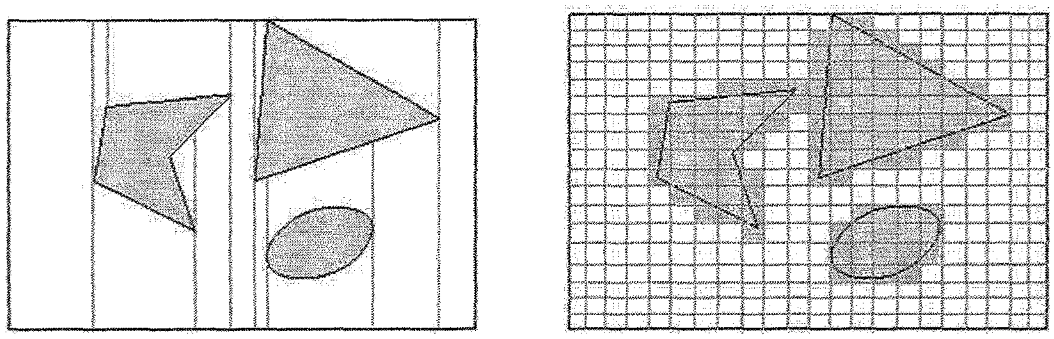
Figure 3.1: Several mappings of the roadmap algorithms

Potential field methods

Potential field methods do not explicitly map preferred routes, like the other classical path planners do. Potential field methods act as heuristics to guide the search of a grid laid over the configuration space of the robot.

In the original formulation by Khatib [23] the complete knowledge of the space was used, although it did not possess the property of completeness. Further development of the theory has produced a classical potential field algorithm which is complete, this is described by Rimon and Koditschek (*Exact Robot Navigation using Artificial Potential Functions*) [36]. Global concept of this group of methods is the virtual assignment of a positive potential to the robot, as well as to the obstacles and an attracting opposite potential to the goal. This class of methods can be illustrated by the following metaphor; The attractive potential of the goal can be depicted as a sink in a imaginary mountain scenery, where the positive potentials are depicted as peaks. The forms of the peaks and sinks can vary for various algorithms.

One of the most applied and fastest developing techniques in obstacle avoidance seems to



(a) Exact algorithm (b) Approximate algorithm

Figure 3.2: Cell-decomposition varieties

be the Virtual Potential Field method. The family of Virtual Potential Field (VPF) methods have evolved into new techniques, like the Virtual Force Field (VFF) method (Borenstein and Koren [3]) for point-like robots and the Combined Vector Field (CVF) for non-point mobile robots (Borenstein and Raschke [34]).

The Combined Vector Field method uses the combination of both VPF and VFF in such a way that the advantages of both methods are retained.

Classical path planners as a group posses the the key advantage of provable completeness and correctness. Furthermore, since the world model is known a priori, allowing these algorithms to be computed "off-line", in general the computational complexity of the classical planners can be analyzed. The cell decomposition methods have the additional advantage that they produce safe corridors between obstacles, rather than hard-to-follow one-dimensional curves adding extra pressure on complementary tracking controllers. However, analyzing implemented research projects using classical planners it can be remarked that classical planners are often impractical to implement, relying for example on geometric properties not able to be sensed easily or at all by the robot as it moves, or demanding excessive computational effort. Additionally, in most of the practical areas were mobile robots are used (especially exploration tasks) there is a priori no complete knowledge of the environment, nor will it be bounded (enough) to guarantee completeness.

3.1.2 Heuristic planning

The class of heuristic planners, such as Brooks' subsumption architecture [6] or the track arbitration schemes developed at CMU [22], [38], as well as the "Go To Waypoint" algorithm employed by the Sojourner and Rocky7 planetary Mars rovers [27] share the useful property of being able to be made sensor-based much more easily than the classical planners and can be applied to unknown terrains.

These planners dispense with the idea of creating global models of the environment in favor of "using the world as its own model" and using only local knowledge of the environments to inform the robot's reactions, usually chosen from a set of "behaviours".

A very good example of heuristic planners is the bug algorithm. This algorithm is based on the behavior of cockroaches. The most simple version can be illustrated with the following

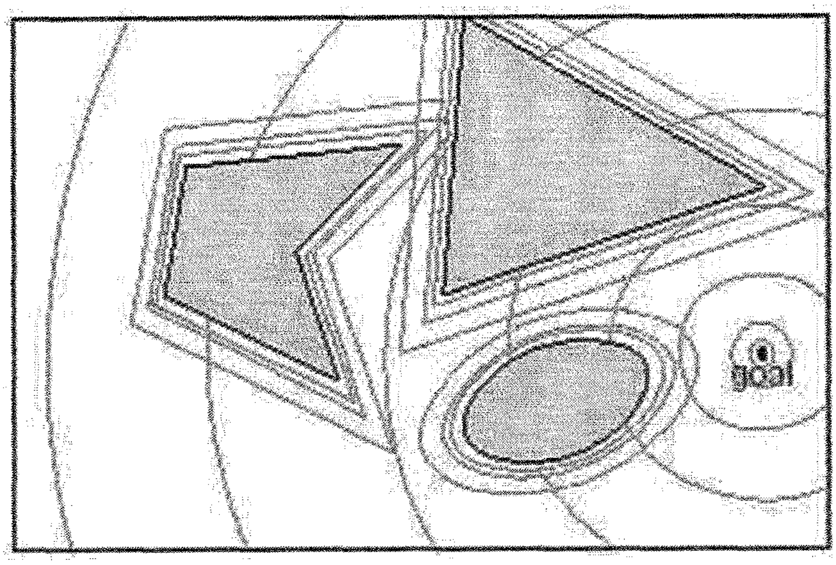


Figure 3.3: Virtual Potential Field method

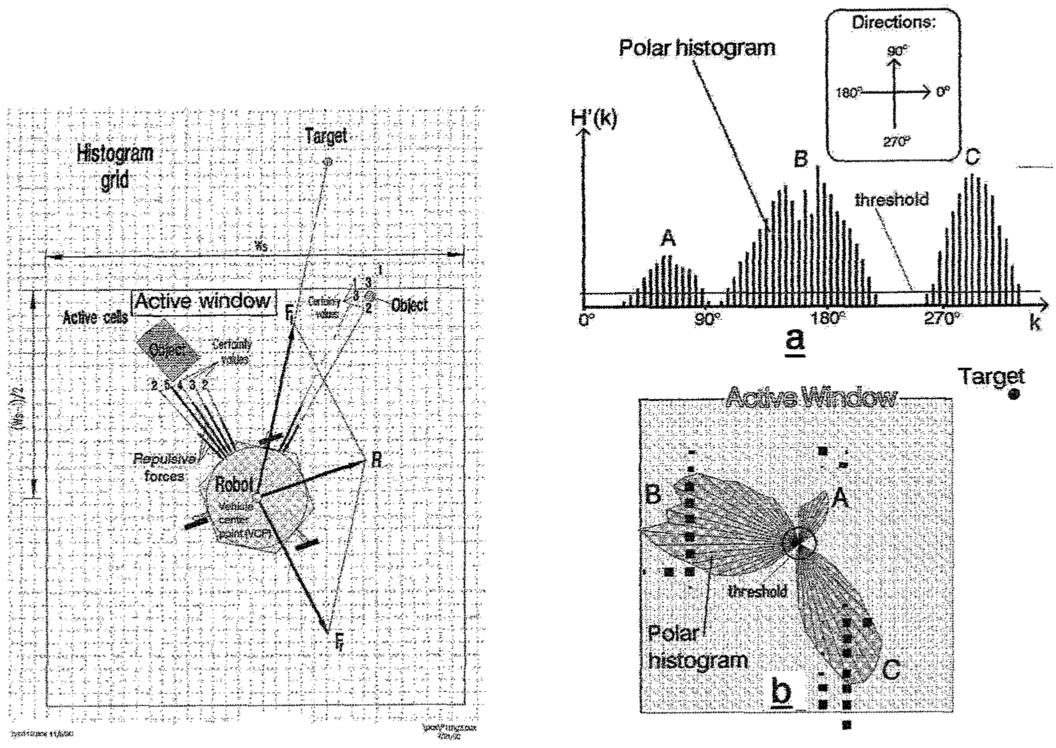
control-loop, where *target-line* is the line between the original position and the target:

```
{
until goal reached
    drive towards goal
    if robot hits obstacle
        turn left
        until robot encounters target-line
            follow obstacle contour
        end
    end
end
}
```

As can be seen in above illustrated control program the robot bases its trajectory on pre-programmed situations and criteria and execute pre-programmed commands. Although heuristic planners are designed to work well in most environment configurations, they lack completeness. There is no guarantee that the algorithm will halt, or that the robot will be able to find the goal even if a path exists. Some research projects report quite lengthy paths using this class of planners, which will be a big disadvantage in most cases.

3.1.3 "Complete and correct" sensor-based path planning

The class of "Complete and correct" sensor-based path planning can be seen as an evolution of both previous described classes, the *complete and correct* classical planners and the *sensor based* heuristic planners. This class is mainly incremental in nature: the robot senses its environment, then determines a local path segment based upon the resultant world model. After moving along the local path, the robot begins the cycle again with its sensors. Using this



(a) VFF Concept

(b) Polar histogram

Figure 3.4: Virtual Force Field method (Borenstein)

model, three distinct approaches have been explored, two of which adapt classical methods to a local sensed region.

One set of methods incrementally builds "roadmaps" within the free space in the visible area, such as Choset's Hierarchical Generalized Voronoi Graph [13] as illustrated in figure 3.6(a) and Rimon's adaptation of Canny's OPP [35].

A successful example of this method is the *Tangent Bug* algorithm, developed by Kamon, Rivlin and Rimon [19].

The second approach is based on approximate cell decomposition, filling in a grid-based world model incrementally, such as Stentz' D^* algorithm [37], [38] (Figure 3.6(b)). The third approach springs from the heuristic planners and includes the "Bug" algorithms of Lumelsky and Stepanov [31] and Rao et al. [33], which combine reactive behaviours with global parameters to reach the goal, as illustrated in figure 3.6(c).

All of these methods maintain provable properties of completeness, yet are fully applicable to unknown terrains. Both heuristic planners and sensor-based planners of this kind share the disadvantage that their computational complexity is difficult to analyse, primarily due to the algorithms' reliance upon sensor input for decision-making. For this same reason, both types of planners are subject to sensor error, and it is uncertain how such errors affect the performance of many of the methods. In particular, several schemes rely upon "good" (or "perfect") dead-reckoning ability.

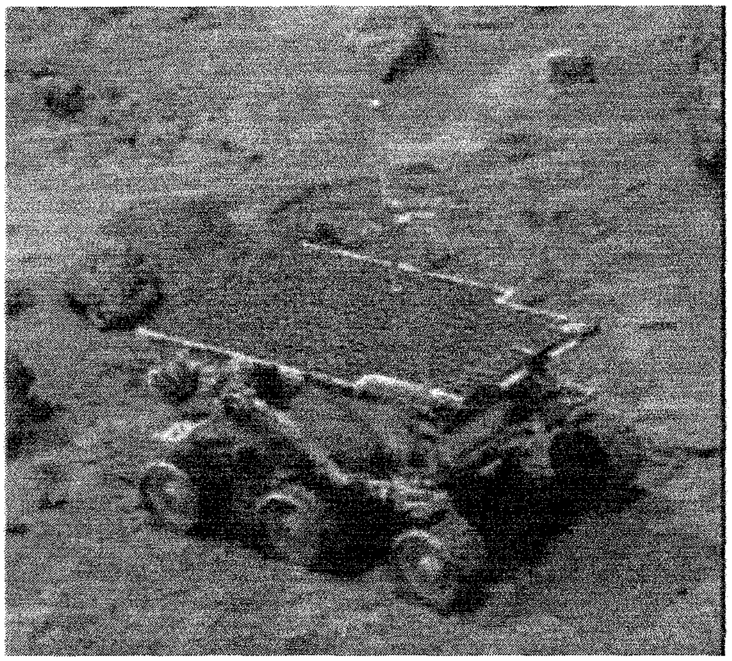
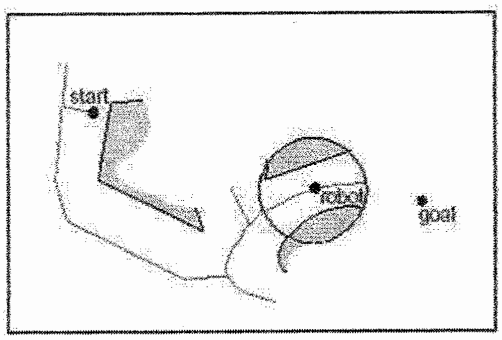
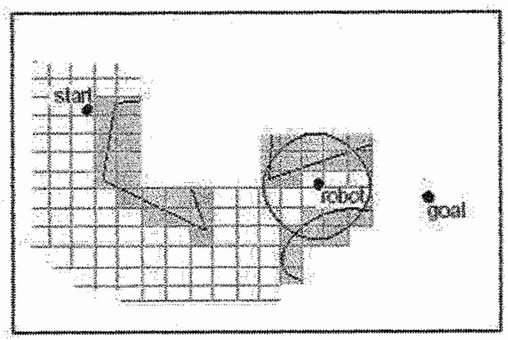


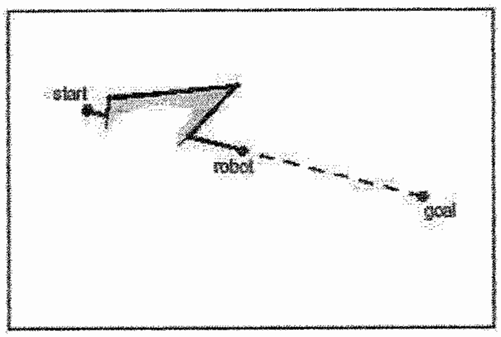
Figure 3.5: The Sojourner planetary Mars rover (NASA)



(a) Roadmap algorithm



(b) Cell-decomposition method



(c) Bug algorithm

Figure 3.6: Sensor-based path planners

WMR Research Projects

4

In this chapter some examples will be presented of robots and research projects dealing with wheeled mobile robotic systems and obstacle avoidance in particular. As one can imagine this overview is far from complete, it is meant to illustrate different ways of dealing with the obstacle avoidance problem.

4.1 Automated Highway System

The automated highway system AHS belongs in the group of practical situations. In this group of projects several companies are trying to bring a high level of automation in automobiles on highways. The current target is set on a train-like motion of several different cars, using the highway at a very short range but constantly aware of possible dangers, e.g. if one car stops for some reason or some other unpredicted incident happens, the individual car should react automatically avoiding any collisions, not compromising the safety of the driver/passenger.

Technically this is analog to a lane following problem including an obstacle avoidance function for a non-holonomous wheeled mobile robotic system. An extra difficulty in this context are the safety concerns that are inevitably in traffic situations. Highways present an unknown and dynamic environment with real-time constraints. In addition, the high speeds of travel force a system to detect objects at long ranges. Although there are a number of methods that can successfully detect moving vehicles, the more difficult problem of finding small, static road debris such as tires or crates remains mainly unsolved.

It is very difficult to present more details on obstacle avoidance algorithms and sensor configurations for these projects, because a lot of different companies (Ford Motors, Toyota, et al) are tackling this problem, keeping their individual results as secret as possible. In America scientific research projects are mainly sponsored by the National Automated Highway System Consortium, like Hancock [17] or Horowitz and Varaiya [18]. A special note goes to Ng and Ahmed [1] with their very practical and advanced Smart Car project.

4.2 RHINO

In the Deutsches National Museum in Bonn, visitors can take a guided tour accompanied by a mobile robot. This robot is not only capable of interacting with its guests and giving an audio-visual presentation it has also the preferable property of not colliding into artworks

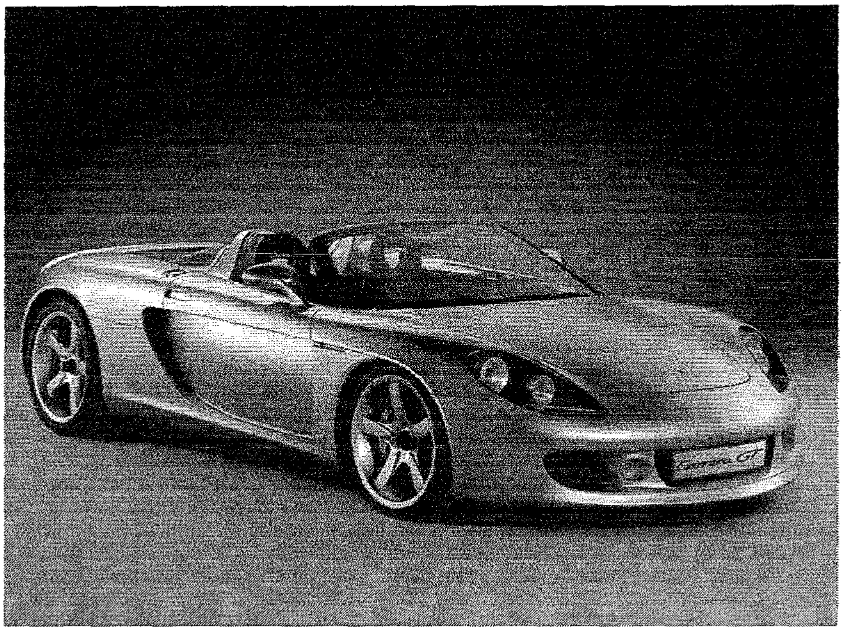


Figure 4.1: Smartcar, an implementation for the Automated Highway System

Figure 4.2: Rhino - An interactive tourguide

and people in the museum.

This wheeled mobile robot called RHINO is designed at the University of Bonn in the group of Burgard [7] making use of probabilistic roadmaps (PRM) and Artificial Intelligence techniques. Its sensor configuration is pretty extended but the results are at least promising.

4.3 SuperMARIO

All previous described projects are almost evolved out of the area of science into more or less commercial available applications. *SuperMARIO* is a research robot meant for testing and improving basic control tasks for wheeled mobile robots (WMR).

The project is based at the University of Rome, Italy in the department of Informatics and Systems and has delivered several insights in the basics of mobile robot control, like feedback and feedforward control. De Luca [30] gives an experimental overview of the project.

4.4 Artificial Neural Networks

Artificial neural network modelling has become quite common in the area of obstacle avoidance techniques. There are many research projects all over the world, like the *ALVINN* and *ROBIN* robots at the Carnegie Mellon University [12]. Lagoudakis [24] uses a Hopfield Neural Network for dynamic path planning and obstacle avoidance, or Neural maps for mobile robot navigation [25] both applied on the *NOMAD* robot. The Boston University's Neurobot Lab uses a neural network for adaptive obstacle avoidance to apply on their *Khepera* robot as described by Chang [9], [10], [11] and Gaudiano [16].

One should note that neural networks are mainly a way to model a problem in a different way. In most of the cases it is possible to describe, or better reformulate the neural network modelling into another existing technique. A useful example is described in Liu and Khatib [29]

4.5 Roadmap methods

As stated in previous section Virtual Potential Field methods are widely applied. Probabilistic roadmap methods are widely developed but not that much applied on 'real' robots. This is mainly because of the relatively large complexity of these methods. There are a few projects worth mentioning though. J.-P. Laumond gives a very extensive overview in his book *Robot Motion Planning and Control* [28]. Probabilistic Roadmap techniques are implemented mainly on simulation level by Kavraki, Svestka, Overmars and Latombe [20], [21].

4.6 Virtual Potential Field Robots - CARMEL

The class of potential field techniques is being applied on a lot of different robots. Borenstein and Koren have developed and applied this method on their robot *CARMEL* (figure 4.3).

CARMEL was not only used to test VPF-like methods. Borenstein also applied a *histogramic in-motion mapping (HIMM)* [4], a *vector field histogram (VFH)* method [5] and a *model-reference adaptive motion controller (MRAC)* [15].

4.7 Other Virtual Potential Field implementations

After the encouraging results on both scientific simulations and implementations, the class of virtual potential field methods have been adapted widely, not only in wheeled mobile robotic systems. A few important implementations lay in the field of aids for the handicapped.

The *nursing robot* (figure 4.4(a)) is a robot which can help in simple tasks. This kind of robot is also available in office-like environments as a substitute for the internal mailman. The Blind guide robot (figure 4.4(b)) is the robotic equivalent for a guiding dog for blind people. The *NavChair* (Figure 4.4(c)) is specific designed for visual handicapped people. The advanced wheelchair implementation is pretty related to the BellyBot project, where the user indicates a desired position where the path planner should warn or change the trajectory to avoid any obstacles on its way where necessary.

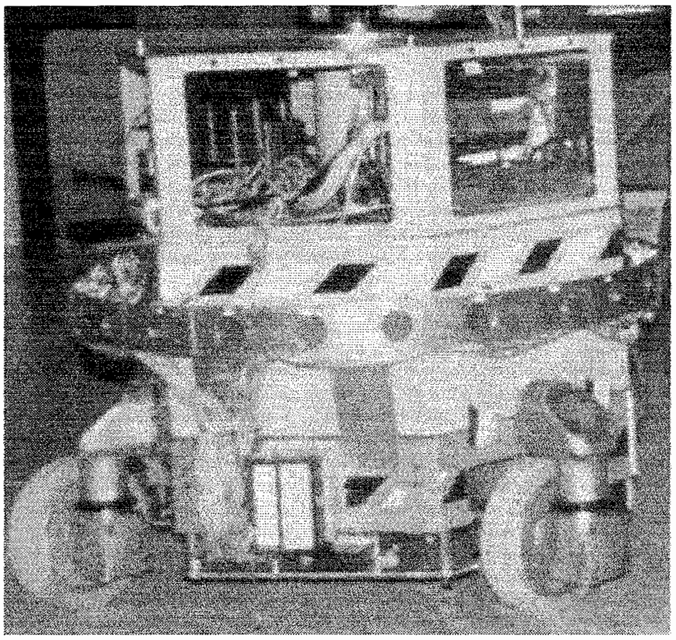
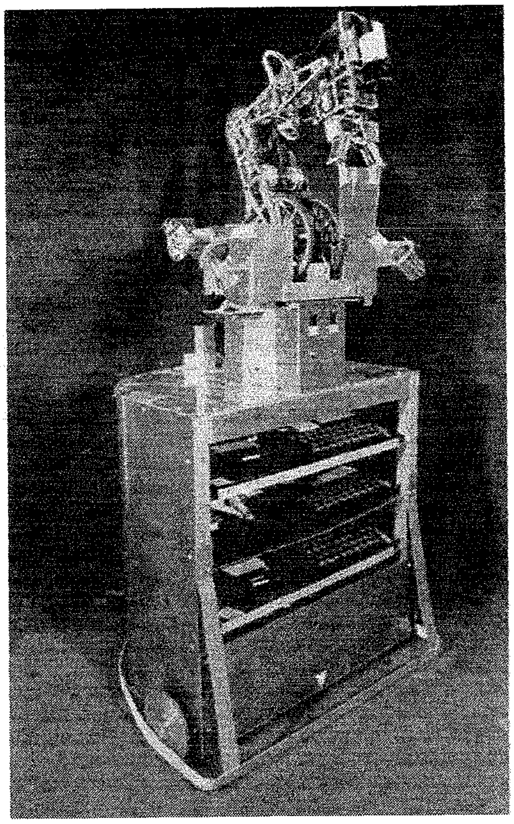
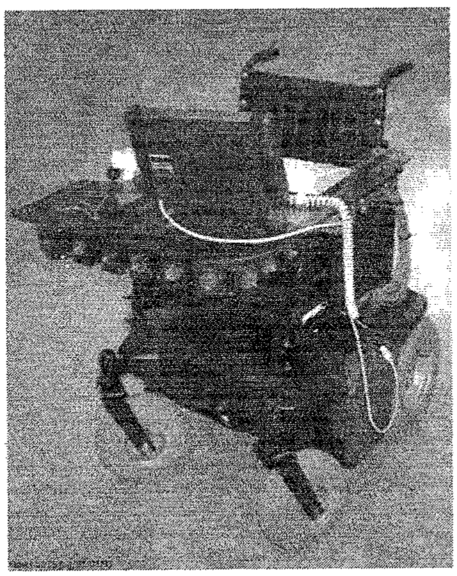


Figure 4.3: CARMEL Wheeled Mobile Robot



(a) Nursing robot

(b) Blind guide robot



(c) NavChair

Figure 4.4: VPF based aids for the handicapped

5 Conclusions

Planning robotic motions is a very active field of research. In this report a literature exploration has been conducted to investigate the possibilities for robotic motion planning, to learn from previous implemented projects and to search for interesting gaps in mobile robotic motion planning which are still to be investigated.

Literature shows that robotic motion planning in general can be divided into implicit and explicit motion planning. Explicit schemes calculate the complete trajectory before the motion is executed, implicit trajectories are calculated during a traversal, using the state of the robot and the environment.

The field of wheeled mobile robotic systems have introduced some specific strategies for obstacle avoidance. These strategies can be separated into three categories; classical motion planners, heuristic planners and "complete and correct" sensor-based path planning.

Classical motion planners posses the very useful properties of correctness and completeness, but need a priori knowledge of the complete environment to deliver a motion plan. The three most important classical planners are roadmap, cell decomposition and virtual potential field algorithms. Roadmap algorithms use different mappings to determine an obstacle free road between the current and the desired position. Cell decomposition algorithms divide the space into two parts, free space and obstacle containing space. Cell decomposition algorithms do not produce hard-to-follow one-dimensional curves but give save corridors between obstacles, this puts less pressure on tracking controllers but imply that a second algorithm has to be constructed which selects one path out of the infinite possibilities. The last important subclass constructs a virtual potential field, where the robot gets a virtual positive potential, the goal an attractive negative potential and every single obstacle a repulsive positive potential. Now letting physical laws do their work, some path will follow. Big setback to this method is the risk of local minima. Solutions to this problem have been proposed, but tuning requires extensive knowledge of the environment, losing generality of the method. Classical methods can all be categorized into the class of explicit motion planners.

Heuristic planners on the other hand require no a priori knowledge of the environment, using only sensor inputs and robot states. All heuristic planners select an action out of a preprogrammed list of commands based on local criteria. Not requiring complete a priori knowledge of the environment is the main value of this class, but in practise these methods resolve mainly in very long paths, because no optimalisation of what so ever can be conducted. This class is completely implicit.

The last class has been developed combining the positive properties of both previous described classes, gaining "complete and correct" sensor-based path planning. Most classical methods have now been redesigned into a sensor-based variety. This is the class where most research activity is positioned nowadays. Because the trajectory is not completely calculated before motion execution, this is an implicit class of motion planners.

Looking at implemented projects four areas can be distinguished. First of all the research testbeds on which different methods are tested on lab scale. The second area is the automotive industry, especially the American Automated Highway System project. In this project cars are designed which are capable of highspeed driving very close to each other considering avoidance of possible obstacles. A lot of research effort is put into the area of space exploration. Main players in this field are the American NASA and the European ESA space organizations. Especially NASA's Jet Propulsion Lab has put lots of effort in developing wheeled mobile space robots for the exploration of the planet Mars. The last area is much smaller than previous areas but not less interesting. This area contains implementations to help daily life and navigation of handicapped people, especially visual limited people.

Bibliography

[1] A. Ahmed and Brian Ng, *EE 401Project: The Smart Car*, April 2001.

[2] H.C. van den Berg, *A wheeled mobile robot - Creating an experimental environment for non-linear controllers.*, Internal report (DCT 2002-20) Eindhoven University of Technology, March 2002.

[3] J. Borenstein and Y. Koren, *Real-Time Obstacle Avoidance for fast Mobile Robots.*,IEEE Transactions on Systems, Man and Cybernetics vol 19, no. 5, September/October, pp 1179-1187, 1989.

[4] J. Borenstein and Y. Koren, *Histogramic In-Motion Mapping for Mobile Robot Obstacle Avoidance*, IEEE Journal of Robotics and Automation, vol 7, no 4, pp 535-539, 1991.

[5] J. Borenstein and Y. Koren, *The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots*, IEEE Journal of Robotics and Automation vol 7, no 3, pp 278-288, June 1991.

[6] R. Brooks and A. Flynn, *A robust layered control system for a mobile robot*, IEEE Trans. on Robotics and Automation, 2(1), 1986.

[7] Burgard, *hier moet de titel van dat paper komen*, University of Bonn.

[8] J.F. Canny, *The Complexity of Robot Motion Planning.*, MIT Press, Cambridge, 1988.

[9] C. Chang and P. Gaudiano, *A neural network model of avoidance and approach behaviours for mobile robots*, Boston University Neurobotics Lab Dept. of Cognitive and Neural Systems.

[10] C. Chang and P. Gaudiano, *Neural competitive maps for reactive and adaptive navigation.*,Boston University Neurobotics Lab Dept. of Cognitive and Neural Systems, 1997.

[11] C. Chang and P. Gaudiano, *Application of biological learning theories to mobile robot avoidance and approach behaviours*, Boston University Neurobotics Lab Dept. of Cognitive and Neural Systems, 1998.

[12] S. Chen, *Learning-Based Vision and Its Application to Autonomous Indoor Navigation*, Carnegie Mellon Univeristy, 1998.

[13] H. Choset, *Sensor-Based Motion Planning: The Hierarchical Generalized Voronoi Graph.*, PhD Thesis, California Institute of Technology, 1996.

[14] C. Ó'Dunlaing and C.K. Yap, *A retraction method for planning the motion of a disc.* Journal of Algorithms, 6, 1982.

[15] L. Feng, Y. Koren and J. Borenstein, *A Model-Reference Adaptive Motion Controller for a Differential-Drive Mobile Robot.*, University of Michigan.

[16] P. Gaudiano and C. Chang, *Adaptive obstacle avoidance with a neural network for operant conditioning: experience with real robots*, Boston University Neurobotics Lab Dept. of Cognitive and Neural Systems, 1997.

[17] J.A. Hancock, *High-Speed Obstacle Detection for Automated Highway Applications*, PhD Thesis, Carnegie Mellon University, May 1997.

[18] R. Horowitz and P. Varaiya, *Control Design of an Automated Highway System*, Berkley University of California, February 2000.

[19] I. Kamon, E. Rivlin and E. Rimon, *A new range-sensor based globally convergent navigation algorithm for mobile robots*, Proceedings of the IEEE Conference on Robotics and Automation, p. 429-435, April 1996.

[20] L.E. Kavraki and J.-C. Latombe, *Probablistic Roadmaps for Robot Path Planning*, Rice University Houston/Stanford University.

[21] L.E. Kavraki, P. Svestka, J.-C. Latombe and M.H. Overmars, *Probabilistic roadmaps for path planning in high-dimensional configuaration spaces.*

[22] A.J. Kelly. *RANGER - An intelligent predictive controller for unmanned ground vehicles.*, The Robotics Institute, Carnegie Mellon University, 1994.

[23] O. Khatib, *Commande Dynamique dans l'Espace Oprationnel des Robots Manipulateurs en Prsence d'Obstacles*, PhD Thesis, Ecole Nationale Suprieure de l'Aronautique et de 'Espace, Toulouse, 1980.

[24] M.G. Lagoudakis, *Hopfield Neural Network for Dynamic Path Planning and Obstacle Avoidance.*, The Center for Advanced Computer Studies University of Southwestern Louisiana.

[25] M.G. Lagoudakis and A.S. Maida, *Neural Maps for Mobile Robot Navigation*,The Center for Advanced Computer Studies University of Southwestern Louisiana, 1999.

[26] J.-C. Latombe, *Robot Motion Planning* Kluwer Academic Publishers, Boston, 1991.

[27] S.L. Laubach, *Theory and Experiments in Autonomous Sensor-based Motion Planning with Applications for Flight Planetary Microrovers* PhD Thesis, California Institute of Technology, 1999.

[28] J.-P. Laumond *Robot Motion Planning and Control*, Lecture Notes in Control and Information Sciences 229, Springer, ISBN 3-540-76219-1, 1998.

[29] J. Liu and O. Khatib, *Practical connection between Potential Fields and Neural Networks*, Hong Kong Babtist University/Stanford University USA.

[30] A. De Luca, G. Oriolo and M. Venditteli, *Control of Wheeled Mobile Robots: An Experimental Overview*, Dipartimento di Informatica e Sistemistica, Universit degli Studi di Roma "La Sapienza", Italy [English].

[31] V.J. Lumelsky and A.A. Stepanov, *Path-planning strategies for a point mobile automaton moving amidst obstacles of arbitrary shape.*, Algorithmica, 2:403-430, 1987.

[32] N.J. Nilsson, *A mobile automation: An application of artificial intelligence techniques.* Proceedings of the first International Joint Conference on Artificial Intelligence, Washington DC, 1969.

[33] N. Rao, S. Kareti, W. Shi and S. Iyengar, *Robot navigation in unknown terrains: Introductory survey of non-heuristic algorithms.*, Oak Ridge National Laboratory, July 1993.

[34] U. Raschke and J. Borenstein, *Real-time Obstacle Avoidance for Non-Point Mobile Robots*, University of Michigan, Fourth World Conference on Robotics Research.

[35] E. Rimon and J. Canny, *Construction of c-space roadmaps from local sensory data: What should the sensors look for?* Proceedings of the IEEE International conference on Robotics and Automation (IRCA'94), 1994.

[36] E. Rimon and D.E. Koditschek, *Exact robot navigation using artificial potential functions.* IEEE Transactions on Robotics and Automation, 8(5):501-518, 1992.

[37] A. Stentz, *Optimal and efficient path planning for partially known environments*, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'94), May 1994.

[38] A. Stentz, *The focussed D* algorithm for real-time replanning.*, Proceedings of the International Joint Conference on Artificial Intelligence, August 1995.

[39] M. Zefran, *Continuous Methods for Motion Planning.* IRCS report 96-34, 1996.