ELSEVIER

Computers and Electronics in Agriculture
25 (2000) 121–132

Computers
and electronics
in agriculture

www.elsevier.com/locate/compag

Symbolic course description for semiautonomous agricultural vehicles

F. Freyberger [a,*], G. Jahns [b]

[a] *Institute of Automatic Control Engineering, Technische Universität, Arcisstrasse 21, D-80333 München, Germany*
[b] *Instituite of Biosystems Engineering, Federal Agricultural Research Centre, Braunschweig, Germany*

Abstract

Agricultural field machinery operators have to control the motion of the tractor and the various implements. Autonomous motion planning and execution capabilities of the tractor could allow the operator to concentrate on the operation of the implements. Commanding and programming should be easy and the semiautonomous operation must be robust. In the field of service robotics several approaches for motion planning and control have been developed and successfully tested. A combined symbolic and numeric/geometric control is presented and advantages for the motion planning and control of agricultural field machinery will be pointed out. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Path planning; Map representation; Course description; Symbolic navigation; Semiautonomous vehicles

1. Introduction

Two basic schemes for mobile vehicle navigation can be distinguished; the Cartesian map-based and the relative sensor-based methods. A combination of both approaches is typically used in practice for mobile robots.

Purely map-based navigation schemes require a detailed 2-D or 3-D model of the environment, as well as a vehicle with real-time localisation capability. The GIS (see Geographic Information Systems, this issue) could serve as a basis for a model containing all data of the field including borders and permanent obstacles. Real-time localisation could be based mainly on advanced GNNS (see T. Bell about

* Corresponding author.

PII: S0168-1699(99)00059-9

carrier-phase differential GPS, this issue). Major discrepancies between the real environment and the model cannot be tolerated.

On the other hand, for purely sensor-based navigation a structural description of the working area may suffice. An intelligent onboard sensory system (see Hague et al. and Debain et al. about machine vision, this issue) is required for real-time recording and converting 2-D or 3-D structural information of the environment into adequate guidance commands.

A sensor-guided vehicle may require switching between different navigation modes of operation as well as introduction of open-loop controlled path sections. Reaching a location defined relative to a landmark often triggers mode switching.

To complete the control concept of semiautonomous robot vehicles such as field machinery an other control aspect has to be pointed out without being further reflected in this paper. The above introduced navigation modes of operation include only poor reactive capabilities to cope with unexpected obstacles. A more intelligent behaviour can be achieved by adding an exception handling mechanism. It will be triggered by state transitions caused either by interpreted sensor information, command inputs, etc. or further internal signals, e.g. power down etc. It has to plan operation sequences based on a set of rules to handle the exception and to control the execution. Based on this basic idea by Glüer and Schmidt (1996b) an approach, evaluated by experiments is described. It combines within a high level co-ordination layer a task decomposition algorithm producing operation sequences, a rule based exception planner and a controller for bumpless switching between normal execution state and exception handling state. Normalised symbolic relations implemented in a relational database (Glüer and Schmidt, 1996a) represent the rules. Main benefits of this approach are a transparent and consistent representation of the rule set, easy capabilities to extend the relations with respect to functionality, application, etc., and a simple on-line modification of the preplanned operation sequence for exception handling purposes.

In the context of sensor based navigation it is of interest to consider how humans specify and perform simple motion tasks. Typically basic symbolic information is preferred over detailed geometrical path specifications, i.e. the desired path is specified through an ordered set of landmarks together with required actions. Typical landmarks during fieldwork are previous paths or rows of crops. An advantage of landmark-oriented visual navigation is its flexibility and robustness with respect to uncertainties and minor changes in the environment. Implementation of this navigation strategy on mobile field machinery requires appropriate sensory assistance for landmark detection.

In the approach reported in this paper no explicit numerical parameters are required to specify the course of the field machinery.

Other approaches often need symbolic commands with additional numerical arguments, such as the length of a move in the field or the left or right turning angle for headland turning. These parameters may simplify the task, but for field machinery with semiautonomous behaviour they should not necessarily be required.

2. Symbolic course description

A human operator is still the best model of a system to guide a vehicle. For this task he/she does not use numeric/metric but symbolic information. Such non-metric course description is oriented at field marks and structures such as cutting edges, field boundaries, etc. For example, in driving parallel to a cutting edge, distance and direction according to the cutting edge are much more relevant values than the distance travelled. The last factor only comes into play when the headland is approached.

By means of a structured symbolic course description, the complete task of guiding a vehicle autonomously may be split up into simple individual actions to which certain unique operations can be assigned. Examples include approaching a cutting edge after switching from manual to automatic, driving along a cutting edge, doing a certain type of headland turning and so on. Each operation is executed as long as the decision by the master control to switch to the next operation becomes necessary.

Gilg and Schmidt (1994, 1995) and Gilg (1997) proposed an interesting procedure to structure the task for a comparable scenario of the indoor guidance of autonomous mobile robots. Emphasis was laid only on the locomotion part as the most important part of autonomous vehicle guidance. The following part will briefly present this approach while a third part show similarities to agricultural field applications.

Fetch and carry tasks in an environment structured by corridors, crossings, turns to the left or the right side, office and elevator doors are most typical for indoor applications of mobile service robots. Taking into account this reduced set of environmental elements and based on a sensor system capable of robust detection of corresponding landmarks, a powerful and very easy symbolic task description and control was developed. The hierarchically structured control scheme includes a symbolic task planning and controller layer, a sequence controller and a Cartesian motion controller as shown in Fig. 1.

2.1. Symbolic map

A priori knowledge about the environment is stored in two consistent graph type maps, a coarse one and a more detailed fine one (see Fig. 2). These symbolic maps describe the graph of subsequent so-called situations (such as door entrances, door passing or passages, turns left, and turns right) and thus represents a graph of all-moveable paths and possible goals. Situations can be seen as environmental arrangements where path alterations are possible.

2.2. Task planning algorithm

A typical symbolic locomotion task commanded by the human operator consists of two elements: the $\langle$type$\rangle$ and the $\langle$goal$\rangle$. The examples in Table 1 illustrate two command forms.

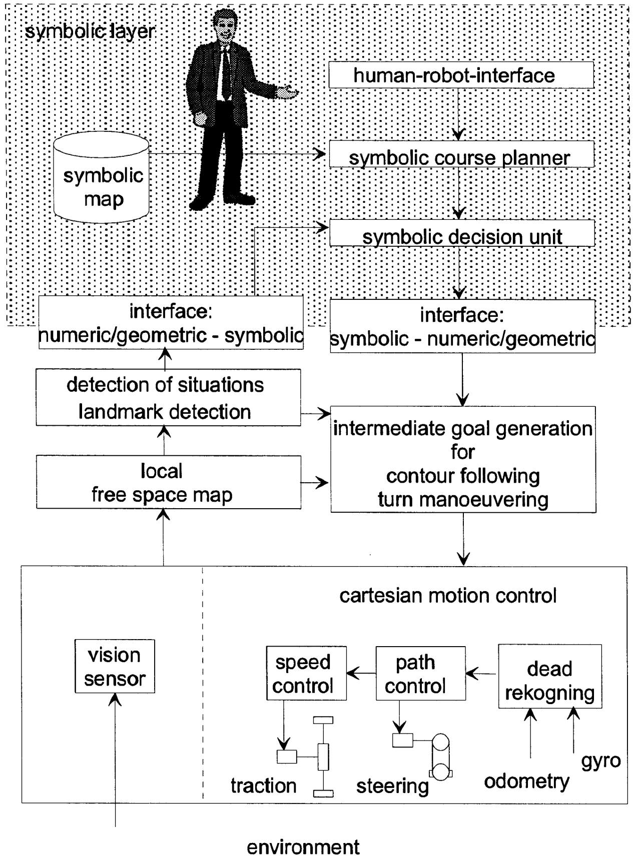

Fig. 1. Control scheme of a semiautonomous mobile robot including symbolic and numeric control loops.

The locomotion specific part of these simple human-like notations will be transformed by a course planning process based on a coarse map. This planning process will generate a sequence of symbolic motion commands.

The planning operation is realised as a graph search process and was implemented based on dynamic programming with a cost function that takes into account the distances and turning manoeuvres. The resulting motion commands

Table 1
Example task commands

<type>	<goal>
FETCH FROM	POST OFFICE
CARRY TO	OFFICE NO. XY

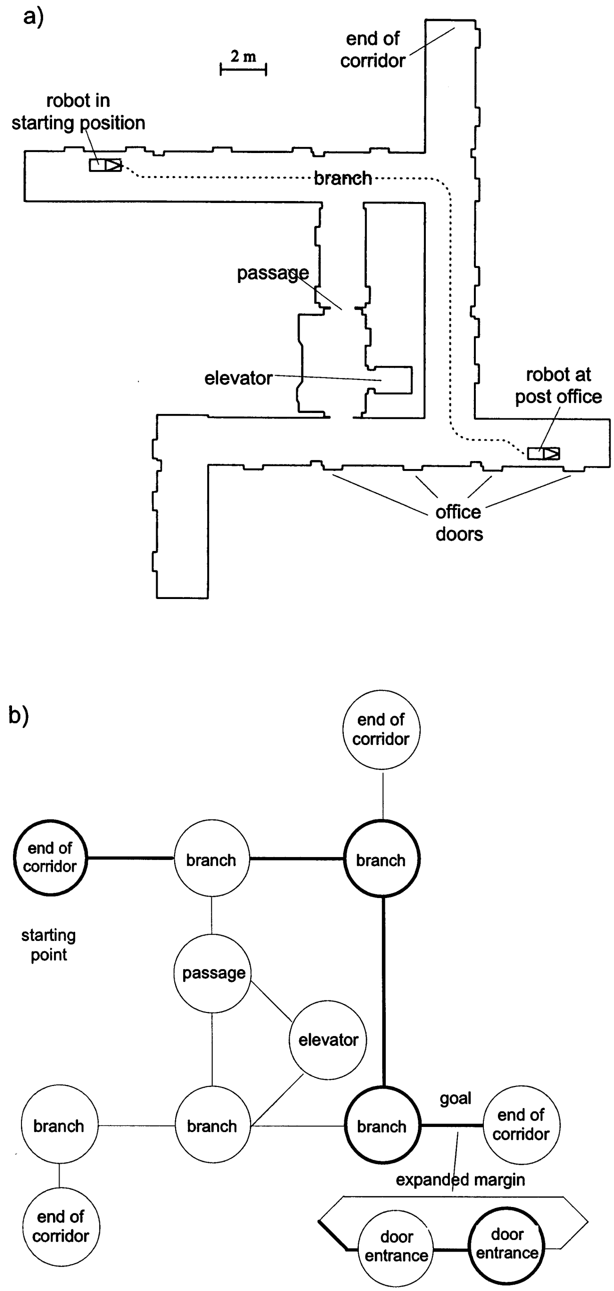

Fig. 2. Example for a geometric floor plan (a) and the corresponding symbolic path graph (b).

consist of only four symbolic arguments: ⟨operation⟩, ⟨situation⟩, ⟨selector⟩, ⟨attribute⟩. These commands and the corresponding syntax rules are similar to a programming language.

In case of the above-mentioned CARRY task command the motion planner will produce automatically the following symbolic motion commands (compare bold-faced lines in Fig. 2(b)) based on the fine map.

The correspondences between the symbolic task oriented goals, e.g. POST OFFICE and the geometric oriented situations for the locomotion, e.g. 2ND DOOR, are stored in a simple list. Since the course plan consists of a sequence of incremental motion macro elements, the starting posture of the robot must be known.

To operate this course plan a decision algorithm is necessary that handles the transitions between the commands and completes the course.

2.3. Decision algorithm

The decision algorithm is an essential part of the symbolic robot motion controller. It switches the motion operations depending on the command sequence and is triggered by the occurrence of a change of the situation. The input information is based on actual sensory data. It monitors whether the robot moves along a corridor, a borderline, etc., or reaches the next situation characterised by a set of landmarks. Output information is a symbolic motion macro command like: move along the left side of the corridor or manoeuvre a turn left. The decisions are made by symbolic rules constructed as follows.

If ⟨landmark set for situation detected⟩ and ⟨course command⟩ and ⟨selector⟩ then ⟨motion macro⟩.

In case of the motion command ‘move turn left’ the following motion macros may be generated. They depend on the result of the situation detection based on the corridor’s contour that is detected by an onboard sensor system, e.g. a video camera:

if situation not detected and move turn
  then follow corridor until end of the contour
if contour detected
  then follow contour until end of the contour
if situation detected and move turn
  then manoeuvre and turn to the left side

The described task and motion planning and the rule-based decision are similar to human actions. As an important advantage they allow tasks and courses by simple textual commands to be formulated without numeric/metric parameters. Also, no detailed knowledge about the kinematics of the robot or the measurement range of the sensor system is necessary within these symbolic planning and control layers.

Nevertheless it is necessary to interface the symbolic part with the numeric/metric working sensor and the motion controller systems of the robot. Appropriate interfaces are necessary to couple both parts.

2.4. Sensor interface

The detection of landmarks in this case is based on vision data and calculation of the borderlines between wall and floor (Gilg and Schmidt, 1994). The interpretation of the calculated line segments allows extracting the following basic landmarks:

- wall or contour left side,
- wall or contour right side,
- wall or contour end left side,
- wall or contour end right side,
- wall edge left side,
- wall edge right side,
- wall edge left and right side.

The detection of situations depends on the occurrence of combinations of basic landmarks; e.g. if contour end left side and wall edge left side than a turn left situation is detected.

2.5. Motion controller interface

The motion controller interface is more complex. It executes two main control operations according to the motion commands:

- contour following motion control,
- local manoeuvring control.

The contour following motion control is used for driving in corridors. Sensor input data are the geometric values of extracted line segments relative to the robot vehicle representing the contour and selector values, e.g. middle path, left or right side. Based on this information intermediate Cartesian goal points will be calculated and executed by the Cartesian motion controller (see Fig. 1).

Local manoeuvring control (Bott et al., 1993, 1995) is active when commands like turning, door passing, parking, etc., are to be operated. It takes into account the free manoeuvre space detected by the vision sensor, the car-like kinematic model of vehicle and plans an appropriate more or less complex path. This planned reference path normally consists of a sequence of line and circle segments and can include forward and backward motion. It is represented by a sequence of Cartesian move commands that will be operated accurately by the Cartesian motion controller (Horn et al., 1992).

Exception handling in some important cases, e.g. obstacles on the path, is done during contour following control reflexively by sensor based path correction or by a local manoeuvre. In cases of blockades symbolic replanning is activated (Glüer and Schmidt, 1996b).

The proposed overall control system is implemented in the experimental mobile platform MACROBE (see Fig. 3) and successfully evaluated by several long-range experiments. Gilg (1997) and Azarm et al. (1994) document them in detail.

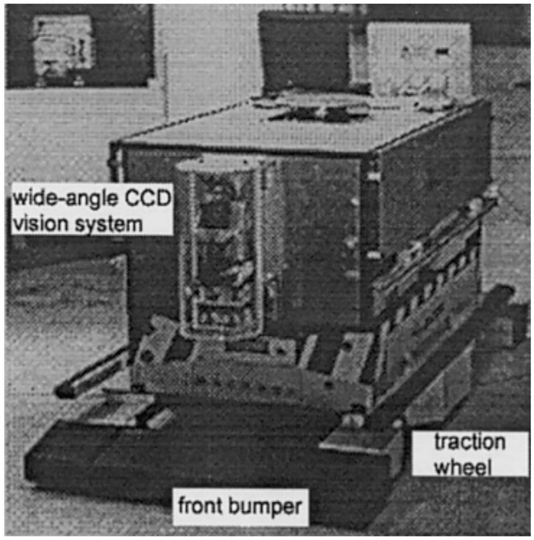

Fig. 3. The experimental mobile platform MACROBE.

2.6. Human-robot-interface

The command and programming interface for service robots must be human centred. For this purpose human modalities like natural speech or graphical symbols play an important role when constructing such interfaces (Fischer et al., 1996). The symbolic tasks and course description used in the highest control level meets these demands excellently. The examples in Table 2 are very close to a spoken command language. Also there are no numeric/geometric values necessary for this description. A simple programming interface based on only graphical symbols is possible.

3. Use in field applications

The described approach also seems appropriate for guiding vehicles autonomously in agricultural fields.

Table 2
Example motion command list

<operation>	<situation>	<selector>	<attribute>
MOVE	2ND TURN	RIGHT	MIDDLE PATH
MOVE	TURN	LEFT	
PARK	2ND DOOR	RIGHT	

Important similarities are:

- Motion operations for most field machines are limited to line following and more or less preplannable turn manoeuvres.
- Simple situations that cause alteration of the path are the demarcation lines and the field edges.
- In most cases the locomotion of field machinery is based on a car-like kinematic thus local manoeuvring can be easily adapted.

The main difference that has to be taken into account can be seen in the motion task itself. In the robotics case it is a point to point motion in general and in the field application an area covering motion. This means that the symbolic motion command sequence is not completely plannable but only in a recursive way. This seems to be no restriction because a symbolic termination of the recursive motion operation can easily formulated and detected by landmarks such as corners of the top and bottom headland or the border edge of the field.

The advantage of this approach is that the decision unit is able to execute the motion command sequence incrementally step by step without knowing the complete driving order. Switching to the next operation as a result of a decision process which may be triggered by events. Such events may be features, like landmarks such as headlands, or the interference of the operator, e.g. commanding the change from autonomous to manual operation. Events may also be calculated by using GPS position data and the driving order. Each operation will be executed until a limiting event occurs, switching must be bumpless and contradiction free.

Contrary to the symbolic course description for mobile robots, those for agricultural machinery may apply additional attributes, containing supplementary information about the operation to be executed next. Such an attribute may be a type of tensor, similar to those known from spline operation, which determines up to which degree the present course will smooth the detected demarcation line e.g. a cutting edge to be followed. The result may be a straight line or a more or less curvy contour as necessary for contour cultivation to prevent erosion.

Table 3 shows examples of symbolic course descriptions. The first and third line may stand for an example of drilling corn where it is desirable to equalise irregularities of the preceding track. Line two and four could serve as an example of drilling sugar beets. In this case it is also desirable to avoid that irregularities of the preceding track are copied, but it is more important to keep the distance deviation between the old and new track below a certain value to make sure that multi-row harvester can be applied. So small irregularities may be acceptable.

Table 3
Examples of symbolic course description

Operation	Situation	Selector	Attributes
Driving	< >.headland turn	Back right	Line, minimise direction error
Driving	< >.headland turn	Back right	Line, minimise distance error
Driving	< >.headland turn	Back left	Contour, minimise direction error
Driving	< >.headland turn	Back left	Contour, minimise distance error

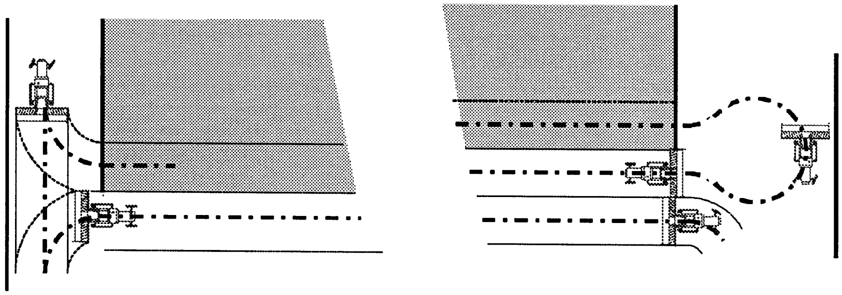

Fig. 4. Tracks of a combine — right, Ω-turn; left, T-turn.

In addition there are adaptations to be made in the numeric/geometric part of the control system. Because of the complex little structured environment a variety of sensors, e.g. vision, laser and GNNS, will be necessary to robustly detect and accurately measure natural landmarks, headland geometry and leading lines such as demarcation line, cutting line or crop rows. A high quality of the sensor input is essential for a proper performance of line or contour following control. Compared to the robotics case planning and execution of the turn manoeuvres might be more easy. Mainly two turn manoeuvres are most likely, a Ω-like and a T-like manoeuvre. They must be automatically selected and parameterised depending on the geometric properties of the top and/or the bottom headlands and by the type of the implement or the implement coupling.

Fig. 4 as an example shows tracks of a combine. The corresponding symbolic course description is given in Table 4. Arriving at the field the operator may drive as usual to a starting point close to the track to be followed and than switch from manual to automatic; bottom right in Fig. 4 and first line in Table 4. The system than will automatically approach the demarcation line by line following control and follow it. While following the demarcation line the optimisation criteria given by the attributes are to avoid overlap and missing as well as smoothing the preceding track. Reaching the narrow headland on the bottom side of the field (left in Fig. 4) the manoeuvre controller plans in this case automatically a T-turn using informa-

Table 4
Example of a symbolic course description for combine according Fig. 4

Operation	Situation	Selector	Attributes
Start up	Mode switch	Automatic	
Begin			
.			
Driving	Top headland turn	Back right	Line, minimise direction error
Driving	Bottom headland turn	Back left	Line, minimise direction error
Driving	Top headland turn	Back right	Line, minimise direction error
.			
Start up	Mode switch	Manual	

Table 5
Drive, steering and implement coupling of agricultural tractor-implement combinations

Drive	Steering	Implement
Front-wheel	Front-wheel	Pulled
Rear-wheel	Rear-wheel	Front-mounted
Four-wheel	Articulated	Rear-mounted
All-wheel	All-wheel	Side-mounted
Chain	Tank	Sub-mounted central-mounted

tion that will be delivered from the CPDGPS signal as well as from machine vision. After performing the T-turn the harvester follows the demarcation line, now on the left side of the combine, up to the headland at the right. This may be a wider headland to allow Ω-turns and so on until the edge of the field is reached.

For the variety of agricultural vehicles and tractor-implement combinations the syntax for autonomous indoor robots will need to be extended. Contrary to mobile robot passenger cars and others, agricultural vehicles do not address a certain type of vehicle.

Agricultural vehicles are versatile, mostly operation depending combinations of different chassis, engines and implements. So 'agricultural vehicle' is a collective term, which refers to numerous types of self-propelled machines like combines or beet lifters, and an even bigger number of tractor-implement combinations, the difficulty of meeting these demanding requirements become evident, Table 5.

4. Conclusion

Recent mobile service robots for indoor using a combination of symbolic and numeric/geometric control layers have been developed and successfully evaluated. The symbolic layer makes it possible to command the robot task and plan the necessary locomotion without any geometric values. It is therefore widely independent from the machinery and generally applicable. It supports the development of human centred man machine interfaces that can be handled not only by computer specialists. To keep the symbolic command language simple, the numeric/geometric control layer must be powerful and execute semiautonomous subfunctions including low level exception handling.

The introduced control architecture and method can be advantageously used to develop agricultural tractor control with semiautonomous locomotion functions. Some examples demonstrate this and also point out necessary modifications.

At the moment it seems that there is not a methodological gap but some lag in improved less expensive sensor systems for agricultural outdoor use. Besides this, it is necessary to bring the operators into contact with this new field machinery at an early state of development. This helps to discuss together with them problems of acceptance, ergonomics of the human–robot-interface and additional support functions they have in mind. Last not least autonomous field machinery but can be seen as a starting platform for teleoperated agricultural production of the future.

References

Azarm, K., Bott, W., Freyberger, F., Glüer, D., Horn, J., Schmidt, G., 1994. Autonomiebausteine eines mobilen Roboterfahrzeugs für Innenraumumgebung. It&Ti 1 (36), 7–15.

Bott, W., Freyberger, F., Schmidt, G., 1993. Automatische Planung und Ausführung lokaler Fahrmanöver für Roboterfahrzeuge. In: Tagung Intelligente Steuerung und Regelung von Robotern. Langen, Nov. 9–10. VDI Bericht Nr. 1094, Düsseldorf, VDI Verlag, 763–772.

Bott, W., Freyberger, F., Schmidt, G., 1995. Automatic local manoeuvre planning and execution for car-like mobile robots. In: Proc. of IFAC Workshop on Motion Control, Munich Oct. 9–11, 387–394.

Fischer, C., Buss, M., Schmidt, G., 1996. Human–Robot-Interface for intelligent service robot assistance. In: Proc. of 5th IEEE Int. Workshop on Robot and Human Communication (ROMAN'96), Tsukuba, November 11–14, 177–182.

Gilg, A., Schmidt, G., 1994. Landmark-oriented visual navigation of a mobile robot. IEEE Trans. Ind. Electron. 41 (4), 392–397.

Gilg, A., Schmidt, G., 1995. Führung eines Roboterfahrzeuges, gestützt auf symbolische Kursbeschreibung und schnelle Landmarkendetektion. Automatisie-rangstechnik 3 (43), 416–423.

Gilg, A., 1997. Weitwinkel CCD-Lichtschnitt-Sensorsystem zur Führung eines mobilen Roboters mittels symbolischer Wegspezifikation, Sinzheim, Pro Universitate Verlag.

Glüer, D., Schmidt, G., 1996a. A relation based approach to flexible exception handling for autonomous mobile robots. In: Bonivento C. et al. (Eds.), Advances in Robotics, The ERNET Perspective, World Scientific, Singapore, 279–286.

Glüer, D., Schmidt, G., 1996b. A new flexible exception handling approach for autonomous mobile service robots. In: Jamahidi M., Pin F., Dachchez P. (Eds.), Robotics and Manufacturing: Recent Trends in Research and Application, Vol. 6, ASME Press, 279–284.

Horn, J., Bott, W., Freyberger, F., Schmidt, G., 1992. Kartesische Bewegungssteuerung des autonomen, mobilen Roboters MACROBE. Robotersysteme 8, 33–43.