XXXI CIOSTA-CIGR V Congress “Increasing Work Efficiency in Agriculture, Horticulture and Forestry“
September 19-21, 2005, University of Hohenheim, Stuttgart, Germany, ISBN 3-00-016346-8

High-Level Control of Robots for Row Crops

JØRGENSEN, RASMUS N.[1]; MICHAEL NØRREMARK[2], CLAUS G. SØRENSEN[1] AND NIELS A. ANDERSEN[3]

[1] Danish Institute of Agricultural Sciences, Department of Agricultural Engineering, Schüttesvej 17, DK-8700 Horsens, Denmark, Rasmus.Joergensen@agrsci.dk
[2] The Royal Veterinary and Agricultural University, Department of Agricultural Sciences, Environment, Resources and Technology, Hoejbakkegaard Allé 2 DK-2630 Taastrup, Denmark, Mino@kvl.dk
[3] Automation, Ørsted•DTU, Technical University of Denmark, Building 326, DK-2800 Lyngby, Denmark, naa@oersted.dtu.dk

Keywords: row crops, semi-structured, semi-natural, hybrid architecture, simulation, test-bed

Abstract

A flexible high-level control language is an important element in the continued task of introducing mobile agents to new application domains. One such new application domain is the use of a small mobile agent in a agricultural row crop field performing intra-row weeding. This paper defines requirements and scope of a process and behaviour based scripting language needed to control the weeding agent in an agricultural row crop. The basis is the transparent and tactical SMR-CL (Small Mobile Robot Control Language). SMR-CL is a real-time control language for mobile robots modified to include necessary motion commands and a supplemental supervisory function to record the progressing coverage of the field. The control language is applied to a case presenting typical field conditions for row crops.

The case stresses the importance of including goal-directing modules like the Supervisory Field Coverage Monitor (SFCM) to coordinate the behaviours. The applicability of the modified SMR language has been successful demonstrated through the vehicle test in an artificial row crop. The analysis of the operational performance of the SMR has revealed its field capacity and rendered probable the scenario of equipping and implementing small weeding agents with a micro-spraying system requiring less than 10% of normal herbicide dosage.

1 Introduction

Autonomous operation of an agricultural inter- and within-row weeding robot has to be carried out in a partially unknown environment (e.g. Tillett, 1998; Hagras et al, 2002). Further, the environment is characterised as “semi-structured” in order to emphasise the notion that some pre-determined structure is present in the field, as opposed to the widely adopted notion of “semi-natural” (e.g. Blackmore, 2002). Enabling a weeding agent to navigate with accuracy within such an environment, motion actions have to be generated as a result of events occurring in the process of performing the weeding. The control structure should be flexible in the sense that the required actions (control primitives) are not predetermined in time and space but rather seen as a function of sensed information from the current environment and the progress of the operation itself. A reactive behaviour must be possible through an adaptation to uncertainty and unexpected events in the environment. Further, the inclusion of a supervisory system for mode generation and goal directing is considered important (Payton, 1986).

The pursued approach to a large extent follows the motto "The world is its own best model" as outlined by Brooks (1991). That is, navigating the weeding agent in an uncertain and unpredictable environment without planning. However, in order to include goal orientation to enable behaviour sequencing a "job completion" module will be included. This module keeps track of the job progress so far and gives advice with respect to guiding the weeding agent in its further coverage of the field. Hence, the proposed control systems resemble that of a hybrid reactive/goal directed architecture often requested by robotic systems (Arkin, 1989; Yavuz & Bradshaw, 2002).

The overall hypothesis is that it is possible for a weeding agent with limited reasoning and planning capabilities to pass over every crop row without any priory knowledge except for the inner and outer field boundaries. The aim is to render the hypothesis by use of simulation using a test-bed approach.

2 Methodology

The applied approach of evaluating the feasibility of adapting an existing agent to a row crop domain involves the following steps: (1) A typical row crop domain comprising a hypothetical sugar beet field is selected and described; (2) The basic system components and functionalities are described; (3) The basis for selecting a potential agent platform together with its control language is detailed; (4) Proposed modifications and extension to selected agent are outlined; (5) A targeted case offering different challenges in terms of motion complexity are designed in order to simulate the use of the modified agent language; (6) In order to test the physical performance of the vehicle incorporating the derived modifications a comprehensive test-bed is designed; (7) An operational performance analysis is carried out based on the results from the testbed.

Characteristics of Agricultural Row Crop Fields

A typical field with cultivation of a row crop includes field boundaries, row structure, row interactions, inner boundaries, stationary obstacles, and headlands. The current robotic weeding approach only requires the field boundary information and permanent obstacle information in order to constrain the agent motion. Information about the distance between crop rows and distance between seeds are well defined and constant throughout the field. This information can also be useful for the agent motions. Supplemental, it is assumed that a prior operation of inter-row weeding has been carried out according to standard agricultural practice.

Basic System Components and Functionalities

The agent will be equipped with a row guidance system capable of sensing three crop rows simultaneously (e.g. the vision system used by Tillett et al (2002) and Tillett et al (1998). The row guidance system will primary use the centre row for guidance and the neighbouring rows as secondary guidance if the centre row is temporally not detectable. The guidance system must also be able to realise headland areas where the rows ends.

The agent must have an obstacle detection system capable of detecting positive and negative obstacles as for example deep wheel tracks. Also, each wheel will have encoders enabling limited dead reckoning used, for example, for headlands turns similar to Tillett et al (1998). Global positioning will be obtained for instance by a DGPS system. For external communication the agent will be equipped with Wireless LAN and GPRS as fallback in case the broadband WLAN communication fails.

The agent needs to know directly or indirectly where to move next when reaching the end of a row (e.g. Hague et al, 2000). According to Andersen & Ravn (2004) this is a soft/non real-time supervisor problem or a strategically controller problem and not a hard real-time problem like the tactical executor level where the SMR-CL language belongs. Hence, a supervisory system is needed keeping track of the field boundaries, the past and the current positions and orientation of the weeding agent.

The Supervisory Field Coverage Monitor (SFCM)

A Supervisory Field Coverage Monitor (SFCM) is central for a weeding agent operating within a row crop field. Neither the weeding agent nor the SFCM has any priory knowledge about the row organisation and location except for the row distance. The SFCM will only know the outer and inner field boundaries. However, since the rows themselves generates a highly organised and structured environment it should be possible for a weeding agent with limited global positioning accuracy, memory, and reasoning capabilities to cover most fields autonomously with a minimum of human support.

3 Selection and Description of a Potential Agent Platform

A small mobile robot (SMR) developed at the Section for Automation, Ørsted, The Danish Technical University (DTU) was found to be an appropriate test platform for this project. Initially the SMR fulfils the criteria for a vehicle small in size, lightweight and easy to handle by one person only. The SMR is a vehicle measuring 0.3 m times 0.3 m in length and width, respectively. The relative small sized and lightweight SMR decrease the risk of conferring any damages to the surrounding environment. However, the size of the vehicle is large enough to enable parts like cameras and other electronic equipment to be mounted on the vehicle.

Table 1: SMR motion primitives provided by the manoeuvre controller

Motion primitives	Parameter	Description
Fwd	*D*	Move forward d meters
turn	*b [‘rad’]*	Turn b degrees in radians around center
turnr	*r b[‘rad’]*	Turn b degrees in radians with turning radius r meter
drive		drive with current speed
Stop		stop vehicle and keep position and orientation
Idle		stop vehicle and keep speed at zero
followline	*‘type’*	Follow line using ‘type’ syntax: e.g. bl (left side of black line), bm (middle of black line), br (right side of black line).
followwall	*‘side’*	Follow a wall to the side of the vehicle using ‘side’ syntax: l (left), r (right)
resetmotors		Reset the motor modules

The SMR real time software hybrid architecture consists of a command arbitration controller, a sensor management controller, and a manoeuvre controller. Motion primitives are used to link and enable the command arbitrary controller to generate actuator control via the manoeuvre controller. Table 1 lists 9 motion primitives.

The Small Mobile Robot (SMR) hardware architecture consists of, among other things, wheels, reflectance sensors and camera for row detection, infrared sensors, and an onboard computer. A detailed description of the hardware is given by Andersen & Ravn (2004) and Børrild (2000).

4 Results and Discussion

Modifications and Extensions to the SMR-CL Language

This section provides the results of utilising the modified SMR for weeding operations within row crop structures. The result of this study comprises SMR-CL language programming sentences consisting of existing motion primitives and addition of new system variables. The development of new system variables was first based on a theoretical case study comprising different types of row structures (Figure 1 left). The results were obtained by a stepwise process, starting with a simple row structure, and then adding more complex row structures.

The case study shown in Figure 1 Left is believed to handle most situations under real field conditions excluding unknown positive and negative obstacles like rocks and erosion channels. The mission was to enable the vehicle to follow all rows and cover the whole area without any external assistance. The row structures include row gaps and headland rows placed in different angels to the main rows. As outlined in the theoretical case study, all situations required information about detection of rows to the left and/or to the right. Hence, the decision made by SFCM in the different situations is based on detected rows in the past, field coverage and vehicle orientation.

Only minor modifications and extensions were necessary to the SMR-CL language used by Andersen & Ravn (2004). The data logging (*log*) command was altered so it is possible to carry out the logging of rows detected to the left and right of the robot. In the following SMR-CL programming the row detection system variable is identified as both *$rowdetectedright* and *rdr*.

Furthermore, this system variable could be set from the SFCM by the command **dr**. Only the camera to the right was mounted, why the *$rowdetectedleft* and *rdl* was simulated by the command **dl** during the test. The row detection system variables are both reset to 0 (~false) for each turn command. The *followline* command was extended with an additional parameter specifying which of the three rows to use as guideline. The specifying variables were *followline “bm”* and *“bmc”* for following black line with the reflectance sensors (center) or the camera, respectively. System variables for stop conditions related to the line sensor were also extended to handle both the reflectance sensor and camera. For example the *$crossingblackliner, $crossingblacklinel, $nolinecamr* and *$nolinecaml* stop conditions were developed to indicate which of the row detecting sensors (right or left camera) that is crossing a black line or do not detect a black line. Furthermore, one additional system variable *$wheretogo* controlled by the SFCM commands was added to the SMR-CL language. The SFCM commands consisted of text code messages send from the host PC via a SSL connection via wireless LAN to the robot.

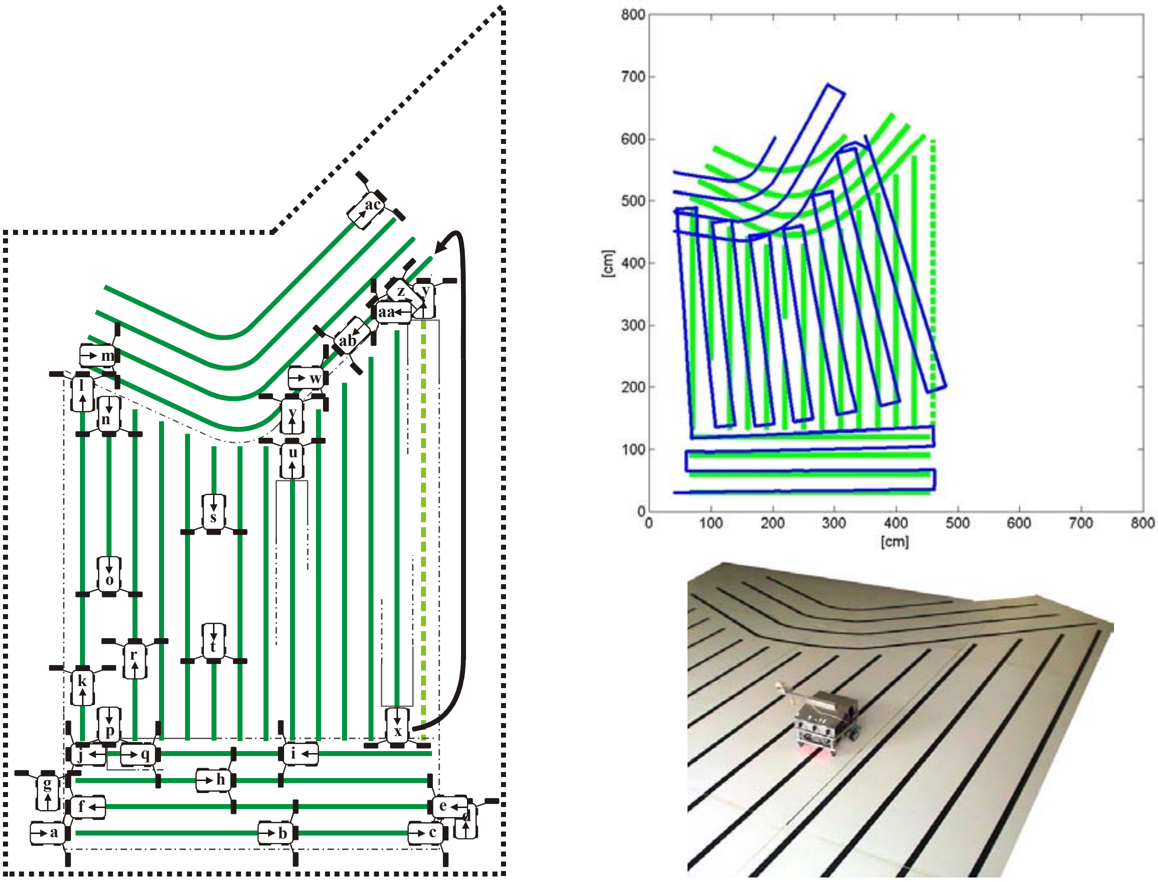

Figure 1**:** **Left -** Illustration of the test case study. The sequence **a** to **x** is referring to different situations. The sequence **a** to **x** is indicating a continues time line, starting at **a**. From situation **x** the chain of events are optional. The dotted row is indicating an optional row. If this row is present, the situation **y** follows situation **x**, but if the dotted row is not present situation **x** is followed by a manual displacement of the vehicle to the uncovered headland area at the top of the figure. The punctuated line (-··-) constrain the area covered so far by the robot during its operation. The field boundary is indicated by the outer-most dotted border line (■ ■ ■).
Right - Test results illustrated as overlay of true map of test bed (Gray) and x- and y-coordinates provided by the vehicle odometry (Black). The dotted line was optional and not used for the test shown in the figure and therefore not covered. The test execution is shown in the right-hand figure.

System Variable Development and Tests

The final SMR-CL code was derived based on the theoretical case study and tested on a 7 x 5 m test bed with row structures similar to the case study. The final code is presented in Table 2. Table 2 include links to the different situations shown in Figure 1 left. The links are also added a number which represents the order of command activation.

Table 2: The commands for coding the SMR-language enabling the SMR robot to deal with all test bed situations. The SFCM is providing the $wheretogo parameters. The $SFCM is the data log file for coordinates and detection flags of whether rows are detected to the left and right of the centre row. The bold characters represent the SFCM commands send to the SMR during its operational mode. (1 = true, 0 = false)

Link to the situations in Figure 1 left	SMR-CL code	
a^1	*Log ”$SFCM”* *Wait 1* *d = 0.21* *h = 0.2* *drow = 0.3* *right = 1* *left = 2* *tcr = 3* *tcl = 4* *pcfbr = 5* *pcfbl = 6* *pcfbrgs = 7* *pcfblgs = 8*	
a^2,e^1, a^3,e^2, a^4,e^3, $a^5,b^1,e^4,h^1,i^1,k^1,n^1,r^1,t^2,ab^1$ l^1 c^1 x^1,ac^1 f^1,j^1,u^1 o^1,s^1,y^1	*label ”frow”*	*set “rdr” 0* *set “rdl” 0* *fwd 0.1* *followline “ bm” @v h :($nolineb)* *if ($wheretogo ==* ***right)*** *”turnright”* *if ($wheretogo ==* ***left)*** *”turnleft”* *if ($wheretogo ==* ***tcr)*** *”tcr”* *if ($wheretogo ==* ***tcl)*** *”tcl”* *if ($wheretogo ==* ***pcfbr*** | *$wheretogo ==* ***pcfbrgs)*** *”pcfbr”* *if ($wheretogo ==* ***pcfbl*** | *$wheretogo ==* ***pcfblgs)*** *”pcfbl”* *goto “end”*
f^4,j^4,l^2,u^4 f^5,l^3,u^5	*label ”turnright”*	*if ($rowdetectedright == 0) ”turnr90”* *goto “driverightline”*
c^2,o^4,s^4,x^3 c^3,o^5,s^5,x^4	*label ”turnleft”*	*if ($rowdetectedleft == 0) ”turnl90”* *goto “driveleftline”*
	label ”tcr”	*if ($rowdetectedright == 1) ”turnright”* *goto ”end”*
x^2,ac^2 $(x^3),ac^3$	*label ”tcl”*	*if ($rowdetectedleft == 1) ”turnleft”* *goto ”end”*
f^2,j^2,u^2 f^3,j^3,u^3	*label ”pcfbr”*	*if ($wheretogo ==* ***pcfbrgs)*** *“gostraitright”* *goto “turnright”* *goto ”end”*
o^2,s^2,z^2,y^2 o^3,s^3	*label ”pcfbl”*	*if ($wheretogo ==* ***pcfblgs)*** *“gostraitleft”* *goto “turnleft”* *goto ”end”*
j^5 j^6 j^7 j^8 j^9	*label ”turnr90”*	*drive @v -0.15 :($crossingblackliner)* *fwd 0.45* *turn -90* *fwd drow* *goto “frow”*

Link to the situations in Figure 1 left	SMR-CL code	
	label ”turnl90”	*drive @v -0.15 :($crossingblacklinel)* *fwd 0.45* *turn 90* *fwd drow* *goto “frow”*
	label ”gostraitright”	*fwd -0.21* *turn -45* *fwd 0.5 @v 0.1 :($crossingblackline) \|* *($blacklinefound)* *goto “turnr45”* *if (condition) “turnr45”* *if (condition == 1) “frow”* *goto ”end”*
y^3 z^1 z^2 z^3	*label ”gostraitleft”*	*fwd -0.21* *turn 45* *fwd 0.5 @v 0.1 :($crossingblackline) \|* *($blacklinefound)* *goto “turnl45”* *if (condition) “turnl45”* *if (condition == 1) “frow”* *goto ”end”*
	label ”turnr45”	*turn -45* *drive :($drivedist > 0.03)* *stop* *wait 1* *goto ”frow”*
aa^1 aa^2 aa^3 aa^4 aa^5	*label ”turnl45”*	*turn 45* *drive :($drivedist > 0.03)* *stop* *wait 1* *goto ”frow”*
f^6,l^4,u^6 f^7,l^5,u^7 g^1,m^1,v^1 g^2,m^2,w^1 g^3,m^3,w^2 g^4,m^4,w^3 g^5,m^5,w^4	*label ”driverightline”*	*set “conditioncounter” 0* *followline “bmc” @v h* *:($nolinecamr)\|(ntrue(0,$nolineb == 0)>200)* *if (ntrue(0,$nolineb == 0)>200) “frow”* *fwd d* *turn -90* *fwd drow* *turn -90* *goto “frow”* *goto “end”*
c^4,o^6,s^6,x^5 c^5,o^7,s^7,x^6 t^1 d^1,p^1, d^2,q^1, d^3,q^2, d^4,q^3, d^5,q^4,	*label ”driveleftline”*	*set “conditioncounter” 0* *followline “bmc” @v h* *:($nolinecaml)\|(ntrue(0,$nolineb == 0)>200)* *if (ntrue(0,$nolineb == 0)>200) “frow”* *fwd d* *turn 90* *fwd drow* *turn 90* *goto “frow”* *goto “end”*
(x^4),ac^4	*label ”end”*	*Stop*

Table 3: The SFCM commands send to the SMR during the test illustrated in Figure 1 left

Link to the situations in Figure 1 left	SFCM commands	Reason
C	**left**	Row detection to the right (*$rowdetectedright*) is false and only row detection to the left (*$rowdetectedleft*) is true. No area covered to the left or right of the vehicle.
F	**pcfbr**	Row detection to the right and left are both true. The test area has been covered to the left of the vehicle.
J	**pcfbr**	Row detection to the right is false and only row detection to the left is true. Area is covered to the left. Area to the right is uncovered.
L	**right**	Row detection to the left is false and only row detection to the right is true. No area covered to the left or right of the vehicle.
o*	**pcfbl**	Row detection to the right and left are both true. The test area has been covered to the right of the vehicle.
s*	**Pcfbl**	Row detection to the right and left are both true. The test area has been covered to the right of the vehicle.
u*	**Pcfbr**	Row detection to the right and left are both true. The test area has been covered to the left of the vehicle.
X	**Tcl**	Row detection to the left is false and only row detection to the right is true. All area to the right of the vehicle is covered.
Y	**Pcfblgs**	Row detection to the right is false and only row detection to the left is true. All area to the left of the vehicle is covered. Only the headland area, where the vehicle is close to in this situation, is uncovered.

Each provided SFCM command is kept active until the next command is send. Therefore the situations p, t and v use the former SFCM command.

The SMR-CL code listed in Table 2 was validated placing the modified SMR agent at position **a** with the centre line sensor above the first row. The vehicle initiated its mission after the start command was sent. The data logging file was also initiated at **a**. The weeding agent followed the row (**b**) based on its centre placed reflectance sensors using the motion primitive *followline*. The speed of the vehicle was defined by *h*, which was set to 0.2 m/s. The main label *frow* or follow row behaviour was active until the centre sensor cannot detect any row. This is illustrated in situation **c**. The vehicle in this situation reads the system variable *$wheretogo* set by SFCM. It will then move forward by *d* meters in order to complete the current row. *d* is a standard travel distance which in this test is set to 0.21 m. The row detection sensors was placed in front of the weeding agent and *d* meter from the vehicles centre of rotation. The commands provided by SFCM during the test in situation **c**, **f**, **j**, **l**, **o**, **s**, **u**, **x** and **y** are listed in Table 3. Based on SFCM relevant input data, a reason for the SFCM commands given is also listed in Table 3. The SFCM software layer is not yet developed. Therefore, the SFCM decisions during the tests were set by a human supervisor.

In situation **u** to **w** it was found that addition of a reflectance sensor sample counter in the labels *“driveleftline”* and *“driverightline”* was necessary. The tests showed that without the sample counter in these situations, the vehicle captured the headland row before it has covered the whole row from **u** to **v**. The sample counter was set to 200, i.e. 200 samples with black line found by the reflectance sensor are needed before the follow line by the camera is replaced by follow line by the reflectance sensor.

The reason for illustrating the dotted row in the case study is because of a potential low accuracy of the coverage. The coverage will be based on GPS under field conditions and it is assumed that the GPS accuracy could be less than the crop row distance (*drow*). Thus, there is a probability of a missing row leading to a wrong decision made by the SFCM. Therefore a control of adjacent rows is needed like in situation **x**. Activation of the label *"tcl"* is decided by the SFCM in situation **x**. The label reads the system variable *$rowdetectedleft*. If the value is true the label *"turnleft"* is activated and the vehicle will follow the dotted row and thereby make a full coverage of the whole field.

The test results is shown in Figure 1 Right as an overlay of the true map of the test bed, and the x- and y-coordinate data provided by the data logging file sampled during the test.

The x- and y-coordinate data is based on the vehicle odometry. The vehicle was calibrated prior to the test according to methods described by Borenstein et al. (1996). The vehicle was able to drive on a square of 3.75 m. The amount of erroneous rotation in total of the four nominal 90^o turns were 0.5 cm and 1.0 cm in the x- and y-direction respectively. Even though the calibration was successful, the drift of the odometry over time was evident from Figure 1 Right. However, the result clearly indicate that the vehicle was able to cover the whole test bed area based on the SMR-CL programming in Table 2 and the SFCM commands listed in Table 3 provided at each listed situation.

The test revealed that it may be rather simple to develop the SFCM as a software layer to be included in the overall control language as many of the necessary logging facilities are currently available.

5 Conclusion

The adaptation of the SMR-CL language to the context of operating a small vehicle in a row crop domain has clearly indicated the feasibility of applying such a high-level language to an agricultural domain. The relative limited modifications necessary to accommodate the operational requirement of the vehicle shows the flexibility and applicability of the SMR-CL language in terms of new domains. However, the need for a deliberative module is evident in order to cope with recording the progress of the vehicle work in the field.

6 References

ANDERSEN, N.A. AND O. RAVN, 2004. SMR-CL, A Real-time Control Language for Mobile Robots. In Proc. 2004 CIGR International Conference, Beijing, China

ARKIN, R.C., 1989. Towards the Unification of Navigational Planning and Reactive Control. In: Working Notes of the AAAI Spring Symposium on Robot Navigation. Stanford University, March 28-30, 1989

BORENSTEIN, J.; H.R. EVERETT AND L. FENG, 1996. "Where am I?" Sensors and Methods for Mobile Robot Positioning. The University of Michigan, USA

BROOKS, R.A., 1991. Intelligence Without Reason. A. I. Memo 1293:-27

BØRRILD, C., 2000. Introduktion til robottypen "Small Mobile Robot". Section for Automation, Ørsted, The Danish Technical University (DTU), Denmark. (Internal work in Danish)

HAGRAS, H.; M. COLLEY AND V. CALLAGHAN, 2002. Online Learning and Adaptation of Autonomous Mobile Robots for Sustainable Agriculture. Autonomous Robots, 13, 37-52

PAYTON, D.W., 1986. An Architecture for Reflexive Autonomous Vehicle Control. In: IEEE Int. conf. on Robotics and Automation. San Francisco, CA, April 7-10, p. 1838-1845

TILLETT, N.D.; T. HAGUE AND J.A. MARCHANT, 1998. A robotic system for plant-scale husbandry. Journal of Agricultural Engineering Research, 69 (2), 169-178

TILLETT, N.D.; T. HAUGE, AND S.J. MILES, 2002. Inter-row vision guidance for mechanical weed control in sugar beet. Computers and Electronics in Agriculture, 33 163-177

YAVUZ, H. AND A. BRADSHAW, 2002. A New Conceptual Apporach to the Design of Hybrid Control Architecture for Autonomous Mobile Robots. Journal of Intelligent and Robotic Systems 34, 1-26