Available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/issn/15375110

Research Paper

Autonomous navigation using a robot platform in a sugar beet field

Tijmen Bakker [a,*], *Kees van Asselt* [a], *Jan Bontsema* [b], *Joachim Müller* [c,1], *Gerrit van Straten* [a]

[a] *Wageningen University, Systems and Control Group, P.O. Box 17, 6700 AA Wageningen, The Netherlands*
[b] *Wageningen UR Greenhouse Horticulture, P.O. Box 644, 6700 AP Wageningen, The Netherlands*
[c] *Wageningen University, Farm Technology Group, P.O. Box 17, 6700 AA Wageningen, The Netherlands*

ARTICLE INFO

Article history:
Received 8 December 2008
Received in revised form
29 March 2010
Accepted 3 May 2011
Published online 11 June 2011

An RTK-DGPS (Real Time Kinematic Differential Global Positioning System) based autonomous field navigation system including automated headland turns was developed to provide a method for crop row mapping combining machine vision, and to evaluate the benefits of a behaviour based reactive layer in a hybrid deliberate systems architecture. Two experiments were performed at the same time: following of pre-planned paths reconstructed from crop row positions based on RTK-DGPS and crop row mapping by combining vision-based row detection with RTK-DGPS information. The standard deviation, mean, minimum and maximum lateral error of the robot vehicle while following a straight path on the field with RTK-DGPS at a speed of 0.3 m s^{-1} were respectively 1.6, 0.1, −4.5 and 3.4 cm. The standard deviation, mean, minimum and maximum of the heading error were 0.008, 0.000, −0.022 and 0.023 rad. The point-in-polygon algorithm proved to be a suitable method for detection in which part of the field the actuator position coordinates or the field of view of the camera are located. A smooth headland path that connected to the subsequent path along the crop was generated in realtime using a spline based algorithm. The hybrid deliberate software architecture with a behaviour based reactive layer allowed a convenient evaluation of the robot performance. Results from the field experiments showed that the implement can be guided along a defined path with cm precision using an autonomous robot navigating in a field.

1. Introduction

In the future autonomous robotic weed control systems may replace human labour for hand weeding and may also provide a cost effective alternative for herbicides (Slaughter, Giles, & Downey, 2008; Van der Weide et al., 2008). Autonomous robot platforms specially designed for weed control have been described by Tillett, Hague, and Marchant (1998), Åstrand and Baerveldt (2002), Jørgensen et al. (2006) and Ruckelshausen et al. (2006). These platforms use machine vision for navigation relative to the crop row. Autonomous navigation with a robotic platform for agricultural field operations including weed control has also been carried out by absolute navigation with GPS (Bak and Jakobsen, 2004; Noguchi, Reid, Zhang, Will, & Ishii, 2001a; Nagasaka, Umeda, Kanetai, Taniwaki, & Sasaki, 2004; Blackmore, Griepentrog, Nielsen,

* *Corresponding author*. Present address: Tyker Technology, P.O. Box 507, 6700 AM Wageningen, The Netherlands.
E-mail address: tijmen.bakker@tyker.com (T. Bakker).
[1] Present address: University of Hohenheim, Institute for Agricultural Engineering, 70593 Stuttgart, Germany.
1537-5110/$ – see front matter © 2011 IAgrE. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.biosystemseng.2011.05.001

Notation

K_c	controller gain.
K_I	integration gain.
$\epsilon\delta_{img}$	the distance to the row in the image at the image border nearest to the robot frame [m].
$\epsilon\theta_{img}$	the heading of the row in the image [rad].
$\Delta t_{image\ proc}$	the duration of the image processing for row detection of an image [ms].
$\xi = [x\ y\ \theta]^T$	the implement position x, y [m] and heading θ [rad] in the field coordinate system.
ξ_{image}	is $\xi(t-\Delta t_{imageproc})$ and is calculated from the latest logged $\xi(t)$ by linear interpolation.
I_{image}	the length of an image in direction of the x-axis of the coordinate system attached to the vehicle frame [m].
c_1	front position of the row in the latest image in field coordinates.
c_2	rear position of the row in the latest image in field coordinates.
$\epsilon\delta$	the lateral error of the robot vehicle to the row [m].
$\epsilon\theta$	the heading error of the robot vehicle to the row [rad].

Nørremark, & Resting-Jeppersen, 2004; Nørremark, Griepentrog, Nielsen, & Søgaard, 2008). The crop row mapping with a mobile robot navigating with GPS, as presented in this paper, can be seen as a first step to integrate both GPS based absolute navigation and machine vision-based relative navigation. With the robot platform two trials are performed at the same time in a sugar beet field: following of pre-planned paths reconstructed from crop row positions based on RTK-DGPS and crop row mapping by combining vision-based row detection with RTK-DGPS information.

To develop a practical method for RTK-DGPS based autonomous navigation, the autonomous field navigation system was designed to limit required a priori information: the field boundary, the headland boundaries, the row locations and the turning direction at the first headland to arrive at.

At the headland automatic headland turning to the next path to be followed is required. One approach is to calculate one wheel angle setpoint per headland turn based on the radius of a simple arc-shaped headland path trajectory. An odometer uses information to determine when to switch to path following of the new path to be followed, in some cases combined with feedback on the heading obtained from compass or gyroscope (Darr, Stombaugh, & Shearer, 2005; Nagasaka et al., 2004). This method is comparable to the approaches often adopted by field robots competing at the Field Robot Event (Van Henten and Müller, 2007; Van Straten, 2004) such as by the Wurking (Hofstee et al., 2007). It works fairly well only when the distance to travel over the headland is not too long because of the cumulative errors. For longer distances this method is less suitable because the accuracy of the position deteriorates with the distance travelled. Using feedback during the headland turn can avert deviations from the subsequent path to be followed after the turn becomes large. Approaches that use feedback start with the generation of a headland trajectory. The computation of the headland trajectory considers constraints like steering wheel speed, maximum steering angle and headland dimensions (Bell, 1999; Kise, Noguchi, Ishii, & Terao, 2002; Noguchi, Reid, Zhang, & Will, 2001b). Some approaches calculate trajectories that let trailed implements return in front of the next path to be followed (Oksanen, 2007; Rekow & Ohlemeyer, 2007). Some trajectory calculation methods can also avoid obstacles (Linker & Blass, 2008; Vougioukas, Blackmore, Nielsen, & Fountas, 2006).

Design of software is of crucial importance for the functioning of mobile robots. Hybrid deliberate software architecture, for example such as that adopted by Bak and Jakobsen (2004) for an agricultural robotic platform for weed detection, is widely recognised to be the most common architecture for robot software (Orebäck & Christensen, 2003). Blackmore et al. (2007) showed that this architecture is very well suited for automation of agricultural operations because the agricultural environment is usually semi-structured. Deterministic tasks can be optimised, based on known structures within the field, while reactive tasks are carried out in real-time in reaction to local conditions that were not known before the operation started.

The three layers of the hybrid deliberate architecture are a deliberate layer, a reactive layer and a middle layer. The top deliberate layer handles planning and interaction with the operator. The middle layer is the supervisory layer: it bridges the gap between the deliberate and the reactive layer. The reactive layer in the hybrid deliberate architecture is often behaviour based (Orebäck & Christensen, 2003). In a behaviour perception is coupled to action, where mechanisms like subsumption coordinate the interaction of the active individual behaviours resulting in the consequential 'emergent' robot behaviour (Arkin, 1998). The navigation system presented in this paper also has a hybrid deliberate software architecture with a behaviour based reactive layer. Subsumption is used as a coordination mechanism for reactive behaviours like path following, crop row following and stop on emergency thus leading to autonomous behaviour in the field.

The objective of this research was to:

- develop an RTK-DGPS based autonomous field navigation system including automated headland turns.
- develop a method for crop row mapping combining machine vision and RTK-DGPS.
- evaluate the benefits of a behaviour based reactive layer in a hybrid deliberate systems architecture.

2. Platform description

2.1. *Robotic platform*

The robotic platform was specially designed for performing autonomous weed control. During the design process, a methodological approach to engineering design was used to

ensure the quality of the result of the design process and to facilitate the communication and decision making. This design methodology was reported in Bakker, Asselt, Bontsema, Müller, and Van Straten (2010b).

The platform (see Fig. 1) had a ground clearance of 0.5 m, 16 cm wide wheels and 1.5 m track for in-row driving in both 0.5 and 0.75 m rows. The wheelbase was 1.91 m and the length and width of the platform chassis, including a hitch, was 1.75 by 2.80 m. A hinged support at the front part of the chassis, to which both the front wheel modules were connected, guarantees that all wheels were always in contact with the ground. Several implements could be mounted at the centre of the platform underneath the chassis, or at the front bar or at a rear hitch. However, only one hoe holder was connected in the middle of the rear hitch. The platform was four-wheel steered and four-wheel driven. Each of the four wheel modules included a fixed displacement hydraulic motor for propulsion that provided direct drive without gearing. Another fixed displacement hydraulic motor mounted on top of the wheel module steered the wheel via a planetary reduction with a ratio of 1–73.56. Although there were no mechanical constraints on maximum and minimum turning angle of a wheel around its vertical axis, maximum and minimum wheel angles were constrained by the control to plus and minus 111° to prevent twisting of the cables from the wheel speed sensors. Power was provided by a 31 kW Kubota diesel engine that powered two hydraulic pumps mounted in series. The hydraulic transmission to the wheels consisted of a variable displacement pump supplying oil to eight electric proportional valves, four connected to the steering motors and four connected to the drive motors. The pump/valves combination was a 'load sensing' system: the pressure drop over the valves controls the displacement of the pump via an hydraulic load sensing connection and was limited to a small value, independent of load pressure. The second hydraulic pump was connected to two valves, one feeding 3 parallel auxiliary connections for implements, and one connected to a cylinder controlling the hitch height.

2.2. *Hardware and sensors*

The weeding robot electronics consisted of 9 embedded controllers connected by a CAN bus using the ISO 11783 protocol. In the inside of every wheel rim a cogwheel was mounted for wheel speed measurement. The two magneto resistive sensors per cogwheel were placed in such a way that the direction of rotation can be resolved. The rotation of the wheels was measured by these sensors with a resolution of 100 pulses per wheel revolution. The wheel angle of each wheel is measured by a Kverneland 180° sensor with an accuracy of 1° (Kverneland Mechatronics BV, The Netherlands). On each wheel a micro controller was mounted transmitting wheel speed and wheel angle via the CAN bus. Two GPS antennas were used to measure both vehicle position and heading. Both were connected to a Septentrio PolaRx2eH RTK-DGPS receiver (Septentrio Satellite Navigation NV, Belgium) with a specified position accuracy of 1–2 cm and a specified heading accuracy of 0.3° (1 σ). A base station with a Septentrio PolaRx2e RTK-DGPS supplies the RTK-correction signals via a radio connection to the Septentrio PolaRx2eH receiver. For image acquisition a Basler 301fc colour camera with a resolution of 659 by 494 pixels (Basler AG, Germany) was mounted on a platform. The camera was mounted at 0.9 m above the ground looking forward and down with an angle of 37° to the vertical. The area covered by one image was 2.4 m long in the row direction and 1.5 m wide at the image border closest to the camera. Sugar beet is grown at a row spacing of 50 cm, so this means that three rows were visible in the image. This camera is connected via Firewire to a laptop on which the image processing algorithm is implemented. The laptop was connected to the CAN bus. One embedded controller running a real time operating system (PXI platform, National Instruments Corporation, USA) also connected to the CAN bus performed the general vehicle control. The GPS receiver, and a radio modem are connected to the PXI via an RS232 interface. The radio modem interfaced the remote control used for manual control of the weeding robot. Remote manual control was used for transportation to and from the field. The ability to switch remotely to manual control was also required for guaranteeing safety during field trials. Coloured lamps in the signal tower could be operated via a microcontroller to indicate the current status of the robot platform. The platform was further equipped with sensors for measuring diesel level, hydraulic oil level, engine temperature and hitch height. The PXI system gathered wheel angles, wheel speeds, GPS data, remote control data and hitch height and controlled the vehicle by sending messages to the three micro controllers connected to the hydraulic valves. A safety system consisting of four red emergency switches at the corners of the vehicle and a remote switch, sets the valves to the neutral position on activation, overruling the computer control. A laptop connected via ethernet to the PXI system functioned as the user interface of the robot platform.

Fig. 1 – Robot platform.

2.3. *Software architecture*

The three layers of the hybrid deliberate architecture are a deliberate layer, a reactive layer and one middle layer. The top deliberate layer handles planning and interaction with the operator. The middle layer is the supervisory layer: it bridges the gap between the deliberate and the reactive layer. The reactive layer performs repetitive time critical calculations. The reactive layer of the robot platform software is behaviour based, so the

subsystem consists of separate behaviours, where each behaviour has one specified task (Murphy, 2000). The following behaviours were distinguished: CheckActuatorInMidField, CheckRowsInViewOfCam, FollowPath, FollowHeadlandPath, FollowRow, Listen2RemoteControl, StopOutsideField and StopOnEmergency, OperateHitch, OperateLamp, Wheel SpeedControl and WheelAngleControl. These behaviours will be explained more extensively in Section 3. Each behaviour can be released and suppressed. For coordination of behaviours two mechanisms were used: release and suppression. As long as a behaviour was released it is active (Murphy, 2000). Suppression prevented the current signal from being transmitted, and replaced that signal with the suppressing signal (Arkin, 1998). In every behaviour perception was coupled to action, except for the behaviours CheckActu -atorInMidField, CheckRowsInViewOfCam, because these did not result in actuator commands, but were perceived if the implement was in the main field area and if the area of view of the camera was in the main field area, respectively. Depending on the perception of these last two behaviours and information from the deliberate layer, the supervisory layer determined if behaviours were released or suppressed as will be explained more extensively in Section 3.2. If different behaviours act on the same outputs, one behaviour suppresses another behaviour. In this case the behaviours StopOutsideField and StopOnEmergency always had the possibility to suppress other behaviours that act on the wheel speed and wheel angle setpoints.

3. Navigation system

3.1. *Initial parameters*

In the autonomous field navigation experiment on a sugar beet field the navigation was based on RTK-DGPS using preplanned paths reconstructed from crop row positions. In order to check the camera system, crop row mapping was performed by combining vision-based row detection with RTK-DGPS information. The experimental field was a sugar beet field sown in the autumn 2007. During autumn and winter the crop plants remained small and when the experiment was carried out on 14 January 2008 the crop was still in the four true leaves stage. The headland width was 18 m. The prior information needed before navigating on the field were the field and headland boundaries, the row locations and the first turning direction of the robot on the first headland to arrive at. The field and headland boundaries were determined by surveying the eight defining corner positions and the row locations were determined by measuring start and end positions of two rows 1.5 m apart (see Fig. 2). Surveying was carried out by driving the robot under manual control to the different locations so that the GPS antenna was located above the location to be surveyed. A lead line was attached to the GPS antenna to simplify visual determination of the correct robot position. With the surveyed GPS coordinates four files were made, two headland files and a path file, each of them just containing a sequence of GPS coordinates (WGS84 coordinates). The coordinates in the path file were the surveyed begin and end positions of the row locations.

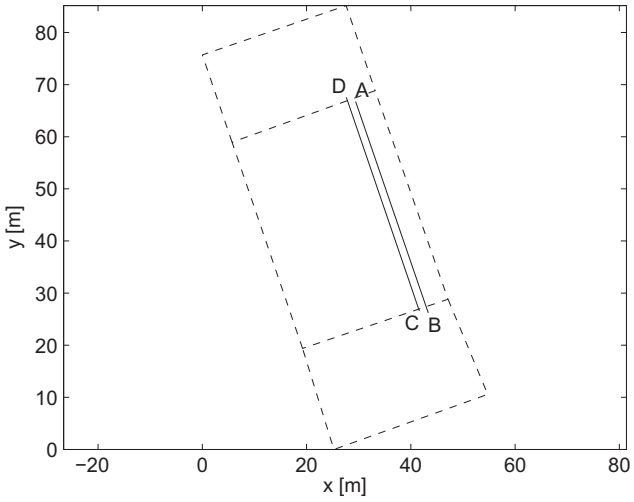

Fig. 2 – Field with a priori information for experiments. Eight surveyed positions define the borders of the two headlands and the main field area. The positions A and B are the surveyed begin and end position of a first row respectively, C and D are begin and end positions of a second row 1.5 m apart from the first one.

At the start of the autonomous navigation the field and headland boundary files were loaded and the minimum values found were chosen as the origin of a local field coordinate system. Next, the field and headland boundaries and path files were calculated to metre scale in the field coordinate system. The speed setpoint for all tests was set to 0.3 m s^{-1}. The user indicated the first turning direction (left or right) for the first headland. This turning direction stayed the same for this headland during navigation over the field, and the turning direction at the other headland was then opposite to the first turning direction.

3.2. *Supervisory layer*

The supervisory layer decided which behaviour was released and which was suppressed depending on information from the deliberate layer and the following perceived conditions:

- Autonomous: whether the robot was in autonomous mode, which was set by the user with the remote control.
- ActInMidField: whether the actuator was in the main field area, which was set by the behaviour CheckActuatorInMidField.
- RowsInViewOfCam: whether the rows were in view of the camera, which was set by the behaviour CheckRowsInViewOfCam.

The following behaviours are always released: StopOnEmergency, WheelSpeedControl, WheelAngleControl, OperateHitch and OperateLamp. From the deliberate layer the supervisory layer knows the robot's mission: navigate autonomously over the field while mapping crop rows. If the supervisory layer perceives the autonomous mode, it stopped releasing Listen2RemoteControl and it released the following

behaviours to accomplish the mission: FollowPath, FollowHeadlandPath, FollowRow, StopOutsideField, CheckIplementOnMidField, CheckRowsInViewOfCam. The behaviours FollowPath, FollowHeadlandPath, FollowRow and StopOutsideField had the same outputs: wheel angle setpoints and wheel speed setpoints. Depending on the perceived conditions the supervisory layer determined by suppression which behaviours setpoints were transmitted to the WheelAngleControl and WheelSpeedControl behaviours as shown in Table 1. StopOutsideField and StopOnEmergency could always suppress the setpoints generated by other behaviours and transmit their setpoints (with zero values) to the WheelAngleControl and WheelSpeedControl behaviours. Because in this study the FollowRow behaviour was only used for crop row mapping, the outputs from the FollowRow behaviour were always suppressed. The robot behaviour for the mission navigation on the field by GPS while mapping crop rows can simply be visualised by Table 1. This table shows under which perceived conditions the different behaviours were released and suppressed.

3.3. *FollowPath*

The path following behaviour controls a user specified position relative to the robot frame. The coordinates of this position in the robot frame to follow the path were set to the position of the hoe in the middle of the rear hitch, because this hoe has to follow the path. The hoe position was calculated from its position in the vehicle frame and from the GPS position and heading. The path was just a sequence of waypoints in field coordinates generated as described in Section 3.1. The lateral error and heading to the path were calculated by a specially designed orthogonal projection. At initialisation this orthogonal projection calculated the smallest lateral error and the corresponding heading error from a path segment to the current position. If no orthogonal projection was possible to any path segment the path segment with the shortest distance between its starting position and the current position was extended and the lateral error and heading error to this vector was calculated. After initialisation the lateral error and heading to the path was calculated by first trying to project on the last segment used and if this was not possible then project to successive segments, at the side connecting to the last segment used, until a projection is possible. The lateral error and heading to the path were controlled by two proportional-integral (PI) controllers (high level control), resulting in low level control setpoints for wheel angles and wheel speeds. A complete description can be found in Bakker, Asselt, Bontsema, Müller, and Van Straten (2007). An improved controller can be found in Bakker, Asselt, Bontsema, Müller, and Van Straten (2010a). The high level PI controller settings were chosen to be $K_c = -2$ and $K_I = 0.5$, for both lateral error and heading error. If the quality of the GPS signal decreases, large lateral and heading errors can occur. Therefore, in that case, the lateral error and heading errors were set to zero.

3.4. *FollowHeadlandPath*

The headland path following was quite similar to path following except that the headland path following generated the headland path by itself. The implemented trajectory generating method was straightforward: there were no obstacles in the field and the feasible paths were not limited by platform constraints because the platform was four-wheel steered and the maximum wheel steering angle is 111°. The headland path must be available well before the robot enters the headland. Therefore FollowHeadlandPath initialised when

Table 1 – Perception conditions under which the supervisory layer releases behaviour with corresponding suppression status during manual control and during autonomous navigation on a field while mapping crop rows (1 = yes, 0 = no).

Perception			Behaviour		
			Name	Released	Suppressed
Autonomous	ActInMidField	RowsInViewOfCam			
1	0	0	FollowPath	1	1
1	0	1		1	0
1	1	0		1	0
1	1	1		1	0
1	0	0	FollowHeadlandPath	1	0
1	0	1		1	1
1	1	0		1	1
1	1	1		1	1
1			FollowRow	1	1
1			StopOutsideField	1	0
1			CheckActuatorInMidField	1	0
1			CheckRowsInViewOfCam	1	0
0			Listen2RemoteControl	1	0
			StopOnEmergency	1	0
			WheelSpeedControl	1	0
			WheelAngleControl	1	0
			OperateHitch	1	0
			OperateLamp	1	0

its suppression status changed to suppressed because of entering the midfield. While being suppressed the headland path following behaviour continually determined its distance to the headland from its position and heading; the robot heading was calculated from the measured robot positions during the last 3 driven m. If the distance of the robot middle position to the headland was less than 3 m, the robot generated a path to the start of the next row to be followed. To create this path firstly the coordinates of the following positions were determined (see Fig. 3).

[*a*] the current robot middle position.
[*b*] position at the end of a vector starting at the robot position extended in forward direction with a heading equal to the robot heading, to a length so that it crosses the main field/headland boundary by a distance of 6.5 m.
[*c*] position at the end of a vector $\overrightarrow{ab}$ made by extending in the direction of *b* with 1.5 m.
[*d*] position at the start of a vector parallel to vector $\overrightarrow{ab}$ at an offset of plus or minus the working width. The sign of the offset of the third headland path vector to the first headland path vector was known from the first turning direction indicated by the user. For the other headland the turning direction was opposite to the first one.
[*e*] position at the end of a vector starting in *d*, parallel to $\overrightarrow{ab}$, extended to a length so that it crosses the headland boundary by a distance of 6.5 m.
[*f*] position at the end of a vector made by extending $\overrightarrow{de}$ in the direction of *e* with 1.5 m.

The polygon {*b*, *c*, *f*, *e*} (see Fig. 3) was used as the control polygon to define a quadratic uniform B-spline curve consisting of 157 positions. Positions *a* and *d* were added to the start and end of these positions and so formed the generated headland path. The robot middle position was set to follow the headland path because the function of headland path following was to turn the platform and not to perform an operation with the implement. The robot middle position was located exactly in between the four wheels. The robot followed the generated headland path if the headland path following was released and if it was not suppressed. The orthogonal projection and control during following the headland path was the same as described in Section 3.3, except that two P controllers are used. During turning the errors increased quite rapidly because of the constraints on the wheel angles. To avoid overshoot, due to the integrating action, and because fast response during turning is much more important then high accuracy, P controllers were used. The P controller settings for both the lateral error controller and the heading error controller were $K_c = -2$.

3.5. *FollowRow*

The FollowRow behaviour controlled a position relative to the robot frame to follow a crop row. The coordinates of this position in the robot frame to follow the crop row was set to the position of the hoe in the middle of the rear hitch. In the future, the FollowRow behaviour could replace the FollowPath behaviour when in between the rows. FollowRow calculated the lateral and heading error of the hoe mounted at the back of the robot relative to the crop row by combining absolute RTK-DGPS positioning with crop row detection by machine vision at the front of the robot. The machine vision algorithm for crop row detection was described in Bakker et al. (2008). If the rows were in view of the camera, as indicated with RowsInViewOfCam, and the quality of the GPS signal was good, the image processing determined for every image:

[$\epsilon\delta_{img}$] the distance to the row in the image at the image border nearest to the robot frame [m].
[$\epsilon\theta_{img}$] the heading of the row in the image [rad].
[$\Delta t_{image\ proc}$] the duration of the image processing for row detection of an image [ms].

Fig. 4 shows the robot near a crop row in the field coordinate system (x_{field}, y_{field}). Let:

[$\xi = [x\ y\ \theta]^T$] the implement position x, y [m] and heading θ [rad] in the field coordinate system.
[ξ_{image}] is $\xi(t - \Delta t_{imageproc})$ and is calculated from the latest logged $\xi(t)$ by linear interpolation.
[I_{image}] the length of an image in direction of x_v [m].
[c_1] front position of the row in the latest image in field coordinates.

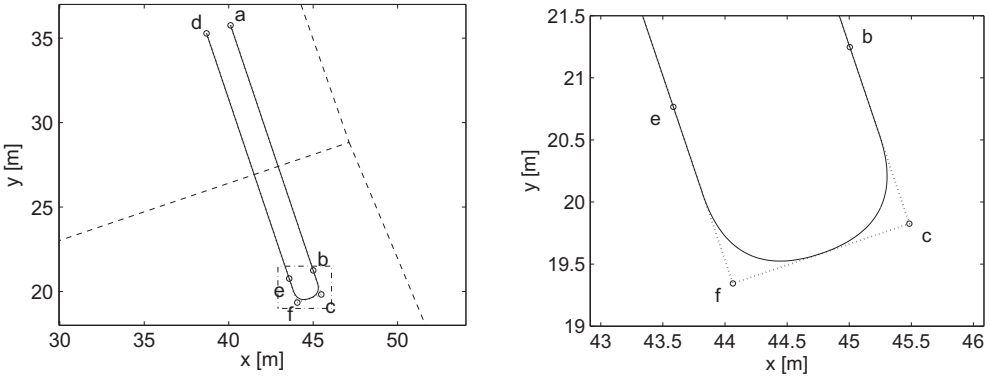

Fig. 3 – Headland path (—) generated from polyline {*a*, *b*, *c*, *f*, *e*, *d*} (···). Part of the left figure indicated by the box (· - ·) in the left figure is displayed enlarged in the right figure.

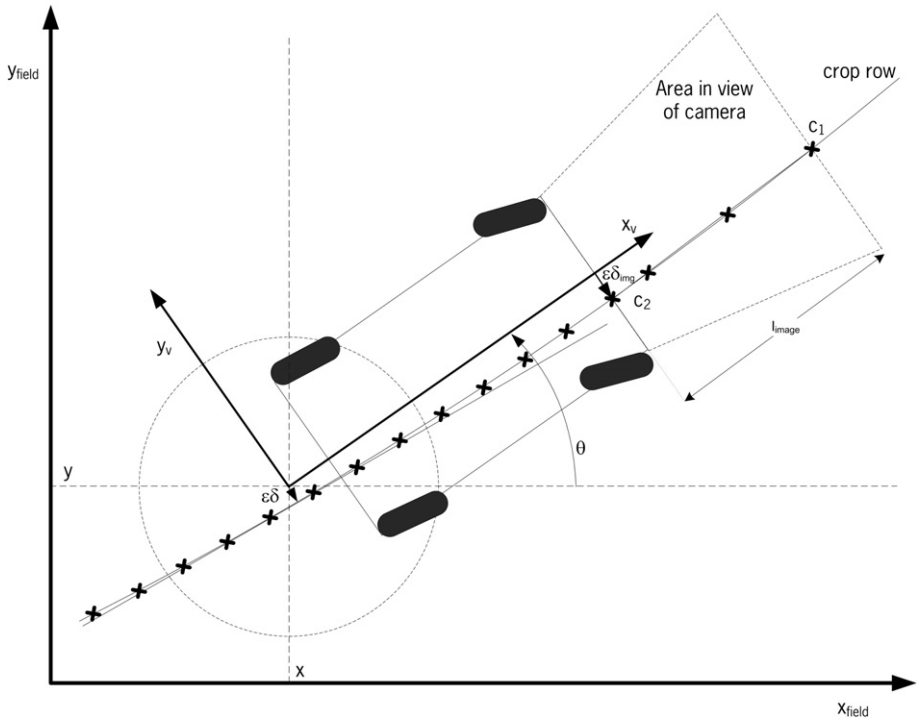

Fig. 4 – Robot driving along a crop row in the field coordinate system.

[c_2] rear position of the row in the latest image in field coordinates.
[$\epsilon\delta$] the lateral error of the robot vehicle to the row [m].
[$\epsilon\theta$] the heading error of the robot vehicle to the row [rad].

If the rows were in view of the camera, and if the difference between the last implement position ξ measured $>$ 2 cm (so the robot is moving), for every new image the coordinates of c_1 and c_2 were calculated. The coordinates of c_1 and c_2 followed from ξ_{image}, I_{image}, $\epsilon\delta_{img}$ and $\epsilon\theta_{img}$ and from the position of the image relative to the vehicle frame x_v, y_v. The coordinates of c_1 and c_2 were added to an array C, indicated by the crosses in Fig. 4. The array C was cleared if RowsInViewOfCam changed from 0 to 1 or on a release of the FollowRow behaviour. C_{reg} was a subset of C including only the positions c_i that were located within a certain distance r from the current robot position ξ. The distance r was equal to the radius of the circle in Fig. 4 and is equal to 1.5 m. A line m was fitted using least squares linear regression through the positions C_{reg}. The lateral error $\epsilon\delta$ and the heading error $\epsilon\theta$ of the robot vehicle to the crop row was now calculated to this line m. These errors were used in the same way for control as in the method described in Section 3.3.

3.6. *Other behaviours*

StopOutsideField calculated if one of the positions of the robot contour was outside the main field area polygon and outside both headland polygons with the point-in-polygon algorithm (Burrough & MacDonnell, 1998). The robot contour in field coordinates was calculated from the robot position and heading and from the robot dimensions. If a robot contour coordinate was located outside the polygons, the wheel angle setpoints and wheel speed setpoints were set to zero.

CheckActuatorInMidField checked if the robot's actuator position was inside the main field area polygon with the point-in-polygon algorithm and then sets ActInMidField accordingly. The actuator position in field coordinates was calculated from the robot position and heading and from the actuator position within the robot coordinate system.

CheckRowsInViewOfCam checked if the contour of the area covered by the camera performing row recognition was inside the main field area polygon with the point-in-polygon algorithm and then sets RowsInViewOfCam accordingly. The contour of the area covered by the camera in field coordinates was calculated from its coordinates in the robot coordinate system and from the robot position and heading.

Listen2RemoteControl converted the remote control joystick position to wheel angle and wheel speed setpoints depending on the selected steering mode resulting in setpoints for wheel angles and speeds of two virtual wheels, one between the front wheels and one between the rear wheels. Possible steering modes were: front wheel, rear wheel, four-wheel opposite and four-wheel parallel steering. The setpoints for each of the four individual wheels followed from the vehicle model (Bakker et al., 2010a).

StopOnEmergency made the wheel setpoints zero if one of the following conditions applied:

- there was an emergency indicated by the remote control
- there was an emergency indicated by the remote switch held by a safety officer.
- the remote control was out of range

- a communication error existed in the CAN communication. This could happen when not all modules are communicating.

Wheel speed control was carried out by one PI controller per wheel. Wheel angle control was carried out by a P controller combined with a Smith predictor to compensate for time delays. This was described more extensively in Bakker et al. (2010a).

OperateHitch switched the hitch position between a minimum hitch height if the ActInMidField indicated that the actuator was in the main field area and maximum hitch height if the actuator was not in the main field area. Controlling the hitch height was carried out with a simple on/off control.

OperateLamp switched the different lamps of the lamp-stack on and off. The five different lamps showed if FollowHeadlandPath was suppressed, if the remote control Emergency button was pressed, if the GPS receiver had an accurate position, and if the robot was in manual control mode or in autonomous mode.

4. Results

4.1. *Path following along crop rows*

Fig. 2 shows the a priori information of the field as measured before the experiment, consisting of the corners of the field and headland boundaries and the begin and endpoints of some row locations.

Figs 5 and 6 show the performance of the path following along the first and the second path. From $t = 4$ s to $t = 144$ s the robot drove along the line A–B. The standard deviation, mean, minimum and maximum lateral error of the robot vehicle to the path were 1.6, 0.1, −4.5 and 3.4 cm respectively. Between $t = 144$ s and $t = 199$ s the robot was turning at the headland, so the FollowPath behaviour was not released. From $t = 199$ s to 355 s the robot drove along the line CD. The standard deviation, mean, minimum and maximum lateral error of the robot vehicle to the path were 1.8, 0.00, −10.9, and 6.0 cm respectively. The figure shows a relatively large lateral error when the actuator was near the end of the path CD. This was possibly due to roll caused by the bumpier

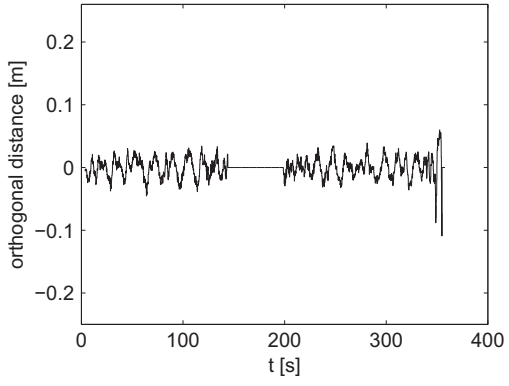

Fig. 5 – lateral error of the robot vehicle to the path during path following over AB from $t = 4$ s to $t = 144$ s and CD from $t = 199$ s to $t = 355$ s.

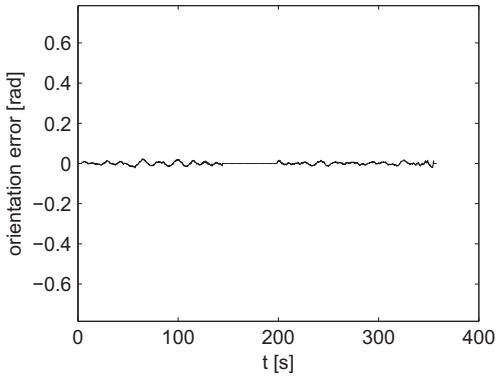

Fig. 6 – Robot vehicle heading error during path following over AB from $t = 4$ s to $t = 144$ s and CD from $t = 199$ s to $t = 355$ s.

headland, because here the actuator was near the end of the path and the robot wheels were already on the headland. When these last measurements were excluded the standard deviation, mean, minimum and maximum lateral error over $t = 199$ to $t = 340$ were 1.4, 0, −3.6 and 3.9 cm respectively, which is comparable to the values from the first path. Fig. 6 shows the heading of the robot vehicle relative to the path. The standard deviation, mean, minimum and maximum of the heading error were 0.008, 0.000, −0.022 and 0.023 rad from $t = 4$ s to $t = 144$ s and 0.007, 0.000, −0.021 and 0.017 from $t = 199$ to $t = 355$ s. The accuracy was very good and RTK-DGPS based path following was very suitable for situations where the crop rows are known from previous treatments such as sowing (Griepentrog, Nørremark, Nielsen, & Blackmore, 2005).

4.2. *Path following on the headland*

Fig. 7 shows the headland path generated while following the path and the driven route. The realtime generated headland path was in line with the position of the crop row and also with the driven route.

Figs. 8 and 9 show the lateral error to the headland path and the heading error while following the headland path. At the start of the headland path following at $t = 144.1$ the lateral error was 1.7 cm and the heading error was 0.00 rad while the last lateral error of the robot vehicle from the path AB at that time was 2.1 cm and the heading error −0.01 rad, thus indicating that the transition from path following to headland path following occurred smoothly. The maximum lateral error and the maximum heading error during the turn were respectively 52 cm and 0.72 rad. This headland path, with such a limited radius could not be achieved by the robot controller due to the constraints on the wheel angles. However, on the headland it is less important to follow the headland path exactly than to arrive in front of the proper crop rows again. After the sharp turn in the headland path, the errors reduced and when the actuator left the headland at $t = 209$ s, the lateral error and the heading error of the vehicle to the path reconstructed from the crop row positions were only 0.1 cm and 0.00 rad respectively.

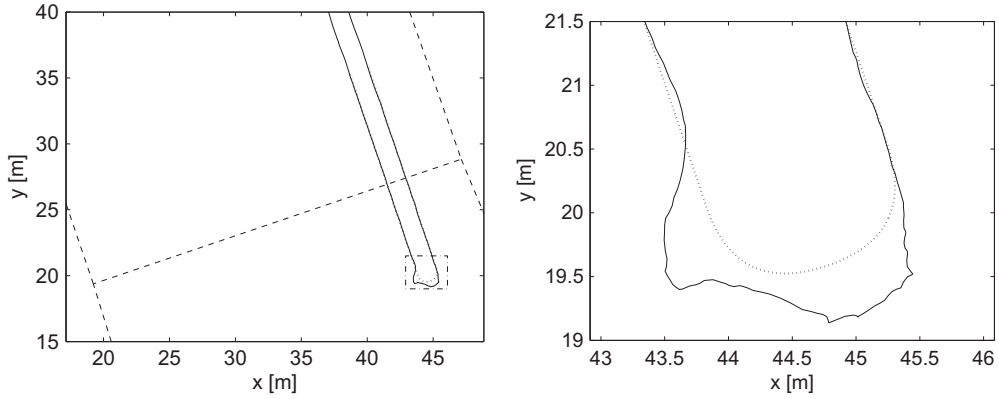

Fig. 7 – Route (—) and path (···) during headland path following over BC. Part of the left figure indicated by the box (· - ·) in the left figure is displayed enlarged in the right figure.

4.3. *Crop row mapping*

Currently, in most situations no accurate absolute crop row location information is available to be used for guidance of a weeding robot. Our original idea was to follow crop rows by camera alone, and to combine this with the headland path following as described in Section 4.2. However, currently the robot can follow the crop rows based on camera alone, but only with a constant offset of about 25 cm. The reason for this has not been explained so far. Therefore, it was decided to firstly just map the crop rows with the camera and the GPS, and then compare the lateral errors and heading errors with those from path following. The hybrid deliberate software architecture permits this comparison by a mission that releases the behaviours FollowPath, FollowHeadlandPath and FollowRow, but suppresses permanently the outputs of FollowRow. Fig. 10 shows the surveyed crop rows, the driven route and the mapped crop rows on the field. In Figs. 11 and 12 the lateral error and the heading error of the vehicle to both path and the crop rows as reconstructed by the vision system are shown. The error to the path and the errors to the crop rows do not match exactly, this could be explained by the fact that the path and the crop rows were not located at exactly the same position. The used path was a straight line in between two surveyed positions, but in agriculture a row is never exactly a straight line. If RTK-DGPS-based steered tractor is used during sowing then the rows could be quite straight, but in our case the seeding tractor was steered manually. When following path AB, path and mapped row were quite close. When following path CD the error to the crop row increased. This was probably caused by not exactly positioning point D above the crop row. The lateral errors measured by the crop row following at the start and end of each crop row were larger than in the middle part. As described in Section 3.5, all row positions within a distance of 1.5 m from the actuator position were used for estimating the row position, but at the start the number of row positions used are limited and the average distance of the actuator to the positions used is larger until the actuator passes the front of the first image taken by a distance larger than 1.5 m. This could explain the larger lateral error. After this the lateral error was reduced because the number of measured row positions being used for row location estimation increased and stayed more or less constant because of the constant driving speed and also because the average distance of the row positions from the actuator position was approximately constant. At the end of the row, if the actuator was closer than 1.5 m from the rear border of the last image taken before entering the headland, the quality of the row location estimation reduced again. This was because the number of positions used for crop row estimation reduced

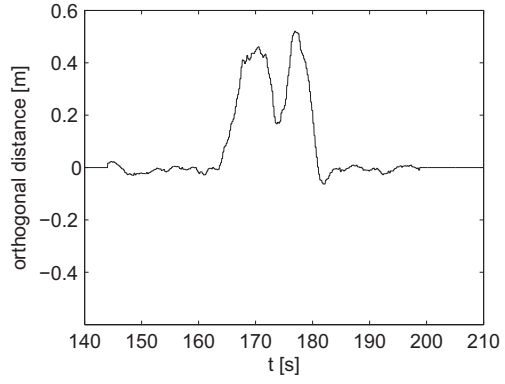

Fig. 8 – Lateral error of the robot vehicle to the headland path during headland path following over BC.

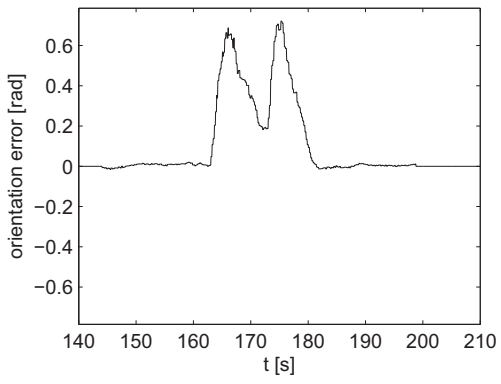

Fig. 9 – Robot vehicle heading error during headland path following over BC.

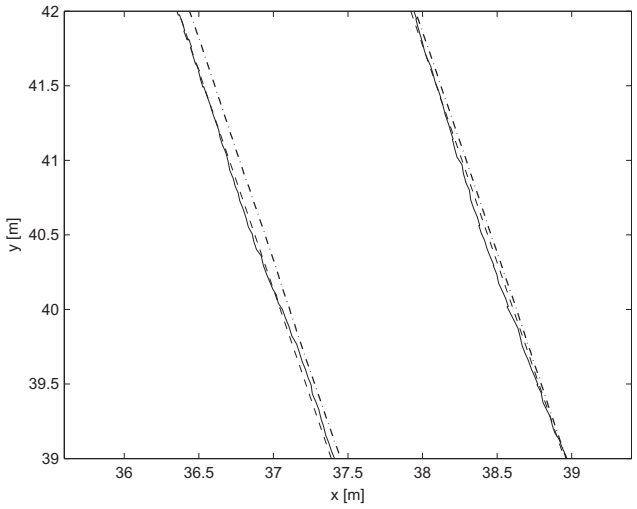

Fig. 10 – Path (- - -), driven route measured with GPS (—) and mapped rows (· - ·) in the field.

and the average distance of these positions to the actuator position increased from here. This possibly resulted in a less accurate row location estimation and in a larger lateral error. So while there were differences when comparing the lateral errors of path following and crop row following, they could still be explained. The row detection worked quite well and showed good perspective for guidance.

5. Discussion

The standard deviation, mean, minimum and maximum lateral error of the robot vehicle following a straight path on the field with RTK-DGPS of 1.6, 0.1, −4.5 and 3.4 cm respectively are very good results. But it should be kept in mind that the position of the robot vehicle was calculated from the antenna positions that were located about 2 m above the soil surface, and are not corrected for roll. In our case the even soil surface did not cause any problems, but to be able to navigate

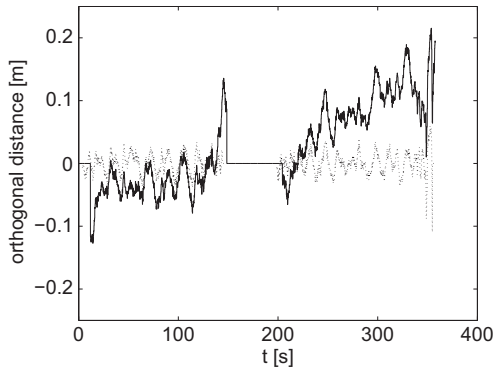

Fig. 11 – Lateral error of the robot vehicle to the crop row as measured by the vision system (—) and to the path (···) during path following over AB from $t = 4$ s to $t = 144$ s and CD from $t = 199$ s to $t = 355$ s.

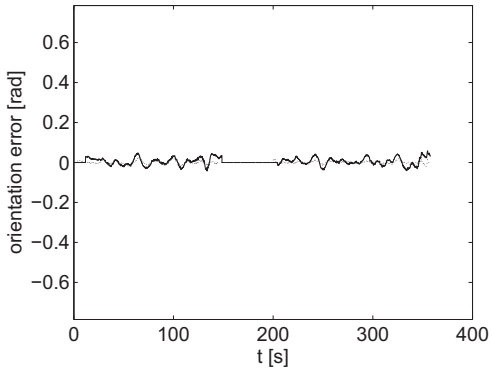

Fig. 12 – Robot vehicle heading error to the crop row as measured by the 767 vision system (—) and to the path (· · ·) during path following 768 over AB from $t = 4$ s to $t = 144$ s and CD from $t = 199$ s to $t = 355$ s.

precisely under bumpier conditions, roll compensation will be required.

The results showed that at the moment the actuator left the headland path, it was located almost in line with the new row. However, in practice there could be a bigger offset when the seeder tramlines are not exactly one working width apart. However, even if there is a small mismatch between the end of the headland path and the new crop row, the actuator could be expected to arrive in line with the new row, because the actuator position starts to be controlled based on feedback from the camera vision system which starts when the field of view of the camera has left the headland. It then takes about the robot length to drive forward before the actuator leaves the headland, which should be enough time to control the actuator position in line with the crop row.

The speed of 0.3 m s^{-1} used in the experiment was slow compared to the 1 m s^{-1} maximum speed assessed to be achievable with intra-row weeding but is close to the 0.5 m s^{-1} used in recently reported experiments (Tillett, Hague, Grundy, & Dedousis, 2008). The maximum speed of autonomous robots that will replace hand weeding will be constrained by the dynamics of the intra-row actuator. Other applications such as inter-row weeding require higher speeds.

The StopOutsideField behaviour stopped the robot if any part of the robot is outside the field boundary. However, it could be reasonable to continue the operation if there is no fence or ditch etc. around a field and the tool is still inside the field. A solution could be to enlarge slightly the field area polygon in these cases.

The hybrid deliberate approach was very helpful, especially because this clear architecture helped in neatly organising the software, but even more because it enabled separate testing of the different behaviours: the path following behaviour controlled the vehicle while the suppressed crop row following behaviour was tested. The agents of behaviours architecture (García-Pérez, García-Alegre, Ribeiro, & Guinea, 2008) also allows the execution of the deliberative and reactive tasks by concurrent acting perceptual and motor agents and a coordination agent, but the peer to peer message passing between the agents implies a high dependency between the agents. In the architecture proposed in this paper the reactive

layer consists of a set of independent behaviours, each consisting of both perception and action, just coordinated by the sequencer layer with no peer to peer communication between the behaviours. The constellation of behaviour to get a total behaviour for the application of a weeding robot can be described easily with a simple table. The hybrid deliberate architecture implemented in an agricultural robot by Bak and Jakobsen (2004) does not include a reactive layer built from different behaviours, which can be understood from the fact that the focus has mainly been on the reactive behaviour of the path tracking control. To cover the increased complexity of robot software performing a task in a field, a good approach is to build the reactive layer as a constellation of behaviours including e.g. navigation, emergency handling and obstacle avoidance. With this architecture the functionality of the robot is also easily extendible by just adding new behaviours and by some small changes to the deliberate and sequencer layer.

Greater complexity is in the perception of the lateral error and heading error of the implement to the row. Crop row location data filtering required to handle the variability of the output of the crop row detection when guidance of the implement position along the crop row is required.

6. Conclusion

The autonomous navigation system enabled the robot to navigate autonomously on a field based on RTK-DGPS. Standard deviation, mean, minimum and maximum lateral error of the robot vehicle following a straight path in the field with RTK-DGPS at a speed of 0.3 m s^{-1} were respectively 1.6, 0.1, −4.5 and 3.4 cm. Standard deviation, mean, minimum and maximum of the heading error over the same distance were 0.008, 0.000, −0.022 and 0.023 rad.

Further improvements in accuracy of path following of a straight path are not to be expected, because of the standard deviation of 1.6 cm is approximately equal to the RTK-DGPS accuracy. The point-in-polygon algorithm proved to be a suitable method for detection in which part of the field given coordinates such as the actuator position or the field of view of the camera are located. With the spline based algorithm a smooth headland path is generated in realtime that connects the subsequent paths along the crop.

The crop row mapping shows results with good perspective for guidance of the robot.

It has been shown that it is possible to apply the hybrid deliberate architecture with a behaviour based reactive layer for the case of autonomous weeding. The benefits are that the behaviours can be tested separately, that the architecture of the software is very transparent and that the software is easily extendible.

Further research should focus on roll compensation of the position measurement for navigation on fields that have partly inclined surfaces or are completely inclined, on evaluation of the system at higher speeds, on equipping the robot with hoes to demonstrate autonomous hoeing of a field and on further developing the crop row following using vision to enable autonomous navigation independent of the availability of a priori information of the crop rows.

REFERENCES

Åstrand, B., & Baerveldt, A. J. (2002). An agricultural mobile robot with vision-based perception for mechanical weed control. *Autonomous Robots, 13*(1), 21–35.

Arkin, R. (1998). *Behavior-based robotics*. Cambridge, USA: The MIT Press.

Bak, T., & Jakobsen, H. (2004). Agricultural robotic platform with four wheel steering for weed detection. *Biosystems Engineering, 87*(2), 125–136.

Bakker, T., Van Asselt, C. J., Bontsema, J., Müller, J., Van Straten, G., 2007. Path following with a robotic platform. In: Agricontrol 2007. 2nd IFAC international conference on modeling and design of control systems in agriculture. Osijek Croatia, pp. 153–158.

Bakker, T., Van Asselt, C. J., Bontsema, J., Müller, J., Van Straten, G. (2010a) A path following algorithm for mobile robots. *Autonomous Robots, 29*(1), 85–97.

Bakker, T., Van Asselt, C. J., Bontsema, J., Müller, J., & Van Straten, G. (2010b). Systematic design of an autonomous platform for robotic weeding. *Journal of Terramechanics, 47*(2), 63–73.

Bakker, T., Wouters, H., Van Asselt, K., Bontsema, J., Tang, L., Müller, J., et al. (2008). A vision based row detection system for sugar beet. *Computers and Electronics in Agriculture, 60*(1), 87–95.

Bell, T., 1999. Precision robotic control of agricultural vehicles on realistic farm trajectories. Ph.D. thesis, Stanford University.

Blackmore, S., Fountas, S., Vougioukas, S., Tang, L., Sorensen, C., & Jorgensen, R. (2007). Decomposition of agricultural tasks into robotic behaviors. *Agricultural Engineering International: The CIGR Ejournal, IX*. Manuscript PM 07 006.

Blackmore, B. S., Griepentrog, H. W., Nielsen, H., Nørremark, M., Resting-Jeppersen, J., 2004. Development of a deterministic autonomous tractor. In: *CIGR International conference*. Beijing.

Burrough, P., & MacDonnell, R. (1998). *Principles of geographical information systems*. Oxford University Press.

Darr, M. J., Stombaugh, T. S., & Shearer, S. A. (2005). Controller area network based distributed control for autonomous vehicles. *Transactions of the American Society of Agricultural Engineers, 48*(2), 479–490.

García-Pérez, L., García-Alegre, M. C., Ribeiro, A., & Guinea, D. (2008). An agent of behaviour architecture for unmanned control of a farming vehicle. *Computers and Electronics in Agriculture, 60*(1), 39–48.

Griepentrog, H. W., Nørremark, M., Nielsen, H., & Blackmore, B. S. (2005). Seed mapping of sugar beet. *Precision Agriculture, 6*(2), 157–165.

Hofstee, J. W., Jansen, R., Nieuwenhuizen, A., Govers, M. H. A. M., Blaauw, S. K., Van Willigenburg, L. G., et al. (2007). Wurking - an autonomous robot for agriculture applications. In *Proceedings of the 5th field robot event 2007*. Wageningen, The Netherlands: Farm Technology Group.

Jørgensen, R., Sørensen, C., Pedersen, J., Havn, I., Jensen, K., Søgaard, H., et al. (2006). Hortibot: a system design of a robotic tool carrier for high-tech plant nursing. In M. Rothmund, M. Ehrl, & H. Auernhammer (Eds.), *Automation technology for off-road equipment 2006. Landtechnik Weihenstephan*, (pp. 13–21), Bonn, Germany.

Kise, M., Noguchi, N., Ishii, K., & Terao, H. (2002). Enhancement of turning accuracy by path planning of robot tractor. In Q. Zhang (Ed.), *Automation technology for off-road equipment* (pp. 398–404), Chicago, Illinois, USA.

Linker, R., & Blass, T. (2008). Path-planning algorithm for vehicles operating in orchards. *Biosystems Engineering, 101*(2), 152–160.

Murphy, R. (2000). *Introduction to AI Robotics*. Cambridge, USA: The MIT Press.

Nagasaka, Y., Umeda, N., Kanetai, Y., Taniwaki, K., & Sasaki, Y. (2004). Autonomous guidance for rice transplanting using

global positioning and gyroscopes. *Computers and Electronics in Agriculture, 43*(3), 223–234.

Noguchi, N., Reid, J. F., Zhang, Q., Will, J., 2001b. Turning function for robot tractor based on spline function. In: *Proceedings of the 2001 ASAE Annual International Meeting. Sacramento*, California, USA.

Noguchi, N., Reid, J., Zhang, Q., Will, J., Ishii, K., 2001a. Development of robot tractor based on rtk-gps and gyroscope. In: *Proceedings of the 2001 ASAE Annual International Meeting*. ASAE Paper 01–1195.

Nørremark, M., Griepentrog, H. W., Nielsen, J., & Søgaard, H. T. (2008). The development and assessment of the accuracy of an autonomous gps-based system for intra-row mechanical weed control in row crops. *Biosystems Engineering, 101*(4), 396–410.

Oksanen, T., 2007. *Path planning algorithms for agricultural field machines*. Phd thesis, Helsinki, Finland: Helsinki University of Technology, .

Orebäck, A., & Christensen, H. I. (2003). Evaluation of architectures for mobile robotics. *Autonomous Robots, 14*(1), 33–49.

Rekow, A. K. W., & Ohlemeyer, H. (2007). Automated headland turns the next step in automated agricultural machines. In. *Landtechnik - AgEng 2007, vol. 2007* (pp. 199–209), Hannover, Germany.

Ruckelshausen, A., Klose, R., Linz, A., Marquering, J., Thiel, M., & Tölke, S. (2006). Autonomous robots for weed control. [[autonome roboter zur unkrautbekämpfung]]. *Journal of Plant Diseases and Protection, 20*, 173–180.

Slaughter, D. C., Giles, D. K., & Downey, D. (2008). Autonomous robotic weed control systems: a review. *Computers and Electronics in Agriculture, 61*(1), 63–78.

Tillett, N. D., Hague, T., Grundy, A. C., & Dedousis, A. P. (2008). Mechanical within-row weed control for transplanted crops using computer vision. *Biosystems Engineering, 99*(2), 171–178.

Tillett, N. D., Hague, T., & Marchant, J. A. (1998). A robotic system for plant-scale husbandry. *Journal of Agricultural Engineering Research, 69*(2), 169–178.

Van der Weide, R. Y., Bleeker, P. O., Achten, V. T. J. M., Lotz, L. A. P., Fogelberg, F., & Melander, B. (2008). Innovation in mechanical weed control in crop rows. *Weed Research, 48*(3), 215–224.

Van Henten, E. J., & Müller, J. (2007). The field robot event; an international design contest in agricultural engineering. In. *Landtechnik - AgEng 2007, vol. 2007* (pp. 169–174), Hannover, Germany.

Van Straten, G. (2004). Field robot event, wageningen, 5–6 June 2003. *Computers and Electronics in Agriculture, 42*(1), 51–58.

Vougioukas, S., Blackmore, S., Nielsen, J., & Fountas, S. (2006). A two-stage optimal motion planner for autonomous agricultural vehicles. *Precision Agriculture, 7*(5), 361–377.