Proceedings 2007

The Farm Technology Group organises

June | 14 | 15 | 16 | 2007

Wageningen, the Netherlands

invitation

Field Robot Event

5th edition

2007

www.fieldrobot.com

Workshops

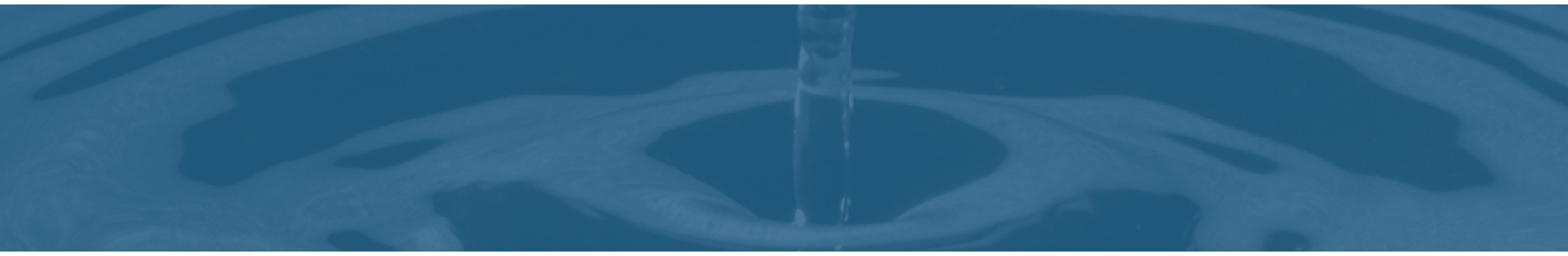
Party

Prototype contest

Bring your own prototype

Photo: Michel Govers

WAGENINGEN UNIVERSITY
WAGENINGEN UR



WAGENINGEN UNIVERSITY
WAGENINGEN UR

Proceedings of the 5th Field Robot Event 2007

Wageningen, June 14, 15 & 16 2007

ISBN	:	978-90-8585-168-4
Date	:	December 2007
Publisher	:	Farm Technology Group
Address	:	Technotron (Building 118) Bornsesteeg 59 6708 PD Wageningen The Netherlands Tel: +31 317 482980 Fax: +31 317 484819
Contact	:	info@fieldrobot.nl

Sponsors Field Robot Event 2007

AMAZONE

AGRI TECHNICA

The World's No.1

Hanover, Germany
13-17 November 2007
Preview days 11/12 November

PLANT RESEARCH INTERNATIONAL
WAGENINGEN UR

kverneland group

CLAAS STIFTUNG

JOHN DEERE

NATIONAL INSTRUMENTS

MTL

MÜLLER Elektronik

RE|PRO|TEX|
relatiegeschenken & promotietextiel

WAGENINGEN UNIVERSITY
WAGENINGEN UR

Preface

In summer 2003, when the 1st Field Robot Event was “born” at Wageningen University, it was an experiment to combine the “serious” and “playful” aspects of robotics to inspire the upcoming student generation. Specific objectives have been:

- Employing students’ creativity to promote the development of field robots,
- Promoting off-curriculum skills like communication, teamwork, time management and fundraising,
- Attracting public interest for Agricultural Engineering,
- Creating a platform for students and experts to exchange knowledge on field robots.

Driven by the success of the previous events in Wageningen in 2003, 2004 and 2005 and the Field Robot Event hosted by Hohenheim University in Stuttgart, Germany in 2006, Wageningen University organized the 5th Field Robot Event on June 14-16, 2007. This event was accompanied by a workshop and a fair where the teams were able to present their robots. Teams also had to write a paper describing the hard- and software design of their robot. These papers collected in this ‘Proceedings of the 5th Field Robot Event’ are a very valuable source of information. This edition of the proceedings ensures that the achievements of the participants are now documented as a publication and thus being accessible as basis for further research. Moreover, for most of the student team members it is the first (scientific) publication in their career - a well deserved additional reward!

Wageningen, December 2007
Eldert van Henten,
Chairman 5th Field Robot Event – 2007

Index

Amaizeing

Modular sensor platform for autonomous agricultural applications

Steffen Meinke, Andreas Ganseforth, Dennis Hagen, Thomas Eichler, Markus Kreienbaum, Matthias Gebben, Tobias Niermann, Jens Egbers, Stefan Haller, Arne Siebe, Dimitri Willms

University of Applied Sciences Osnabrück, Faculty of Engineering and Computer Science/ Interdisciplinary Research Center Intelligent Sensor Systems (ISYS), Albrechtstr. 30, 49076 Osnabrück/Germany, Phone +49541 969 2090, Email: a.ruckelshausen@fhos.de

Abstract

The autonomous vehicle Amaizeing has been developed to participate at the Field Robot Event 2007 (Müller et al. 2006). The requirements for the project team which consists of 12 students, was to make a concept and to realise a low-cost microcontroller based platform. The robot has to consist the rough field conditions and has to be able to fulfil the tasks of the event. The Team benefits of the previous developments of field robots at the University of Applied Sciences. The system architecture is based on a modularly, bus oriented sensor platform. The field robot has shown that low-cost robot systems are able to successfully fulfil tasks within a broad range of field conditions. Due to varying boundary conditions the complexity of such systems is rather high (VAN HENTEN et al. 2006, RATH and KAWOLLEK 2006).

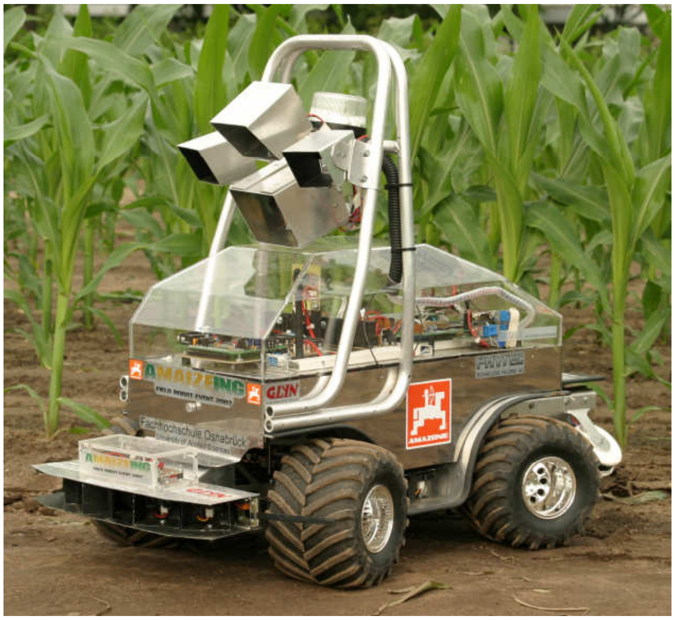


Fig. 1: Field robot Amaizeing

Keywords: **Field Robot Event, autonomous robots, sensor fusion, modular sensor concept, maize**

1 Introduction

The 5th Field Robot Event takes place at Wageningen UR from 14th to 16th of June 2007. The University of Applied Sciences, Osnabrueck participated for the fourth time. Already becoming a tradition, a group of students designed a robot to compete the –partly new designed- tasks of the event. The team of 12 students of different departments implemented within a time of 3 months parallel to their studies 8 microcontrollers and 19 sensors in the Amaizeing, their autonomous vehicle is based on a monster truck model.

This are the tasks of this years field robot event.

- **Robust navigation in a maize field with curved rows**

In a time period of 3 minutes the robot has to drive off as much distance as possible, while navigating within curved rows of maize. At the end of the rows the robot has to turn to the neighbouring next row. The task should be absolved in most fast, accurate, tough and smooth way.

- **Advanced robust navigation in a maize field with straight rows**

In the second task the robot has to drive along a predefined pattern within straight maize rows. Arriving at the headland, the robot should drive to a predefined row (e.g. the fourth row on the left side). The predefined pattern is presented one hour before the competition and in the meantime there is no testing is allowed. To enhance the complexity, plants were missing in either one or both rows for a maximum length of 1 m. The headland width is limited to 1.5 m of length.

- **'Weed' – control in a maize field**

While driving straight maize rows, the robot has to detect randomly placed yellow golf balls during the third task, representing weed. When the robot detects this 'weed' it signalizes this by producing a flash-light or sound and performs a weed killing action.

- **Free style**

For the first time, the free style contest is no longer a separate contest but flow in the total results. The robots should perform a task that is on the one hand funny and has on the other hand agricultural relevance.

The money spent for the robot is also a criterion in the overall standings. The cheaper the system is the more points are assigned.

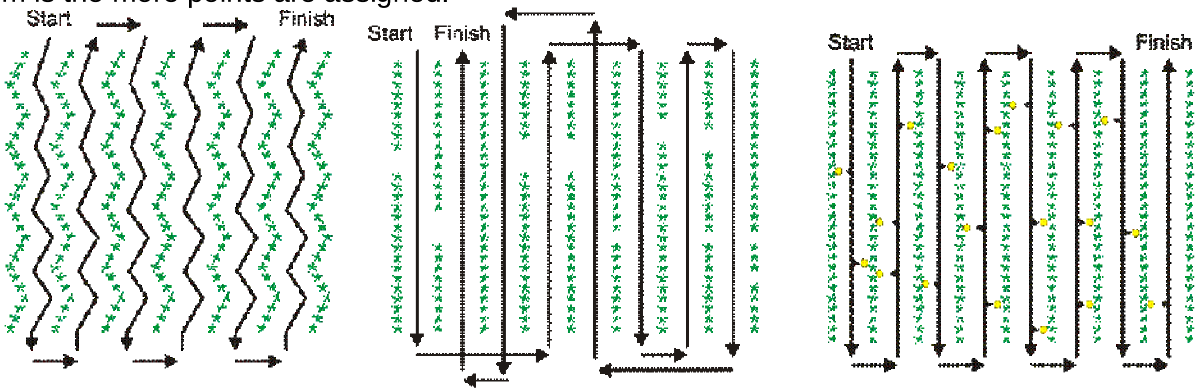


Fig. 2: Course of the event tasks curved rows, predefined robust navigation, weed detection (source www.fieldrobot.nl)

2 Concept

The concept of the new autonomous vehicle “Amaizeing“ is based on the developments of robots like “Eye-Maize” (Eye Maize 2004), “optoMAIZER” (optoMAIZER 2005) and “Maizerati” (Maizerati 2006) from the University of Applied Sciences Osnabrück which started at former Field Robot Events. The decision for the approved TXT-1 Tamiya model is based on the good experiences with the robot Maizerati.

To eliminate the distance measurement faults under unsteady field conditions a slipless distance measurement system was implemented. The approved sensor system from Maizerati was taken over with further development in the software. To increase the flexibility and scalability the CAN-Bus oriented system was extended by small microcontroller systems (SPC) around the robot which are responsible for preprocessing the sensor information. To realise the weed detection and to relieve the Phytec microcontroller, a separate microcontroller system was implemented and also connected with the whole system via CAN-Bus.

The high power consumption and the application area of the system raised the need of a reliable and powerful supply with separated energy sources for motors and electronics. Additionally it was important to attend EMC influence of the system to increase the reliability. The higher complexity of the system led us to build a new Graphical User Interface with a wireless LAN connection to the Phytec microcontroller. Figure 3 shows the whole system overview of the robot.

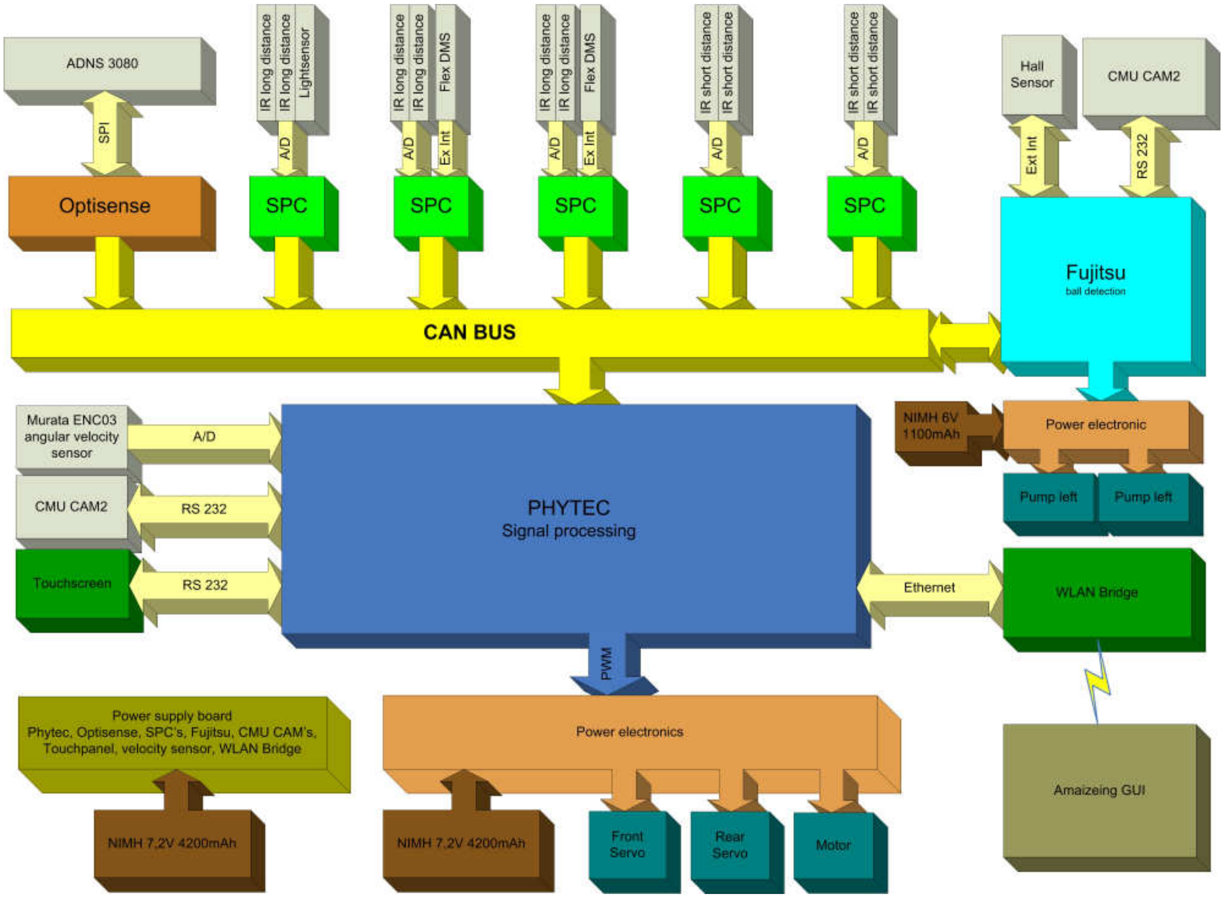


Fig. 3: System overview

3 Mechanics and Electronics

3.1 Mechanics

The base of our robot consists of a modified Tamiya TXT-1 RC-Monster truck model. This unit was also used by the previous model “Maizerati”. The model approved at the last Field Robot event in speed, agility and capacity.

Based on the CAD model, shown in figure 4, a new platform was designed. At this stage it was important to regard the intended structure of the control units and the concluding space consumption. Furthermore the mounts for the sensors, a weed killing module, a distance measurement module and especially two cameras had to be realized.

These requirements led us to use an approved combination of aluminium, high grade steel and Plexiglas. The Plexiglas provides the opportunity for interested people or spectators to have a look inside our robot. The high grade steel and the aluminium case design also give a high degree of security for the electronic components and handling. The construction of the Amaizeing platform and the Plexiglas were designed with the CAD software CATIA. In the original model the Tamiya TXT-1 RC-Monster truck contains suspensions. The platform of the Amaizeing robot had to bear such a weight, as the result of this the dismount of the suspension was necessary to keep good driving characteristics. So this design of aluminium and high grade steel was very important to protect the hardware against rough field conditions.

The base is equipped with two RS-540 electric engines which are connected to the wheels via a four-step gearbox. The all-wheel steering is driven by two servo motors. This combination allows controlling the front and rear axle individually.

The model also includes a no limit power electronic drive controller which is connected to the two engines energised with a seven cell battery stack. This power module can be linked to a microcontroller via PWM channels for speed control. The two servo motors for the steering are also controlled by a microcontroller via PWM and have their own power supply with a fixed voltage regulator.

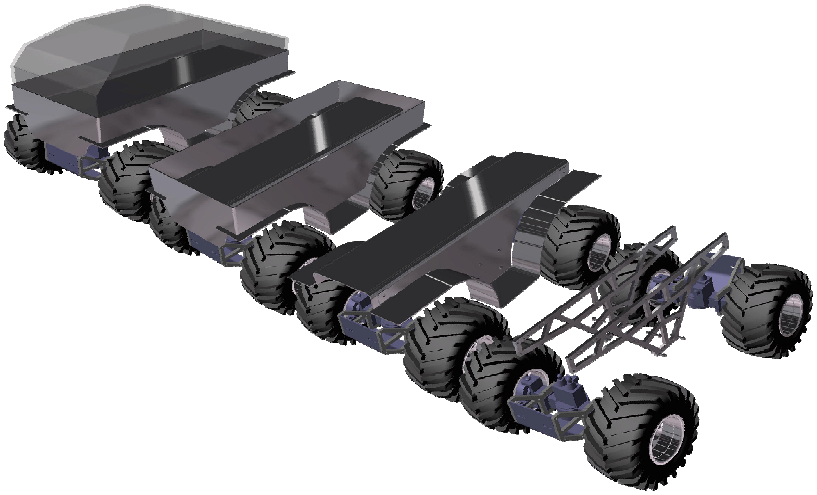


Fig. 4: CAD Model

3.2 Microcontrollers

As mentioned before the system contains a combination of different microcontrollers. One low-cost microcontroller is not capable to handle all needed functionalities. To remain with the existing algorithms for the Infineon C167CS and to keep a low cost system it was decentralised. The biggest advantage of using a combination of low cost microcontrollers is that they are able to work in parallel and only the important information's are exchanged over the CAN-Bus System with high transfer rates.

3.2.1 phyCore 167HSE

The Amaizeing contains the Phytec development board "phyCore-167 HSE" equipped with an Infineon C167CS microcontroller. This is responsible for the main signal processing in the System. Furthermore it processes the information from the CMUcamII, the Touch Display and it also does the A/D conversion for the gyroscope. The microcontroller collects most of the preprocessed sensor information on the CAN-Bus. Based on all these data, an algorithm computes a decision regarding course and speed variation. Another important task for the microcontroller is to handle the communication over a wireless LAN interface to the GUI. The additional circuit board contains the power-electronics for the flash-light, horn and the status LED's. Furthermore an optocoupler was implemented to realize the division between the different energy supply circuits.

3.2.2 Glyn evaluation board

The Glyn evaluation board has a Fujitsu MB90F345CA microcontroller. The main tasks are weed detection, preprocessing the hall sensor information and initiation of the weed killing action. The self developed additional circuit board contains power electronics for the weed killing action and LED's. Furthermore there is a CAN driver to realize the CAN connection and an EEPROM which saves all the parameters for the weed kill. Additionally there are external interrupts, I^2C bus connection and a third serial interface available for further developments.

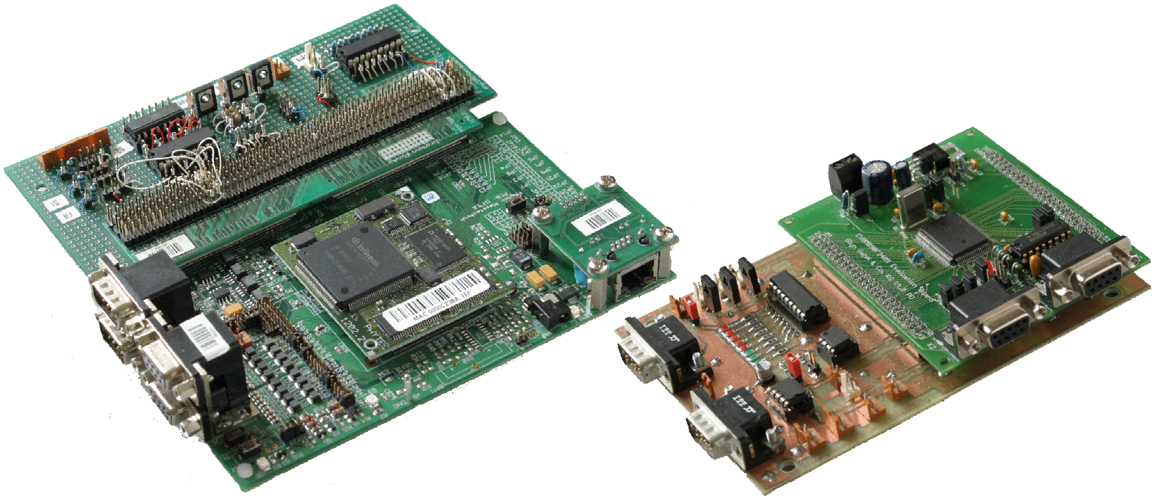


Fig. 5: Phytec and Fujitsu evaluation board with additional circuit board

3.2.3 SPC

With the demand of modularity, and due to the complex wiring of the former field robots, there was the idea of a decentralized sensor read-out and digitalisation. Therefore small modules had to be developed, which were able to connect the different types of sensor output signals. These modules shown in figure 6 are called SPC (Sensor to PIC to CAN). The overall idea was to use this module for all types of sensors. Therefore it had to provide analogue and digital inputs whereas I²C and SPI interfaces. Over this, all sensor data should be available on the CAN bus. So the node needed a CAN interface. As the market gave no cost-efficient module with all the desired interfaces, the decision to build own modules was born. With the help of DIP switches one can configure a module ID, which affects the sent out CAN message identifiers. By this, more than one module could be connected to a CAN network. Over this, the configuration via switches provided the possibility to use the identical software on all modules. The controller decision led to the PIC 18F258, giving the ability of easy In-Circuit programming at moderate cost with a large variety of interfaces. Different standardised connectors for the CAN bus and power supply were defined to provide mixing up. After layouting the circuit, a printed circuit board was etched within the laboratory. The soldering of all –mostly surface mounted- devices was performed manually. Afterwards the software-implementation started. The result was a 40 by 40 mm double layered module with 4 10-bit analogue inputs with a range from 0 V to 5 V, two external interrupts for fast edge detection of digital signals and an I²C interface. The cost of the parts of the module is about 15 €.

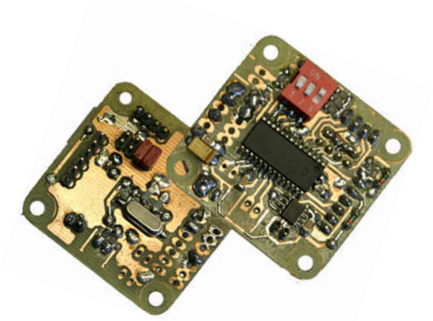


Fig. 6: Both Sides of SPC module

3.3 Sensors

Amaizeing uses different sensors in a sensor fusion system. As most of the sensors are already described in former publications, here is only a short overview over the implemented sensors.

3.3.1 Whisker

The whiskers, which are mounted on the front of the robot, protect it from colliding with maize plants. They consist of very thin strain gauges that are wrapped in isolating tape for mechanical stability. The strain gauges are wired in a full bridge circuit. The output of this bridge, a voltage difference goes to an operational amplifier that outputs a digital signal, which is processed from a SPC module with the help of an external interrupt.

3.3.2 Triangulation sensors

Triangulation sensors are used to measure the distance between the sensor and the maize row. Overall there are 10 triangulation sensors on the Amaizeing. Four long distance sensors with a scope from 200 mm to 1500 mm and six short distances with a scope from 100 mm to 800 mm are mounted. Two sensors look lateral to the driving direction and ensure that the robot drives in the middle of the rows. Four other sensors are in the bar at the front of the robot. The triangulation sensors have a large diffusion in their output values. Therefore, each of these sensors was calibrated with the help of a very accurate sensor. The result was a table to compensate the non-linearity. These tables are stored in the C167 controller, which uses them for linearising.

3.3.3 Gyroscope

A Murata ENC-03 solid state gyroscope was implemented. This micro-electro- mechanical system (MEMS) has a voltage output that corresponds to the turning rate. Integrating this output over time results in the turning angle. The measurement is based on the Coriolis Effect. Because of its high sampling frequency, the gyroscope is directly connected to the C167 controller and is used for measuring the turning angle on the headland turns.

For high precision measurements, the effect of temperature drift and the suppression of noise components can be achieved by a band-pass filter with a lower cut off frequency of about 0.3 kHz and an upper cut off frequency of about 1 kHz. The filter is implemented on the self developed sensor board.

3.3.4 optiSense

optiSense is a slope free optical track measurement system. It sends out CAN messages with the actual speed over ground and the milage. optiSense reaches an accuracy lower than 1%. The functional principle will be presented in further publishments.

3.3.5 CMUcam II

A CMUcam II is used for colour tracking to detect golf balls. This smart camera does not only give out raw pictures but has the ability to do preprocessing within the image. For example in colour tracking mode the camera only outputs the centroid of the colour within the picture and the diameter of the coloured space. All information is provided via a RS-232 connection, which is attached to the Fujitsu board. A second CMUcam II is used for the row navigation.

3.3.6 Light sensor

A light sensor measures the intensity of the sunlight. This information is used to realise an adaptive setting for the CMUcam II. While the light conditions vary, the registered colour of the golf balls, the soil and maize differs very. Especially on cloudy days, there is an ample need of this colour adaptation.

The whiskers, gyroscope, optiSense and light sensors work with self developed circuit boards. All of them were self designed, etched and populated within the laboratory. Figure 7 shows the used sensors and their additional circuits.

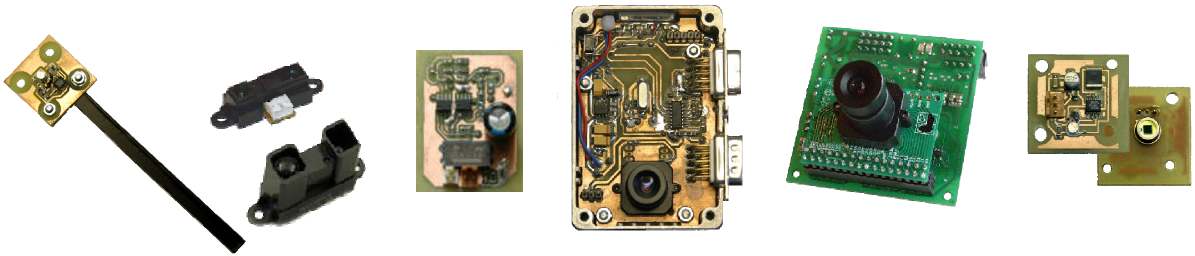


Fig. 7: Sensors from left to right
Whiskers, distance sensors, gyroscope, optiSense, CMUcamII, light sensor

3.4 Actuators

The Amaizeing is equipped with different actuators. To signal the detection of weed a horn and a flash light are used. Furthermore a weed killing action is performed with the device shown in figure 8. It contains two pumps and an own energy supply. The tank is able to contain one litre of fluid. To decrease the use of the fluid and to perform a precise weed kill the device is able to divide both sides. By this the connected microcontroller has to decide if the weed killing action should be performed on the left or on the right. If there is a demand for a weed kill the fluid is pumped from the tank threw a holding valve and injector.

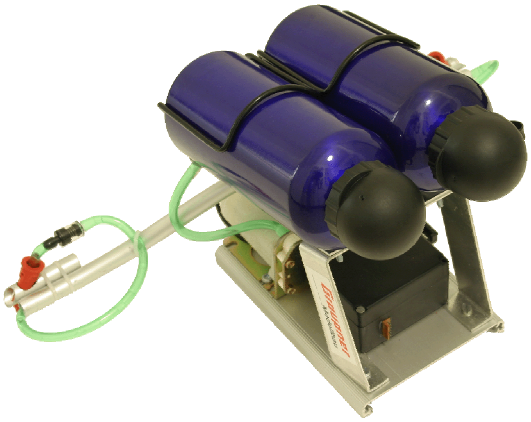


Fig. 8: weed killing device

3.5 Power supply

To realize a robust and reliable system there are high requirements to the power supply. The rough field conditions raise the need of stabile connectors. Also temperature can become a problem when the sun is shining. Another problem is that the signals can be influenced by EMC affects. To ensure this EMC stability two different voltage circuits with separate grounds are needed. The connection between both circuits is provided by an optocoupler. As a result of all these requirements two different power supply boards were built. The board in figure 9 is responsible for the supply of the electronic on top of the robot like the C167 microcontroller, the Glyn evaluation board, or the Wireless LAN Bridge. It consists of three fixed voltage regulators. It is very important to use high cross-section for the ground connections to decrease the EMC affects. A second board was implemented down in the chassis of the robot to realise the supply for the optiSense and the SPC modules. Furthermore there is a switching power supply for the servo motors. For safety reasons the power supply for the driving motor and the servo motors can be shut down by the C167.

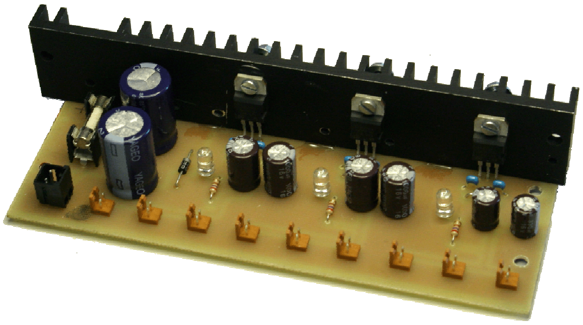


Fig .9: power supply board

3.6 External interfaces

3.6.1 Touch Display

To allow the user to interact very fast and easy with the robot a touch display was implemented (see figure 10). It gives the opportunity to start and stop the different tasks and displays the status of the robot and the result of the weed detection. Furthermore the user is able to control the status of the Wireless LAN connection to the robot swarm. Additionally the predefined path for the second task can be displayed before starting the robot. The touch display is very important for the test phase because of the need to abort and restart the actual task.

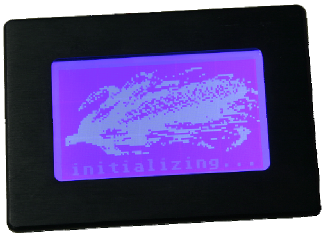


Fig. 10: touch display

3.6.2 WLAN-Bridge

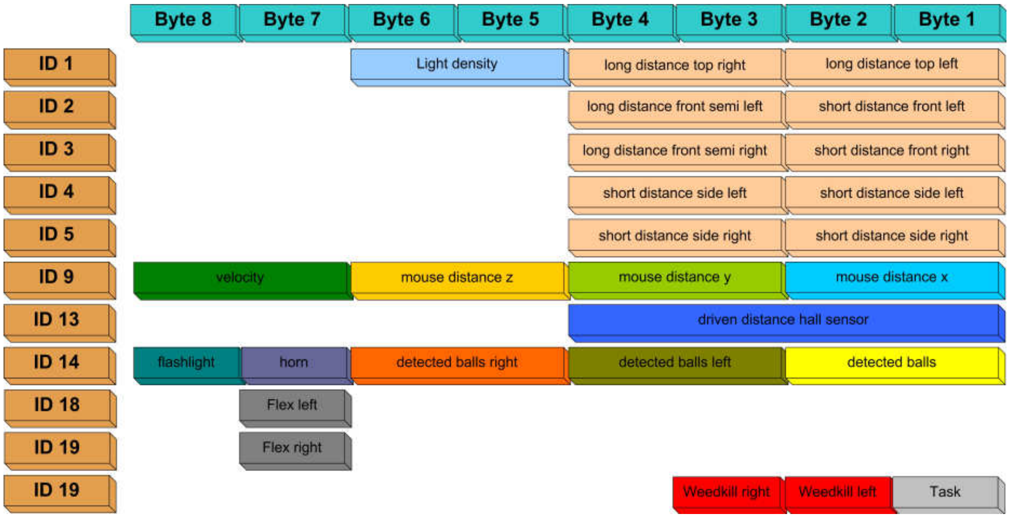
The Amaizeing is equipped with a WLAN Bridge for the connection to an external PC. The use of a graphical user interface during the test phase is very important to find the optimal parameters or to get an overview about the actual sensor data. The requirements for the wireless LAN bridge were 5V voltage supply and the size has to be as small as possible. This led us to a WAP-0003 bridge from Level1. The wireless LAN bridge is wired to the ethernet connection of the Phytec microcontroller board.

4 Software

4.1 CAN Bus

The CAN-Bus is our main connection between the microcontrollers in the system. Data is transmitted by this asynchronous, serial, two wire bus system. It works with difference signals to make the transmission of data very robust. Additionally the data is saved by automatically added redundancy.

At the robot the bus carries most of the pre-processed sensor information. This bus system is very flexible and scalable. New microcontrollers which collect sensor information or control actuators can easily be added to the existing system only by software changes. The biggest advantage is that microcontrollers work independently in the system and collect the needed sensor information parallel. As a result of this the combination of a lot of low cost microcontroller systems affects a nearly real-time operation and a high working performance. Figure 14 shows the existing CAN-Bus identifier specification.

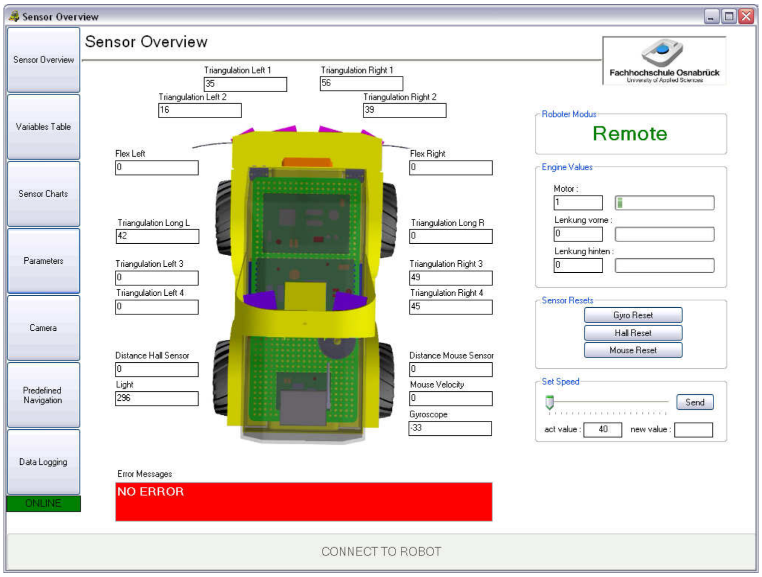


	Byte 8	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1
ID 1			Light density		long distance top right		long distance top left	
ID 2					long distance front semi left		short distance front left	
ID 3					long distance front semi right		short distance front right	
ID 4					short distance side left		short distance side left	
ID 5					short distance side right		short distance side right	
ID 9	velocity		mouse distance z		mouse distance y		mouse distance x	
ID 13					driven distance hall sensor			
ID 14	flashlight	horn	detected balls right		detected balls left		detected balls	
ID 18		Flex left						
ID 19		Flex right						
ID 19						Weedkill right	Weedkill left	Task

Fig. 11: CAN-Bus overview

4.2 External communication channels

To influence all the parameters and to get informed of the actual sensor data a wireless LAN communication can be established to the Phytec microcontroller with a graphical user interface shown in figure 12. By this way the user is able to affect up to 90 parameters to configure the robot. Additionally all the sensor data and other variables of the robots data is logged. As a result of this the user is able to interpret this data after a test drive to improve the system and adapt the algorithms or react on errors. Furthermore he is able to calibrate the course for predefined navigation for task number two. Another interesting feature is the ability to remote control the robot via WLAN link.



Sensor Overview

Triangulation Left 1: 35
Triangulation Right 1: 56
Triangulation Left 2: 16
Triangulation Right 2: 39
Flex Left: 0
Flex Right: 0
Triangulation Long L: 42
Triangulation Long R: 0
Triangulation Left 3: 0
Triangulation Right 3: 49
Triangulation Left 4: 0
Triangulation Right 4: 45
Distance Hall Sensor: 0
Distance Mouse Sensor: 0
Light: 296
Mouse Velocity: 0
Gyroscope: -33

Roboter Modus: Remote

Engine Values
Motor : 1
Lenkung vorne : 0
Lenkung hinten : 0

Set Speed
act value : 40
new value :

Error Messages: NO ERROR

Fig. 12: graphical user interface
with sensor overview

4.3 Turning manoeuvres

Amaizeing has a turning radius of 75 cm, so it is impossible to reach the neighbouring row within one direct turn. To fulfil both requirements, a fast and smooth turn for the first task and a minimised headland width for the second task, two different turning manoeuvres were implemented shown in figure 13. The first is the Omega-plus turn. When the robot wants to turn to the right side, it firstly strikes out to the left side. Afterwards it does a 180° turn to the right. The strike out helps, that Amaizeing's position after the 180° turn is directly in front of the new row. Ther efore the robot stands very early in a parallel position to the row and can change very early to the in-row navigation mode. The disadvantage of the Omega-plus turn is that it needs more space in the headland. With the Z-turn, a second turning manoeuvre is implemented. Turning right again, the robot at first does a 90° right turn. After this, the robot driv es backwards lateral to the rows so that he reaches the row with the following 90° turn. Although there a re two changes in driving direction, which require time for decreasing, stopping and inverse acceleration of the robot, the Z-turn has the advantage of using minimised headland width.

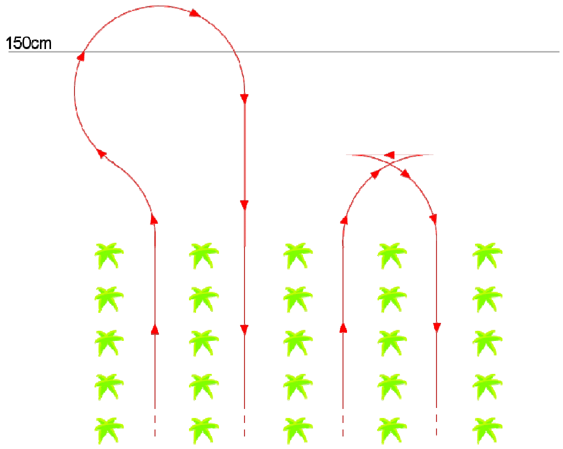


Fig. 13: implemented turning manoeuvres

5 Realisation of Tasks

5.1 Curved Rows

The basic point of the robot is to navigate between rows, whatever the rows are straight or curved. To realise this, the microcontroller has to interpret the data which is given from the sensors and has to estimate the orientation in the row. There are three different sensor-systems with an adjustable priority to give a steering decision. An adjustable priority is very helpful because of the varying surrounding environment. Unlike navigating in really high plants, navigating in plants with only 20cm height has to count on another sensor-system. These mentioned sensor-systems were with a descending order of the priority in small-sized plants the CMU-Cam, the two Sharp sensors for long distance on the top of the Amaizeing and the four Sharp sensors in the front. With the measured distance of the sensors, which target at the plants, it is possible to allocate the robot to one of five zones as it is shown in figure 14. The software algorithms try to control the robot in the centre of the row. Furthermore the robot has to turn at the end of a row in the next one, but where is the end? In contrast to a robot, for a human being it is easy to see the end of the row. There are two terms which must be complied to make a turn. First term which has to be complied to detect a turn is that the robot has to travel a given distance which is measured by the optical way measurement and the second term is that the infrared sensors at the front don't see plants for a couple of milliseconds. If both terms apply, the robot is turning with the z-turn in the neighbouring row.

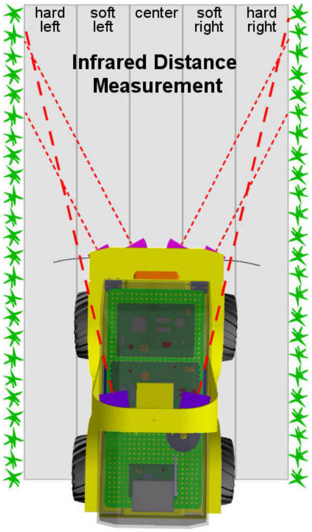
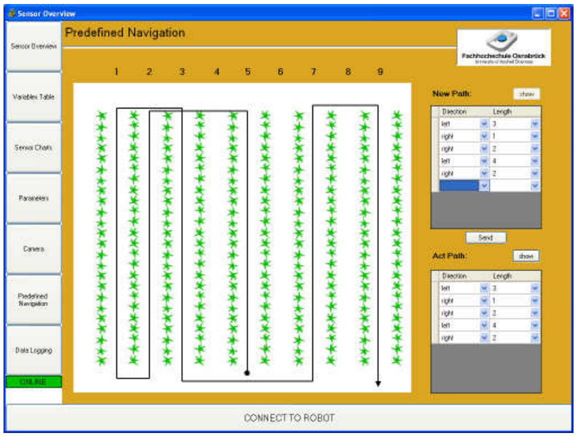


Fig. 14: usage of distance measurement

5.2 Predefined robust navigation

The navigation strategy is the same like the strategy in the first task. For the robust navigation it must be possible to navigate a small distance without any plants at both sides. This is given because of the foresighted sensors (CMU-Cam, sharp long distance sensors) and because of the slipless way measurement – the robot doesn't detect a turn in the middle of the row. Furthermore this task desires that the robot is navigating a predefined way. To allow this it must be possible to turn in each row, whatever it is the first one or the fourth one. Those will be enabled by the reliable distance measurement. To turn in the neighbouring row the Amaizeing uses the mentioned z-turn, because there is only a headland of about 1,50m. A turn in another row than in the next proceeds as follows. After a right/left turn of 90°the robot will go a predefined distance, and then the robot will drive again a 90°turn in the same direction so that he is in the requested row. You can implement the actual way the robot has to travel through in the programmed GUI shown in figure 15.



Predefined Navigation

New Path:

Direction	Length
left	3
right	1
right	2
left	4
right	2

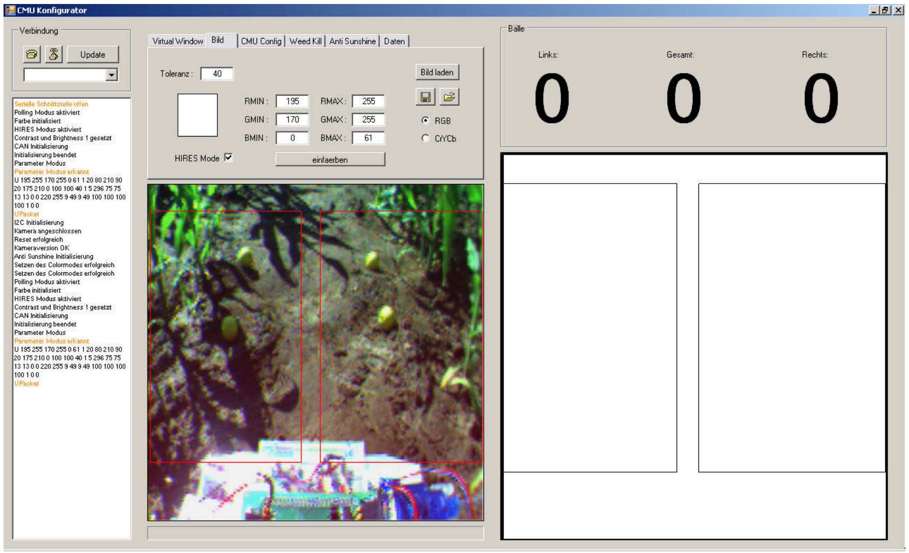
Act Path:

Direction	Length
left	3
right	1
right	2
left	4
right	2

Fig. 15: GUI interface to
configure predefined path

5.3 Weed detection

To realise the weed detection a CMUcamII is used. To do image processing the camera has a frame buffer and a small microcontroller. The resulting preprocessed information is then sent over a serial connection to the Fujitsu microcontroller who does the final golf ball detection. To get the needed information from the camera the used settings are very important to realise a reliable detection. Additionally the light conditions massively influence the parameters. Because of this a graphical user interface (see figure 16) was built to affect all these settings only a few minutes before the event starts. Furthermore the interface was very helpful to develop and debug the detection algorithm because the information from the camera can be logged and displayed. After configuring the parameters they are saved into a connected EEPROM on the additional circuit board to ensure that the parameters are still the same after restarting the whole system.



```
Serielle Schnittstelle offen
Polling Modus aktiviert
Farbe initialisiert
HIRES Modus aktiviert
Contrast und Brightness 1 gesetzt
CAN Initialisierung
Initialisierung beendet
Parameter Modus
Parameter Modus erkannt
U 195 255 170 255 0 61 1 20 80 210 90
20 175 210 0 100 100 40 1 5 296 75 75
13 13 0 0 220 255 9 49 9 49 100 100 100
100 1 0 0
UPacket
I2C Initialisierung
Kamera angeschlossen
Reset erfolgreich
Kameraversion OK
Anti Sunshine Initialisierung
Setzen des Colormodes erfolgreich
Setzen des Colormodes erfolgreich
Polling Modus aktiviert
Farbe initialisiert
HIRES Modus aktiviert
Contrast und Brightness 1 gesetzt
CAN Initialisierung
Initialisierung beendet
Parameter Modus
Parameter Modus erkannt
U 195 255 170 255 0 61 1 20 80 210 90
20 175 210 0 100 100 40 1 5 296 75 75
13 13 0 0 220 255 9 49 9 49 100 100 100
100 1 0 0
UPacket
```

Toleranz : 40

RMIN : 195 RMAX : 255
GMIN : 170 GMAX : 255
BMIN : 0 BMAX : 61

HIRES Mode ☑

● RGB
○ CrYCb

Links: 0 Gesamt: 0 Rechts: 0

Fig. 16: GUI interface to
configure weed kill

The Amaizeing detects the weed by controlling the differences of the colour tracking information in the predefined virtual windows. The algorithm has to differ between golf balls which are already detected before and which appeared in the field of view. This can only be realised by also attending the past sensor information. Furthermore the algorithm has to ensure that noise doesn’t cause a detection. Another problem was to redetect balls which are covered from plant for a short time. If there is a detection of a new golf ball the weed killing action is performed after a calculated distance for a short time. Additionally the information is send over the CAN-Bus to give the detection signal with the horn and flash-light. The detection information is also displayed on the touch display and the GUI which is connected via wireless LAN.

5.4 Freestyle (Swarming)

At the freestyle event the idea behind swarming was presented. Swarming is an increasing topic when it comes to the agricultural use of robots under field conditions. It is nearly impossible to build one robot that is able to do the whole work on the field. A lot of different sensors and tools are needed and the field size is too big to do all the work with one robot. To solve this problem a swarm which consists of many robots for the different tasks have to be build. The idea is to let robots work like the way bees work together in a hive or like other social insects for example ants. Social insects are known to coordinate their work to realise tasks that are beyond the capabilities of a single individual. To coordinate this so called robot swarm a communication (see fig. 17) between the robots is very significant. For example one robot on the field detects weed. This information is send to the robot swarm. Now the robot who is able to do the weed kill and who has the nearest position to the weed is able to drive to the exact position and performs the weed kill. By this way of communication a coordinated work in a swarm on the field is possible.

At the Freestyle event the idea behind the communication in a robot swarm was shown to the audience. The robot Amaizeing tries actively to establish a connection to other accessible robot systems. In this case the robot Maizerati. Afterwards colour tracking was performed at the robot Amaizeing. Colour tracking means that the robot follows a specific colour. The driving information from the robot Amaizeing was transferred to the Maizerati. By this way the robot Maizerati drives around only with the information from the other robot. At the second part of the Freestyle Event the Amaizeing also drives by the information of the colour tracking. The Maizerati was still connected and imitates the movements of the robot Amaizeing. As a result of this both vehicles nearly drive the same way synchronously.

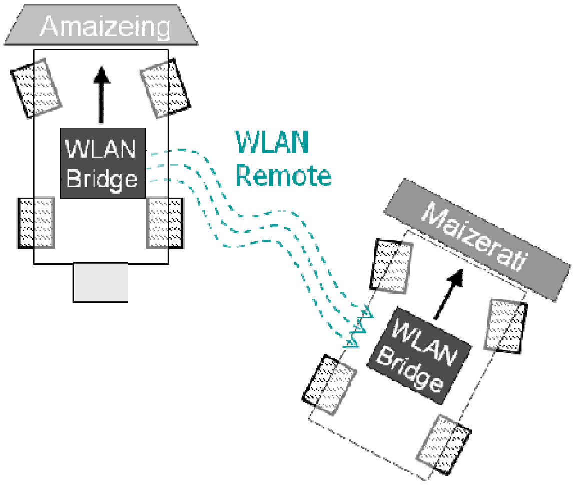


Fig. 17: example of communication in a swarm

6 Results and Conclusion

During the Field Robot Event, the robot performed very well. The development shows that low cost robot systems are able to participate at the field robot event. The robust construction consist the field conditions and no mentionable hardware problems occurred. Due to the high driving speed, some interactions were necessary to prevent the vehicle from destroying maize plants during the first task. The high driving speed was necessary, as the torque of the driving servo motor was not powerful enough so that the speed could have been even slower.

The second task, driving the predefined pattern, went excellent. The slope free track measurement and the robustness of the sensor fusion concept helped working so well.

The third task showed how tight the drive train really was. The weed killing actor enlarged the weight of the robot by approx. 1 kg. As the front axle lost grip on soil, the team enlarged the soil pressure on the front axle with the help of an accumulator. Summarizing, the robot weight increased for about 2.5 kg. To ensure, that the robot detects every golf ball, the driving speed was again decreased. The combination with the extra weight and a small acclivity in the field, led to the fact, that the motor was not powerful enough to drive the robot. It stopped and manual interaction was necessary to push-start it again. Nevertheless, the weed killing action did very well. The freestyle task showed the swarming principle. Also this performance worked very well. In the overall standings the Amaizeing reached an earned second place. In summary one can say, that the drive train was the bottleneck of the robot. But enlarging the power of the motor brings new problems in the drive train. The gearboxes from the modelmaking sector are not solid enough to pass higher power. So one has to enlarge the whole drive train. If the whole drive train has to be changed, there is no longer the advantage of using the platform of the model. A completely constructed solution would be necessary.

The electronic principle of using microcontrollers instead of PCs worked very well. The guarantee of fulfilling answering times of the controllers is a large advantage over PCs, which might delay the tasks for some seconds in extreme situations. Also the modularity and decentralism makes the complex system very flexible. Comparing this robot to the competitors, the computing power should have enough reserve to fulfil the tasks for the next years.

The next large evolution step could be to change to a professional platform using industrial motors and self designed gearboxes for high torque transmission.

References

[Eye Maize 2004] “Field Robot EYE-MAIZE” ; Frank DIEKMANN, Jens FLEISCHHACKER, Johannes HENKEL, Ralph KLOSE, Torsten KÖNIG, Martin MEIER, Nicola MOCCI, Axel MÜHRING, Daniel NEGD, Tobias NOLTE, Evert NORD, Maik SCHOTMANN, Johann SCHULZ (Student project supervised by N.Emeis, A.Linz, A.Ruckelshausen); Field Robot Event 2004, Wageningen / The Netherlands, Proceedings, ISBN 90-6754-818-9, March 2005

[optoMAIZER 2005] “Field Robot optoMAIZER” ; Ralph KLOSE, Martin MEIER (diploma thesis supervised by A. Ruckelshausen, A. Linz); Field Robot Event 2005, Wageningen / The Netherlands, Proceedings, ISBN 90-6754-969-X, November 2005

[Maizerati 2006] "Sensor fusion based navigation of the automous field robot Maizerati" R.Klose, J.Klever, A.Linz, W.Niehaus, A.Ruckelshausen, M:Thiel, M.Urra Saco, K.-U. Wegner, Bornimer Agrartechnische Berichte, Heft 60, 2007, S.56-62.

[Rath, T., Kawolllek, M] „Probleme des Einsatzes von Robotern zur selektiven Ernte von Pflanzen und Pflanzenteilen"; Bornimer Agrartechnische Berichte, Heft 56, 2006, S.67-75.

[Van Henten, E.J., Van 't Slot, D.A., Hol, C.W.J., Van Willigenburg, L.G] "Optimal Design of a Cucumber Harvesting Robot" ; Proceedings of CIGR EurAgEng/VDI-MEG, FAO World Congress 2006 (Bonn, Germany) – Agricultural Engineering for a better World, 2006.

[Mueller,J.; Walther,S.; Boettinger,S.; Ruckelshausen,A.; van Henten,E.J] "Field Robot design contest – a showcase in Agricultural Engineering education", in: Kataoka T, Noguchi N, Murase H, editors. 3rd IFAC International Workshop on Bio-Robotics, Information Technology and Intelligent Control for Bioproduction Systems 9-10, September 2006. Sapporo (Japan); p. 276-281.

Acknowledgement

The whole team would like to thank it's generous sponsors:

AMAZONE-Werke H.Dreyer, Hasbergen-Gaste, Germany
INOEX GmbH, Bad Oeynhausen, Germany
Phytec Technologie Holding AG, Mainz, Germany
Glyn Jones GmbH, Idstein, Germany
Powermagnetshop, MTG Europe e.K., Weilbach, Germany
Graupner GmbH , Kirchheim/Teck, Germany

AMAZONE
PHYTEC
TECHNOLOGIE HOLDING AG
iNOEX THE FUTURE OF EXTRUSION
Magnete MTG Europe Magnet Technology Group
www.powermagnetshop.de
Verschiedene Formen und Größen ab Lager lieferbar!
GLYN
High-Tech Distribution
Graupner
Innovation im Modellbau

Moreover, the team thank Arno Ruckelshausen and Norbert Emeis to make this Amaizeing project possible. Many thanks to the assistants of the different sections especially Ralph Klose and Andreas Linz.

Why did “Cornickel 3” not perform like it was supposed to?

Sebastian Aschoff, Matthias Grimsel, Thomas Haase, Jürgen Lobe, Erik Thiel
TU Dresden, IVMA, Professur für Agrarsystemtechnik, D-01062 Dresden,
Grimsel@Landmaschinen.tu-dresden.de

1 Introduction

Like the name suggests, Cornickel 3 is the third edition of a so far quite successful concept. All that was planned for this year's competition was a design freeze and some minor improvements with respect to stability and robustness. Especially we wanted to overhaul the wiring and some mechanical weaknesses such as the steering and some fittings. Another point was the user friendliness and the service comfort. Unfortunately we seem to have done just one improvement step too much and the competition caught us in the middle of an unstable phase. So our robot did not quite perform like we had expected and we ended up 9th place. Why did this happen?

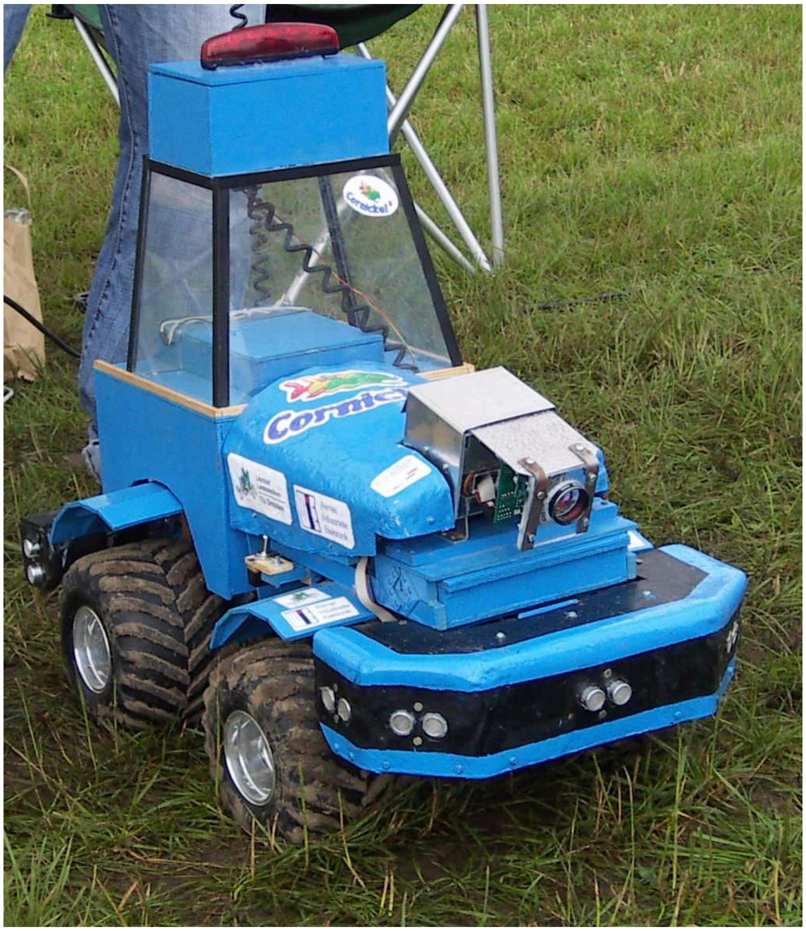


Figure 1: Cornickel 3 waiting for its field test.

2 Conclusions

So rather than giving you all the details of our excellent but non working hardware, we want to share with you the lessons we have learned in order to avoid such “unpleasant surprises”:

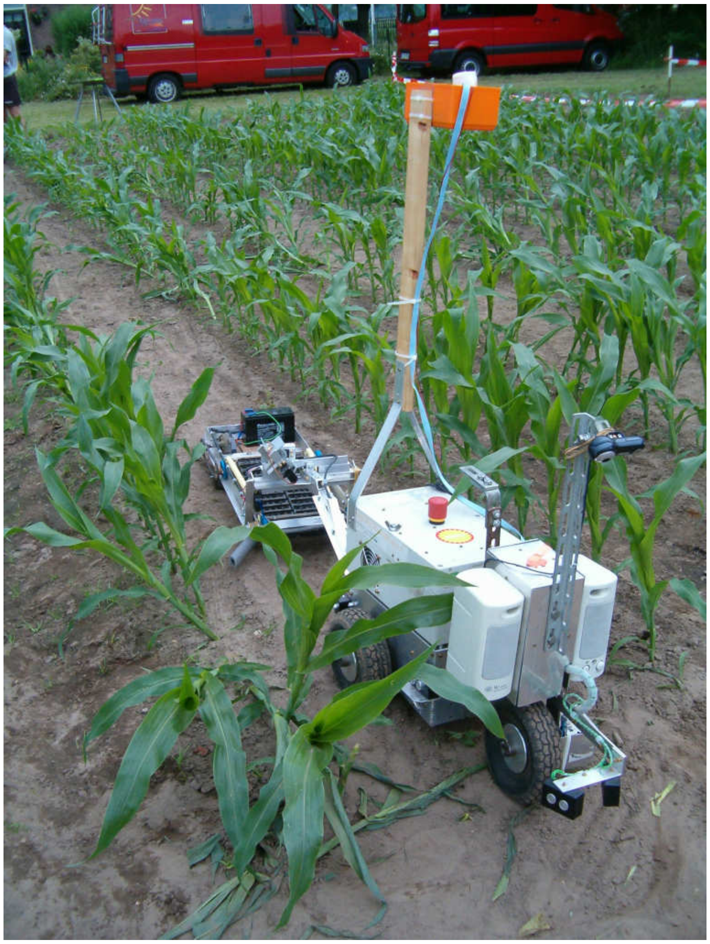
- Spend some time to gather your team. Personal sympathy and creativity may be worth more than just expertise. 5 to 7 people seems like a good starting point. And although democracy is generally a good idea, you need a team leader to coordinate the tasks and to pull some final decisions.
- Don't start working on any details before you have made up your big plan. During the first week or two nobody should use a computer but rather work with pencil and paper and put down the general idea. Set up a schedule with milestones and put it on the wall where everyone can see it.
- Don't start patching last year's software if you are not completely comfortable with it. Put everything in question. Maybe it's better to grab just the idea behind it and then start from scratch.
- Have always at least two people working on the same problem. Four eyes see more than two and there is always somebody around who knows what you are doing.
- Don't wait with the tedious work of documentation until everything else is thought to be ready. You and all the other team members will need the documentation during their current work. It will save you quite a lot of time in debugging.
- Have team meetings in regular intervals. Use the time for brain storming, problem discussions an checking your schedule. Some beer will not necessarily downgrade the creativity.
- Allow enough time for testing the components and the whole system. What works in the lab will rather fail in the field. And even if it does work, there is still Murphy lurking behind the corner. Don't forget the operator training, because under competitive conditions you won't have much time for thinking.

We will try to keep these lessons in mind when we build our next field robot and look forward to see you next year.

Acknowledgments

We would like to thank our sponsors GWT-TUD GmbH in Dresden and Werner Industrielle Elektronik in Kreischa.

Forward Thinking



An autonomous vehicle

Tim Kool, Bastiaan Vroegindeweij, Harmen Wollerich, Theo van der Zwaag
Students Agricultural Engineering, WUR
Contact: Harmen.Wollerich@wur.nl

Abstract

The Field Robot Event is an international competition with field robots between teams of students, institutes or enterprises. The field robots are supposed to be able to navigate autonomously in a maize field.

This article is about the further development of the hard- and software of the Fieldmeister, a field robot made by two different student teams of the Wageningen University for the two foregoing Field Robot Events. The further developed robot has been given the name Forward Thinking.

Several improvements were made, for example the end of row detection, the golf ball detection and the compass correction. Furthermore a new device is developed to pick up the golf balls. This device is also used to pick up eggs from the ground as a freestyle demonstration.

The Forward Thinking has shown that it can navigate almost perfect through maize rows, and can most of the times turn on the headland in the right way. Detection of golf ball can be made better, but is ok. The golf ball picking up didn’t work properly, due to communicating problems inside the robot and difficult soil properties. Picking up eggs can be done smoothly, such that the eggs don’t break. Steering with help of the camera to the eggs must be further developed.

1 Introduction

For the fifth time the Field Robot Event was organized. This year a team of four Bachelor Agrotechnology students of Wageningen University participated and made Forward Thinking. This was done as a learning project, the project started in March 2007. The robot is based on the robot FieldMeister, this was the Agrotechnology student robot of last year. This robot was further developed, because the priority was on the soft ware development. The team was divided in a hardware and a software group. At the Field Robot Event Forward Thinking had to perform four different tasks: 1 navigation in curved rows, 2 follow a pre-programmed pattern, 3 find and destroy yellow balls, these balls represent weed and 4 do a free style, our free style was collecting eggs. For navigation the robot relies on four sensors systems: 1 ultrasonic sensor, 2 electronic compass, 3 pulse counter and 4 a camera. For each task different sensor systems or a combination of systems where used. The Robot used National Instruments LabVIEW 7.1 as programming software. The weed killing action preformed by Forward Thinking is done by picking up the balls by the pick up wagon, this is different from all the other teams. The reason here fore was that it is unique and something different from everyone else. The pick up wagon is a combination of a pulled potato harvester and a combine harvester.

2 Material & Methods

2.1 Hardware

2.1.1 Chassis of the Forward Thinking

The chassis of Forward Thinking is the same as the chassis of the FieldMeister robot of last year. The chassis is also used in previous years: 2003 Agrobot 2, 2004 Challenger, 2005 RowBo and 2006 FieldMeister. By using the same chassis as last year it was possible to concentrate on the things that where important and it was also a lot cheaper. The chassis consist of 3 wheels, two at the back and one front wheel. The Front wheel is the steering wheel. The trike has as the advantage that it is easier to steer then 2 independently driven steering wheels. The robot uses 3 DC motors to drive and one to steer. To protect the engine against high stress forces the axle was not directly mounted on the steering engine, but connected with a second axle. This construction is very strong and allows Forward Thinking steering mechanism to resist high forces from several directions. Both axles are connected by gears, so a decrease in rpm to increase the steering power was possible. In this way a solid driven front wheel with a steering translation of 85 decrees to the left and right was realized.

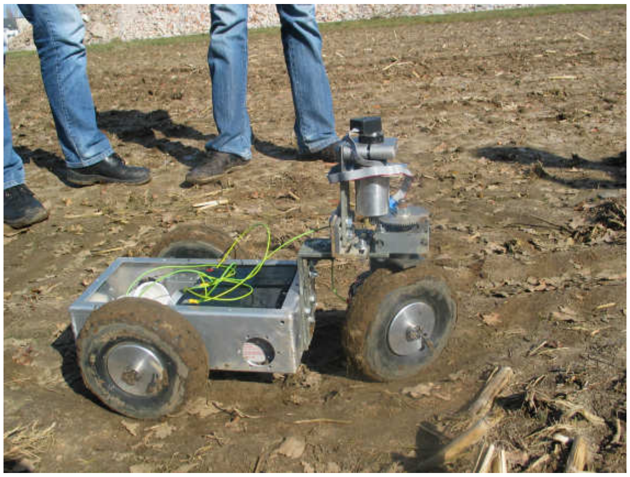


Figure 1: the chassis of Forward Thinking

To connect the pick up wagon to the robot a tow bar was mounted on the backside of the robot.

2.1.2 Chassis of the Pick up wagon

To be able to do an action on the golf balls we made a pick up wagon. This was done because we thought that a spraying mechanism should be done by the other teams, so it’s not that original. First we made a prototype with technical Lego components to see if the idea was possible. We used the tracks of the student robot from a few years ago, to transport the balls to the bin. In front of the tracks we placed rotating flaps which clamped the balls against the tracks. We tested the prototype inside and we were convinced that it should work properly.

After this we made a more full-scale version of the pick up wagon which we used during the contest. The chassis of the pick up wagon consist of aluminium L-profiles which was easy to deal with. The other advantage was that the aluminium has no negative effect on the compass measurements. We used bearings to be able to let the axes of the tracks rotate smoothly. Also the axle of the flaps on the front has a bearing on each side. The tracks were driven by a 24V dc engine, which was found in the barn of one of the team members. We used a 12V battery for the power, because with 24V the tracks would rotate too fast. The power of the engine was brought to the tracks with a small chain to have a solid rotation. The flaps on the front were driven by three elastic belts, which were crossed to make it possible to let the axle rotate in the opposite direction. Below the wagon we placed two small wheels which we found also in the barn of the group member.

On the front we placed also placed plastic arm, which had to collect the balls from near the rows and let them roll before the pick up device. We made it from PVC pipes whit a soft coat around it, which had to protect the plants from destructing.

2.1.3 Steering mechanism

The golf balls were placed 5 to 8 centimetres from the centre of the rows. The pick up wagon is about 35 cm wide so we had to move the wagon aside to be able to pick up the golf balls. We used a small servomotor to let the towing bar of the wagon swing to the left and right side of the robot. We saw that the servo was not strong enough to do the job. Therefore we manipulated the servo in such a way that it worked well. First we placed a gear to have more power on the second axle which had to move the towing bar. Then we took the potentiometer out of the servo and placed it on the second axle to be able to control the quantity of steering. We had also to remove a small nut out of the servo gear, which protected the servo to turn more than 180 degrees. Now the servo could turn more than 180 degrees but was still able to control the second axle to turn 180 agrees.

The pick up device of the wagon was also steered by a servo. We placed it at the front and with a small wire we could lift the pick up device. We also placed a switch to stop the 24V motor when the pick up device was lifted. In that way we showed that we only made an action when a golf ball was detected.

2.1.4 RoboteQ AX3500

The wheels of the robot are controlled by RoboteQ’s AX3500. It is a microcomputer, which is able to control two different DC engines. The driving engines are connected in series on one channel and the steering engine was put on the other one. The board can deliver up to 60 Amps on each channel at up to 40V. For feedback on the performance of the engines there is a step counter connected on the driving engines on the front and the right wheel. This counter is also used to know the exact position of the robot on the headland. On the steering axle a potentiometer was attached to control the steering angle. This information is used by the AX3500 to control the engines. This is done to keep this at a low place in the software and not in the computer itself (high), which could take some seconds and would be far too slow then.

The size of the board is 4.2” x 6.75”. We can use an R/C radio to control it but we used the serial port of our computer to communicate with the AX3500.

Other places where the AX3500 is used is in Underwater Remote Operated Vehicles (ROVs), Automated Guided Vehicles (AGV), mobile robots for exploration, hazardous material handling, military and surveillance applications (RoboteQ 2005).

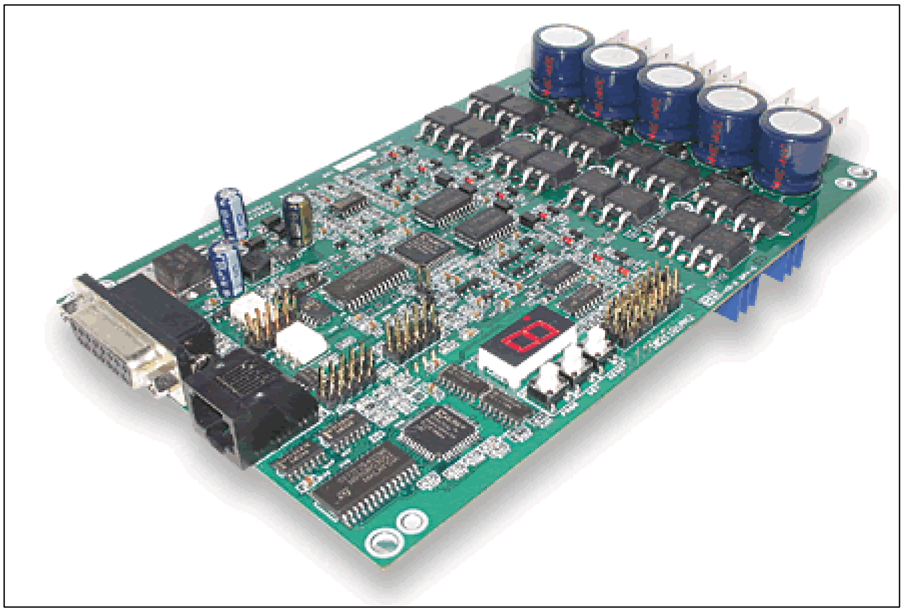


Figure 2: RoboteQ's AX3500 motor controller (RoboteQ 2005)

2.1.5 Basic Atom

To steer the two servomotors we used the basic atom. On this board there are four connections which can be used for servo control. When the basic atom was programmed in the right version for the several tasks we were able to make the pick up wagon 'plug and play', because the board only send information and didn't receive information back. The advantage of this is that it is not necessary to restart the program or the computer. We thought it is just like the 'ISO-bus' connections on new tractors which is also plug and play.

2.1.6 Sensors

Table 1: sensors used by Forward Thinking

Sensor	**Type**
Ultrasonic sensor	SRF08
Camera	Logitech QuickCam Fusion
Electronic compass	HMR3200
Pulse counters	CNY 37

For navigation the robot relies on different sensors. The in-row navigation is done by six ultrasonic sensors. These sensors are evenly distributed on both sides of Forward Thinking. The type of ultrasonic sensor used is a very good and accurate active sensor (max 3 cm deviation). The sensors where told to look 60 cm ahead, but can look up to 6 meters when told so. The sensor works with sonar at a frequency of 40 kHz. The sensor works as follows, it sends a 40 kHz pulse out and the receiving part of the sensor detects pulses coming back which are reflected by objects. Based on the time which it takes to receive the pulse back the distance can be estimated. All the sensors are connected to an I^2C serial bus. This is done to improve the communication with the sensors. The first output fires four sensors, this are the front and back sensors. After 6.5 ms the process of sending out a pulse and receiving it back should be complete, then the middle sensors where executed. This was programmed by the team of last year, the team if this year did not change this because it works well. However it should be possible to execute all six sensors at the same time. The process of sending and receiving the pulse back is executed 14 times a second. The SRF08 has a wide beam width of almost 40 degrees, this is an advantage because the risk of missing a plant is very small. The front two sensors where placed on the steering wheel, this means that the position of these sensors relative to the robot changes. For this software was written so that for every measurement the exact position of the sensor was known.

The camera is used for the end of row detection and for the weed detection. The camera is mounted above the robot this is done to get a “Top down” view, this is necessary for the end of row detection. When the front ultrasonic sensors see no objects in their measuring range the camera is turned on and takes a picture. The pictures is analyzed on the percentage green in the picture, and when this is below a minimum value the robot is assumed to be on the headland and starts to make a turn. The camera is also used for the weed detection, this is done by looking for the colour yellow in the pictures.

The electronic compass is used for turning on the headland. When the camera gives the signal that the headland is reached, the compass steers the robot in two turns of 90 degrees. When driving in the row the compass collects values and calculates an average of it, because the assumption may be made that on average the robot drives in the middle of the row. This method enables the robot to make two turns of 90 degrees and come straight in the row that the robot needs to come in. This in combination with the pulse counters make it possible to drive in a pattern. This is possible because the navigation course, and distance on the headland are known

2.2 Software

1.1.1 Programming Software

The programming software we used was National Instruments LabView 7.1. We used a standard state machinery for the general control, which included navigation, image processing and user-communication. Furthermore there were different sub-VI’s created for specialized task, like image-processing and calculating driving properties of the robot. Also the Basic Atom Micro IDE was used to create the code for the Basic Atom to read the data from the ultrasound sensors and to control servo’s for the pick-up wagon.

2.2.1 Standard State Machinery

The Standard State Machinery (SSM) was used for the main control. The frontpanel of the SSM contained several tabs for different purposes, like one for error messages, one for the headland turn, one for driving parameters etc. On the backside of the SSM, a sequence was implemented in which 5 loops were running in parallel, each performing different task with different requirements in loop speed and behaviour. This sequence was used to make sure that initializing the program parameters and in the end the closing of different parts was done properly. The five different loops were the following:

1. A loop to read and activate the buttons and led’s on the robot itself, for changing the program in the field, and starting and stopping the driving of the robot.
2. A loop to read the data from the motor controllers, for use later on in the navigation and logging process
3. A loop to let the robot navigate and drive, including also the “End of Row”-detection, the Headland Turning and the driving in the “Egg-detection” modus.

4. A loop for the rest of the camera- and image processing issues
5. A loop to create a Datalog of the most important parameters of the robot.

Within these loops, all processes concerning the function of the robot were handled, and communication between them was done with the help of local and global variables, depending on the need of presence of that information elsewhere in the system.

2.2.2 External user control loop

This loop was used for direct communication with the Button's and LED's on the backside of the robot. These were used to select a driving mode when the SSM was running. Also, starting and stopping of robot was controlled with these buttons. Furthermore, for regular driving the rotation direction (e.g. Left or Right) on the end of the row could be changed with these buttons. The hardware for this loop was directly accessed through the LPT port of the robot.

2.2.3 Getting data from the Motor controllers

Last year, and in the beginning of the development this year, this part was embedded in the different parts of the navigation and driving. However, we decided to bring them together in on location to prevent communication errors. These might happen when to different processes are asking things at the same time from the same component. This didn't solve the problem completely, because there was still some possibility for double communication (since the control of the RoboteQ board was done apart from the reading in different sub-vi's) but the availability of information should be better regulated now. The reading loop first asked from the absolute distance and the steering angle of the controller, and then these were processed to give the angle in degrees and the distance in centimeters. The values are stored in global variables, to let the data be continuous present for other program parts. After this is completed, the real speed of the tires of the robot is asked, and processed to give a speed in centimeters per second. This value was also stored in a global variable.

2.2.4 Navigation loop

This loop is discussed further in the part of "Navigation" and the part of the "Headland Turn".

2.2.5 Using the camera

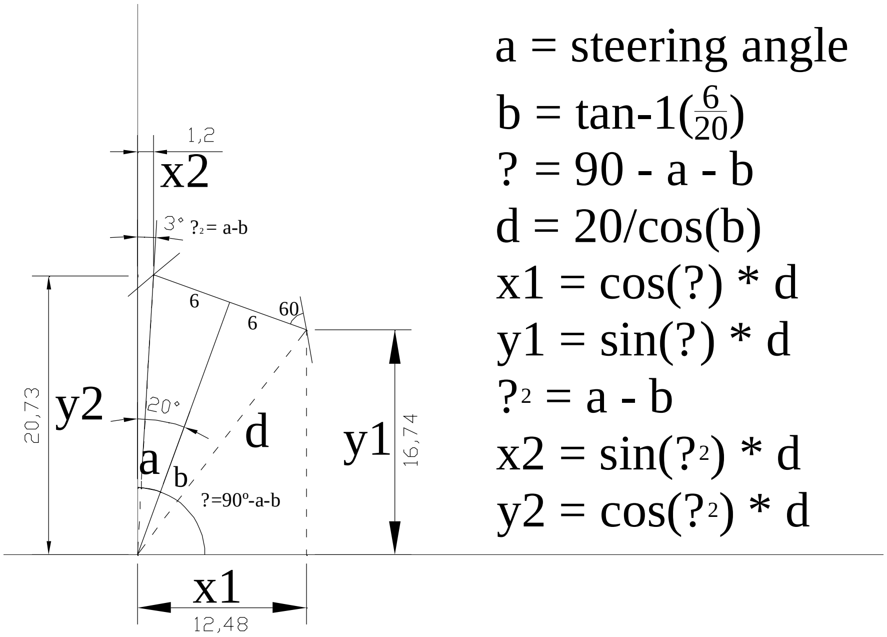
This loop is mainly discussed in the part of the "Golf ball Detection".

2.2.6 Creating a Datalog

A datalogging system was implanted to be able to system what the robot was doing when. To prevent the system of a computational overload, we decided to let logging be done only about 4 times per second, and it could eventually be done on a lower speed. In this logfile, all important parameters, like speed, steering, driving modus, distance, navigation parameters, etc were stored. This file was very helpful for testing, because we were now able to see why the robot did some actions, like was this turn ended based on the compass-value or on the maximum allowed distance? The system created a new file for every run of the program, and the filename of each log included date and starting time. The log itself also contained these, but then more specialized, with times expressed also in seconds, and having a number which defined the cycle number of the logging loop.

2.2.7 Navigation within the Rows: Ultrasonic sensors and navigation

For navigation in the rows, 6 ultrasonic sensors were used. The ultrasonic sensors were connected to the basic atom. The basic atom checks if the value a sensor give is a real value. The basic atom sends the 6 measured distances to the computer. From these 6 distances the computer derives a steering angle, such that the robot moves to the middle of the row. Also a code for detection of the location within the row (plants on the left/right/front/rear) is send. The two sensors on the front wheel of the robot, are moving when the robot steers. Therefore the angle en position of these sensors should also change inside the LabView program. The steering angle is measured and from that angle the position and working angle of the sensors are calculated.



$$a = \text{steering angle}$$
$$b = \tan\text{-}1\left(\frac{6}{20}\right)$$
$$? = 90 - a - b$$
$$d = 20/\cos(b)$$
$$x1 = \cos(?) * d$$
$$y1 = \sin(?) * d$$
$$?_2 = a - b$$
$$x2 = \sin(?_2) * d$$
$$y2 = \cos(?_2) * d$$

Figure 3: Calculation of the position of the ultrasonic sensor on the front wheel

2.2.8 Navigation within the Rows: End of Row Detection

When both ultrasonic sensors on the front of the robot and one sensor in the middle give a non-real value, the basic atom sends a code for 'end of row' to the computer. The computer decides if it is the end of the row or only a gap in the row with help of the camera. The camera takes one picture of which the green is detected. This is done with the excessive green method (Gonzalez et al., 2002). If the area of the green is more then 5%, it is assumed there only a gap in the row. Otherwise the end of the row is reached.

2.2.9 Headland behavior: Calibrating the compass

When the robot was adjusted last year from a skid steer to a front steer type, there was a lot of massive steel added onto the front of the robot. This has a bad influence on the behavior of the compass, resulting in non-roundness and unexpected behavior. Too overcome these problems, the compass was already placed at a higher level (from approximately 50 cm to more than a meter), but still the problem was present. Then we tried to use the build-in calibration mode of the compass, to re-adjust it. Since we have a 3-axis compass, the manual said it was necessary to rotate the compass for 360 degrees with a constant speed, taking a least 1 minute for the calibration, and at the same moment, give it as much roll and pitch as possible. We tried to do this, but it was not very easy, since it had to be mounted on the robot for a correct calibration (redoing this in the correct way should improve the results a lot more). It gave some improvements, but we weren't satisfied with it, so we decided to adjust it in another way. Therefore we took a number of measurements when rotating the robot in steps of 45 degrees, and used them to establish a formula to correct for the disturbances. After implementing this formula, the measurements were quite correct (deviations of max 1.5 degrees).

2.2.10 Headland behavior: Reading the compass

Not only the calibration wasn't correct, also the reading processes of the compass needed some improvements. In the program of last year, just some data acquisition was implemented, but no real communication was present. This year we revised that part of the compass reading program in such a manner that the communication was working properly. First we had to tell the compass that it should stop sending data, and only react if we asked for it. We also decreased the buffer size, to let only the most recent data enter the program, and not the full amount of previously collected data. When we had implemented this, the compass reading was also quite improved, and reactions on turning came through a lot faster.

2.2.11 Headland behavior: Getting an average compass value

In last year's turning procedure, the navigation while turning was completely based on one measurement just before the turn. We decided that this was not reliably enough, since the robot could end up in a shifted and a rotated position compared with the center of the row. Therefore we created a procedure to continuously measure the compass value, and calculate an average of the last 50 measurements. This operation could also be performed by the compass itself, but in that case no direct measurement was available any more. The averaging process created a quite big problem: just averaging the measurements when driving around the north gives a value which heads to the south. This problem was solved by transforming the degree values in radians, and calculating then sine and cosine of the values. After that the sine and cosine were averaged and retransformed into degree values. By this, the value was always the correct value, and didn't suffer any more from the previously mentioned problem.

2.2.12 Creating a new turning procedure to be able to follow a pattern

Since the competition this year also included a task were a given pattern had to be followed and the space on the headland was limited, we decided to adjust the headland turn procedure in such a way that it was nearly universally applicable. This was done by splitting the turn procedure in 5 different parts, which had to be performed each after another. The different parts were:

1 driving out of the row in a straight line. This was done by setting the steer to 0 and driving on for a given time. It would have been better if we did this with steering based on the compass and with driving until a certain distance was passed. This improved method was actually implemented in golf bal-turning (for reasons of making a turn with respect to the size of the wagon), but not yet in the other driving modes.

2. Making the first 90 degree turn. For this, the steering angle was set to such a value that (with two of these turns) it would result in entering the adjacent row. The turning time was limited by a maximal turning distance and a target value of the compass. Since the compass was still not always working correct, the max distance was set very tight.

3. Driving on the headland, to pass by a given number of rows. This was done by steering on the compass until a given distance had been passed. This distance was given by multiplying the inter-row-distance with the number of rows that had to be passed.

4. Making the second 90 degree turn. This was performed similar to the first one.
5. Driving forward to enter the row. This was also done similar to part 1.

2.2.13 Detection of golf balls

For detecting the golf balls in the field the camera is used with a mask, such that the camera 'sees' a rectangle of 50 cm length and full width. The pictures are taken with such an interval that the pictures never overlap. To do this precisely, there is a timed loop with the interval dependent on the driving speed. The main advantage of this method is that you never detect golf balls two times.

To extract the yellow golf balls from the picture, different methods are tested: Extracting different colors, like red, green and blue, and the hue value that expresses the color in a number. All methods were very sensitive for the light intensity, such that you have to change threshold values for every picture. Because in the hue-range green is beside yellow, you cannot make a distinction between green and yellow very well with the hue value. Also to calculate the hue value for each pixel takes a lot of the available computation time of the computer.

By trial and error we found an other method to detect the yellow: Add up red and green (red + green = yellow) (Gonzalez et al., 2002). Substract from that the intensity. By subtracting the intensity, the effect of changing intensity in the picture is reduced. We found that this is the best method we tested.

From the detected golf balls, we wanted to know the exact position. With the position the golf balls can be picked up. To convert the pixel-position of the golf ball to a real-world position a transformation matrix is used. This matrix is derived from a picture with exactly known points. All the found golf balls are stored with an x and y value in an array. The x-value is meant for steering the pick up to the right side. Each time the robot picks a golf ball up, or passes a golf ball, this golf ball is removed from the array. When the end of the row is reached, the array is cleaned up.

2.2.14 Controlling the Wagon

The wagon needed some moveable parts to let it behave like we wanted it to. Since several older robot prototypes were present in our working location, and they have not been used for years, we decided to look if we could use some parts of them. It turned out that in the Spider robot a number of servos were located, and they could be very useful for our wagon. However, to let them work, some kind of communication and controlling for them had to be developed. First we tried to do this with the help of the control boards of the spider, but it turned out that we needed an extra COM-port on our PC for that. The problem here was that there was one left, but after some hard research we concluded that it was broken down. Trying to find another board to replace the existing one also turned out to be useless. This conclusion was made about one week before the event, so time was running out and still a solution had to be found.

After some paperwork, we found out that on the board we used for analyzing the sensor data, also some ports were present to steer up to four servos. Knowing this, we decided to use them for steering the robot. After connecting them and trying some examples on the board, they still didn't work. It then turned out that we also had to give them a power supply, and from that moment on, we were able to let them turn following a preprogrammed sequence.

Though, for a good performance, this turning had to be started and influenced from the main control program of the robot, so some communication between them was also needed. Therefore we adjusted the sensor reading program in such a manner that not only sending information was possible, but receiving could be implemented too. Due too facts as time and complexity, we choose to use the simplest (but less convenient) method, which was just a regular Serial In. This worked as follows: the program listens for a certain prespecified period for incoming signals and than goes on with executing the scripts given. This implies that sending the data should be done at the right moment, or otherwise it won't arrive in the program. After some testing we were able to send one character for control correctly, so all the different statuses were programmed in such a manner that only one character could define which of them had to be executed. When testing this way of working, we found out that the wagon control worked quite well, but there were some side-effects due to the combination of two different processes in one loop. The reading of the ultrasonic sensor was a lot slower now, and also the data send by the sensor was not always read correctly by the main program. Also, since we wanted the sensors to perform as well as possible, we had to decrease the listening and control time for the servo's, having the effect that wagon control worked, but not yet in the best possible way.

All together, this wasn't the most beautiful and secure way of working, but it did the job, and the general idea and principles were visible.

3 Results

This part deals with the results of our robot in different aspect, especially concerning the four tasks of the FRE itself.

3.1 Testing and adjusting the system

We have some general comments on the used processes for testing and adjusting the robot. We already started with some indoor tests in March, and at that moment turned out that the in-row driving was already very well, and therefore didn’t need a lot of attention. Therefore we decided to focus on the turning process and the golf ball-detection, and problems concerned with these task. While working on the row, we continously did some indoor tests to see if improvements were useful or not. Later on, we also went outdoors to check the performance in a simulated and real crop, and to gather data to be able to set the different parameters. This looked quite well, but we also found out that indoor testing is only useful for general checking if things work well and that real testing and adjusting should be performed outside. This mainly because the conditions there can differ very fast and easy, thus creating higher demands for the functioning of the system. Also, simulating the outdoor inside was very hard (even very little things could create major influences on the system, like the wooden plate on which our artificial maize was mounted influenced image processing)

3.2 Task 1: robust navigation and driving in a curved row

The robot performed quite well in this task, especially on the part of navigation in the row itself. This was granted by the jury with a third price for this part. In our opportunity the driving went very well, and the speed might have improved also. The only problem then was that a higher speed would result in a lower accuracy within the row, so we decided not to do this, since high accuracy was also judged by the jury, and the improvements of faster driving would not directly improve the total driven distance. The last part was due to the fact that the number of human corrections might have been higher, and every touch gives a subtraction of 5 meters from the total distance. We only had some small problems on the headlands, due to our steering mechanisms. On one side of the field, the turns of the robot were a bit too short and therefore we had to help him to find the row back. We think this was primarily due to the fact that turning was controlled both by distance and compass value, where the allowed distance was very sharply set, and therefore the first one to end the turning procedure although the next row was not yet reached. This caused two “touches” for our score for this part. On the other headland however, this problem was not present and turning went very well.

3.3 Task 2: Advanced robust navigation while following a predefined path

For the driving part of this task, results were very similar to task one. In turning, there were actually no real problems. Only one time the robot decided to turn when he was in a gap, and this was caused to our interaction, since we set the lower threshold for “end of row” processing a bit higher. The reason for this was the amount of leaves that were still present on the headland, and we felt that this might cause the robot to drive on instead of turning when needed. We think that if this adjustment had not been performed, the robot would have completed this task without big errors. Also, our headland turning mechanism (with compass-based driving) turned out to work quite well, since passing by up to 4 rows was no problem at all. For this task we were also granted with a third price.

3.4 Task 3: Detection of weed (golf balls) and performing a harmful operation

This task did work quite well to our opinion, but we still suffered from some little problems. The first one was on the driving part. We tested our system on the morning before the contest, and we found there a suitable speed for driving that worked well but was not too fast. However, when we implemented this speed at the golf ball task, it turned out be too low for the robot to get the combination rolling. Therefore, we had to readjust it with a driving robot, resulting in three “touches”. After this adjustment driving went better, but was still not perfect. This also was a problem at the headland, and therefore also the turning wasn’t very nice. It had to be as follows: our robot had to drive on for about a meter and start turning then, to be able to return in the second row to the right and not killing any plants (since the total length of the combination was about nearly 2 meters). Another problem we encountered just only after we started was that the cable connecting the wagon with the robot was not plugged in, and therefore the wagon didn’t perform an action at the moments he had to. After this error was repaired, the wagon started indeed to work, and the principle became clear, but the quality of work remained low. Most of the times the golf balls were hit, but rolled away in the wrong direction, so a real picking of a ball was not done. This might also come from the uneven soil, which makes a good pickup harder to do. One thing went very well on this task, and that was the detection of the “Weeds”. All weeds were clearly detected and signalized by a “Tadaa”-sound. Our own-developed image-processing routine therefore has some advantages compared with the use of a HSI or RGB plane. We think that this, together with the very nice idea of a pickup wagon still brought us the fourth place in this part of the competition.

3.5 Task 4 Freestyle

For the last task, we had transformed our robot with pickup wagon to an egg-collector for so-called “ground-eggs”. This task went very well with, with picking up 5 of the 6 eggs present in the field. If you compared it with the amount of work that was used to set up this particular application, it was even better, since the whole software to control the robot for this task was developed in a few hours. The main work here was combining the different parts into one program, since the image-processing was already present with the golf balls and the driving system was adopted from last years “Cornerflag”-task.

The eggs were not really damaged, so you could say that with some modifications, a real use of the system might be in range. The jury also found this idea very nice, and granted us the second price for this task.

3.6 Overall

We think that this contest was very interesting and gave us a nice insight in the world of building a robot. Overall, the performing of the robot during the contest was quite well, and problems were mainly caused (and could be solved) by (adjustments of) the software. This means that the hardware we used this year, is very suitable for use in field robots. The software adjustments were also very depending on the field conditions and can only be improved by doing a lot of field-tests to find the optimal parameters of the different processes.

Our final ranking was also not so very bad, since we ended up on a very nice Third place.

4 Conclusion

During the test the robot did a good job. Driving and navigating through the row was no problem for Forward Thinking. Only turning on the headland went wrong a few times. This is because of the slow reaction of the compass. The robot made a good 180-degree turn, but when he drove a little to the left or right at the end of the row the turn was too sharp or too wide. The second task with the gaps was in the first case no problem but we placed the parameters for the camera a little bit different for the other conditions in the testing field. The problem was that the robot recognised a double gap as a headland so he turned halfway the row.

Picking up golf balls was also not easy, because the balls were situated between the plants instead of 5 to 8 centimetres from the row. The robot recognised all the balls and switched the pick up wagon to the right side and the pick up device down. We think that if the balls where situated more from the row the wagon could pick them up. Also the speed was a little low, which resulted in a standstill in the beginning because the wagon is relatively heavy to pull.

The free style worked well, however this was not tested so the distance from the row with eggs and the robot was too big. In despite of that he picked up all the eggs, which we putted on the ground.

For a next time the compass has to operate better so the headland turn can be made better. Also the way of counting the rows on the headland can be done by the robot instead of driving a specific distance. The camera did a good job, but it is good to make this system more robust so it works for more environments and weather conditions.

We think that the idea of the pick up wagon is a good idea for poultry farmers who have to collect ground eggs. To place this system in such an environment asks for more demands because there are living animals around the robot. The machine has to make save and you have to protect the machine against the animals. Here is a lot of work to do.

5 Recommendations

For further development of the Forward Thinking, some things can be done.

The compass gives sometimes values with a deviation of 20 degrees with the real value. Therefore the compass should be calibrated very well and maybe faraday material can be used to enhance the performance of the compass. Also installing a better compass can help.

The robot has until now no feedback control on driving speed. Because an encoder is installed on the front wheel motor, there can be feedback with the Roboteq controller. The problem with driving without feedback was the speed going to 0 with the high friction of the golf ball wagon on the ground.

For making good turns on the headland, a camera-detection of the rows is essential to get into the right row.

Also an option for the turn on the headland is to drive the whole turn, from the end of the row until the beginning of the next row, with the navigation of the compass. This only works if the compass is much more precise.

For detection of the end of the rows, the robot now uses the amount of green in the picture it takes. Better should be to detect (in some way) if the green parts are really maize plant rows.

Upgrading the robots computer and software is recommended, because the maximum of the CPU capacity is reached. Especially with using vision to detect the golf balls out of the rows, the computer gets too slow.'

The Basic Atom is controlling the servo motors of the golf ball wagon. Because it takes some time for a servo to reach the given position, the Basic Atom must wait some time with each servo controlling action. In this time, the values of the ultrasonic sensors cannot be read, and navigation becomes worse. Therefore the controlling of the servo motors must be done with another device. Another reason to do the servo control not with the Basic Atom, is that the servo motor do not hold their position active, because the position-signal of the Basic Atom is only for the time that it steers the servo to the right position.

Picking up the golf balls from a few centimeters beside the maize row should be done in a better way.

Picking up the golf balls when they lay in a hole in the field should be done in a better way.

The navigation with camera to pick up the eggs for the freestyle is not very advanced. No obstacles can be detected. The camera is only detecting the eggs. When the egg is detected, the robot can only steer with a linear control to the egg. The robot must steer very clever to pick up the egg with the wagon. A new method must be developed for this problem.

Word of thanks

The Forward Thinking team would like to thank some people, without there help the task of building a robot would by very difficult. Firstly we would like to thank the Agrotechnology student teams of previous years, for their help, support and input at difficult times. Secondly we would like to thank Sam Blaauw, Ard Nieuwenhuizen and Kees van Asselt for their help, technical know-how and support in difficult times. Thirdly our thanks goes to the FTE and SCO groups of Wageningen University for given us as students the change to participate in this contest. Fourthly we would like to thank the organization and crew of the Field Robot Event 2007. Last but not least we would like to thank Jan Willem Hofstee for his help on Machine Vision and his support during the whole development process.

References

- Digital image processing, international edition, second edition, Rafael C. Gonzalez, Richard E. Woods, Prentice-Hall, Inc. (2002)
- All kinds of documentation coming with the different devices as Compass, Roboteq controller
- The different LabView help files

Autonomous Field Robot System „HELIOS“

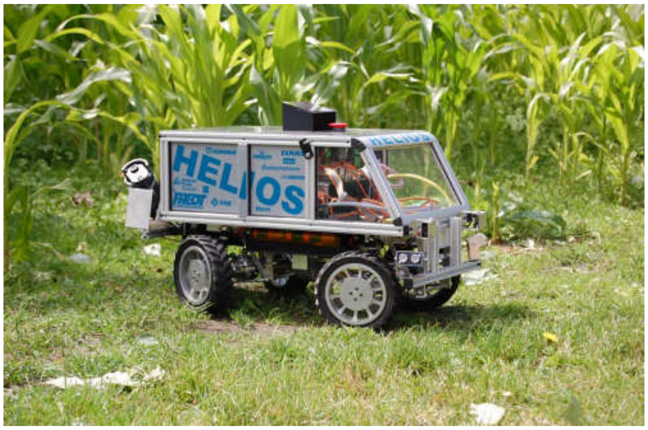


Figure 1: Helios

Team members: Jörn Knaup, Robert Meyer, Michael Meinecke, Markus Robert, Jan Schattenberg, Jörg Schlott

Technical University of Braunschweig, FREDT, Institut für Landmaschinen und Fluidtechnik, Langer Kamp 19a, 38106 Braunschweig
www.fredt.de
info@fredt.de

Abstract

„Gaia“, the first goddess of the Greek mythology, emerged from chaos, was the eponym of our first robot. Her grandchild, the sun god "Helios", who drives at day in a golden car along the sky and returns at night with a golden bowl, should be the godfather for our second robot.

For "Helios" the focus was in the improvement of the chassis. But there was also a large improvement potential in electronics and software. The experiences, which were won in the previous year, incorporated the development.

With the Event 2007 the new chassis could finally prove its fitness. Navigation in the rows was clearly more durable and faster in relation to the previous year and also the turns at the row end were safer.

Keywords: robot, precision farming, autonomous, FREDT

1 Introduction

FREDT… that is the Field Robot Event Design Team of the Technical University of Braunschweig. This working group joined in December 2005 on initiative of the Institute for Agricultural Machinery and Fluid Engineering, in order to develop and build an autonomously navigating vehicle.

With this vehicle the team participates at the annually Field Robot Event and measures with other teams from all over the world.

2 Concept

After the first participation at the Field Robot Event 2006 with our robot named „Gaia", the new experiences were methodically evaluated. At that time the entire sensor technology and logic were composed on basis of a Tamiya Monstertruck (of financial and temporal reasons).

But very soon the weaknesses of this concept showed up. The play in the steering elements and in the drive strand made it extremely difficult to control the robot. The bad material quality led again and again to problems (e.g. drive shaft broken after six hours of continuous operation).

Thus the chassis of the new robot „Helios" should be conceived new. On the basis of a requirement list and some thesis a completely new chassis was sketched in the following weeks. First some sketches on paper, finally 3D-models in CatiaV5.

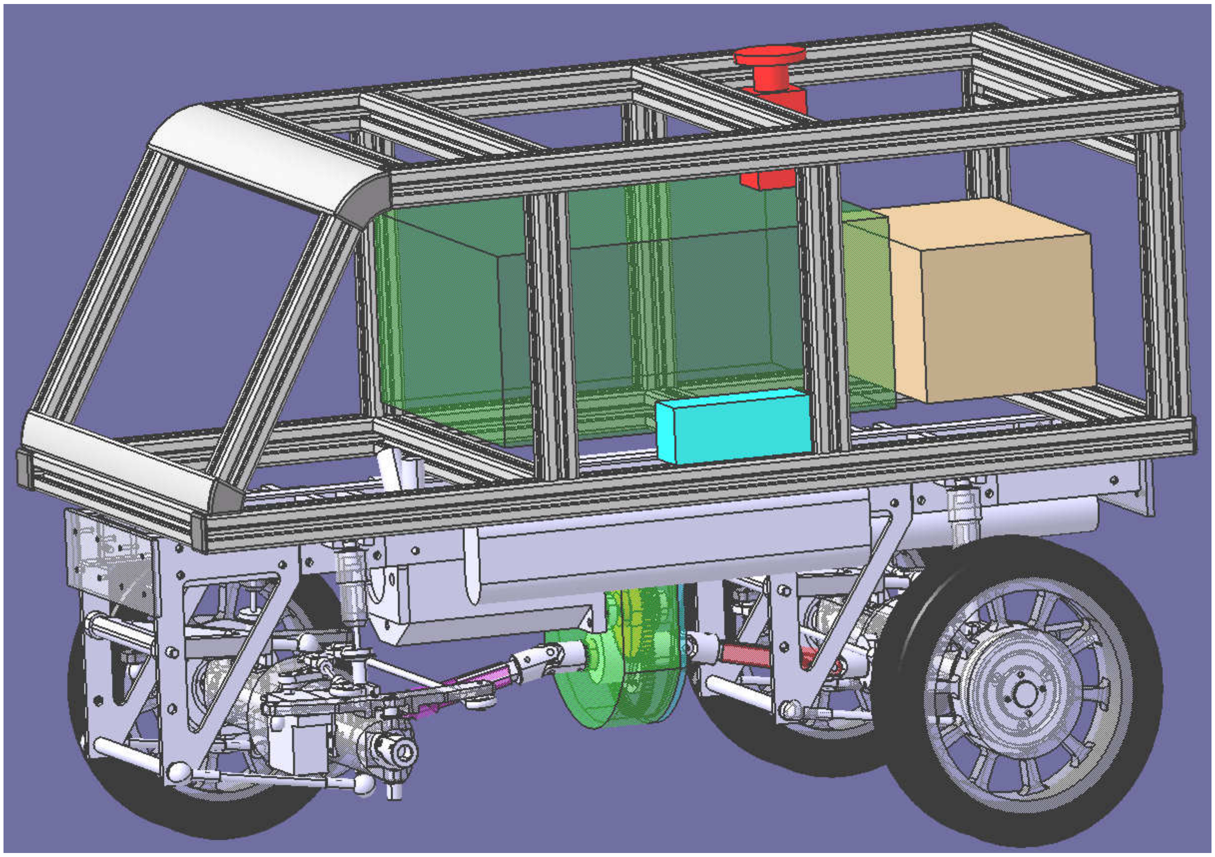


Figure 2: Drawing in Catia

The manufacturing drawings were revised afterwards in co-operation with the lead mechanic and the chassis was manufactured till the end of February.

We also developed a completely new electronic concept.
The idea was a three-divided system. On the high level an ITX computer with sufficient computing power, in order to be able for digital image processing. On the low level a small µController for the actuating elements as well as a high performance µController for the sensor technology. This controller should be able to take over some tasks of the ITX board if necessary.
To arrange the system also lastingly usable, interfaces for RS232, RS422, RS485, CAN, I2C and WLAN had to be planned.

In the previous year a PDA was extremely practical for the man-machine interface (MMI). So we wanted to use a similar system. This time the PDA should have beside a serial also a WLAN as well as a Bluetooth interface.

3 Hardware

3.1 Chassis

With the development of the new chassis the following aspects were in the focus of attention:

- Four-wheel chassis: cheap; sufficient cross-country capability

- Rigid axle: very durably; sufficiently for medium speeds; simple to realize

- All wheel steering (both axles separately steerable): small turning circle (<1.30 m)

- 3 differentials, 1 engine

- Identically constructed axles: cheaper

- v_{max} = 10 km/h => (oil)-absorption; v_{max} from empirical value of the last competition (speed race winner with approx. 8 km/h); higher speeds would need very high engine performances and are more difficult to regulate

- m_{max}: = 20 kg => sufficient engine performance; the weight results from the building method (metal), complex chassis construction, large engine, high accumulator capacity and high additional load (max. 8 kg of sensors, logic, accumulators)

- as less as possible clearance in the steering element => servo directly at the axle; simple tie rod

- Selection of the individual components by availability, sponsoring, price and other criteria

- Construction: only cutting production; to manufacture as simple as possible

All these points had to be considered in the new chassis.

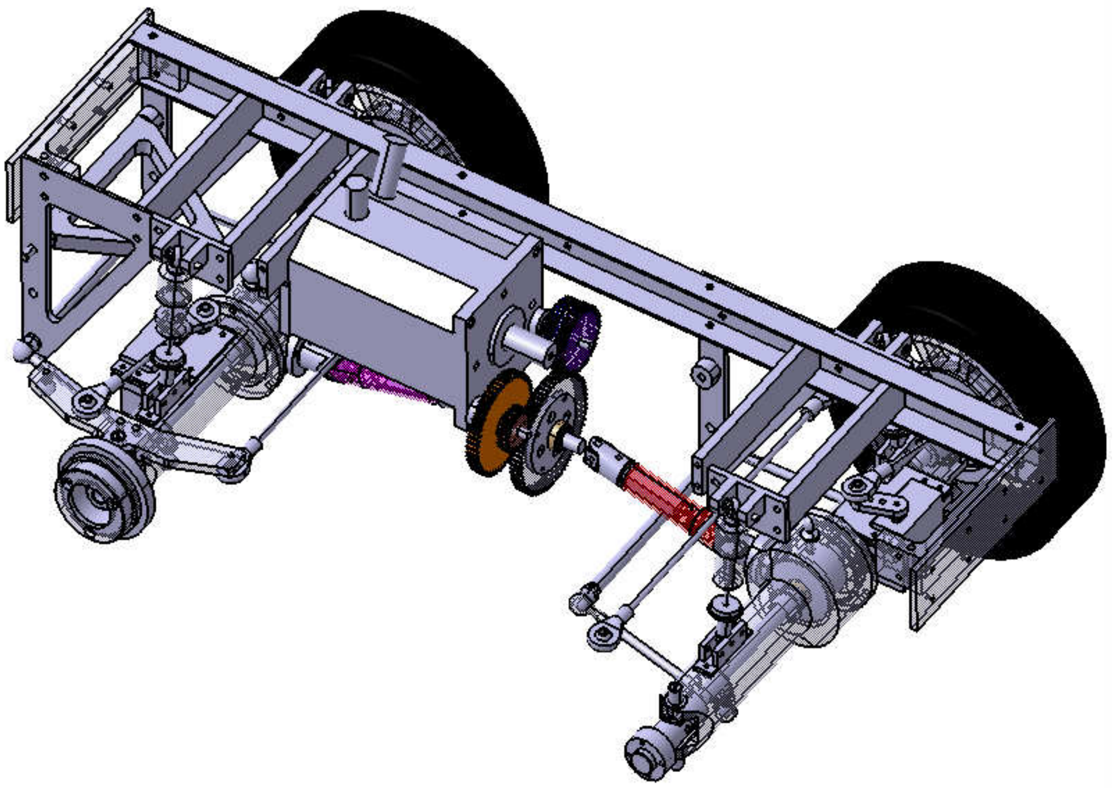


Figure 3: Chassis

3.2 Framework

For a maximum of flexibility during the arrangement of the sensors the FREDT FlexFrame has been developed. A frame construction with aluminium profiles permits here the free assembly to any sensor type or other periphery at the vehicle. Thus it was possible to test and permanently change different sensor concepts within shortest time.

The assembly of hinges at the tail of the structure makes it possible to open the framework, in order to be able to reach all important units fast and uncomplicated.

3.3 Electronics

3.3.1 Current supply

For the supply of electronics four 12V Ni-MH accumulators with a capacity of each 3,6Ah were used. The 12V logic voltage was produced by two parallel switched accumulators and the 24V for the actuating elements from two accumulators switched into row. Both electric circuits were completely galvanically decoupled.

To preserve the accumulator during the longer stationary test and programming phases, it was possible to switch “in situ” to external supply. Just by plugging in a Neutrik patch cord at the tail of the robot.

3.3.2 µController

Because of the experiences from the previous year, the good on-line support and the large existing knowledge in the team we decided again for Atmel controllers.

For the actuating elements we chose an ATmega8. It had to control the engine and made the PWM signals for six servo channels and a stepping motor channel available.

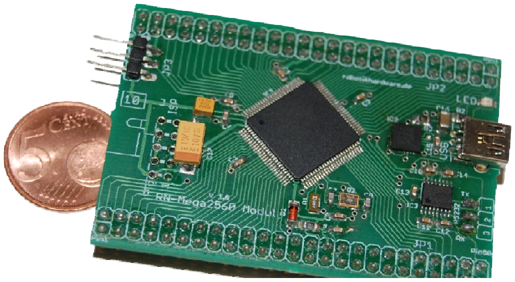


Figure 4: ATmega2560

For the logic an ATmega2560 has been selected, which was already pre-mounted on a small PCB. Calling the sensor values as well as pre-processing the data has been the major tasks of this processor. When it turned out, that the ITX board would not be necessary for the competition, the ATmega2560 finally took over the complete data processing.

3.3.3 Bus concept

For the main bus of the system we used a 400kHz clocked I2C. Also here we could profit from the last years experiences and work thus time-efficiently.

By the grown sensor concept, unfortunately the bus showed fast its fault liability. In case of failure of only one sensor the total failure of the bus was the result.

Therefore other bus conceptions were considered, e.g. a CAN bus, a USB, as well as RS232, RS422 and RS485.

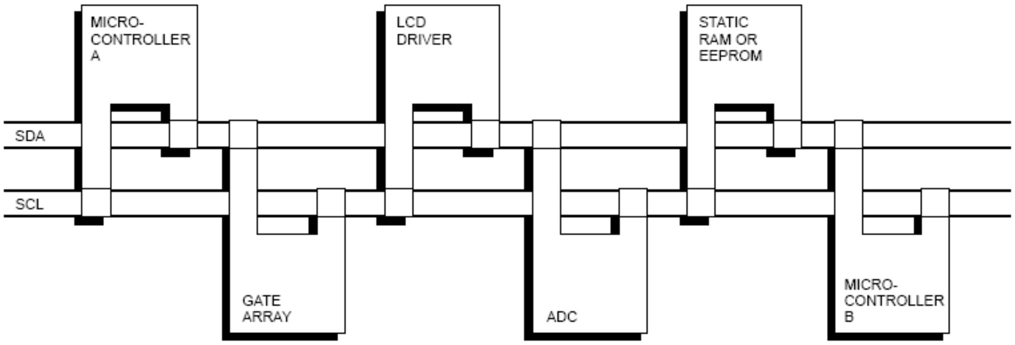


Figure 5: Bus structure

3.3.4 Display

A 20x4 character display from Electronics Assembly has been inserted for the monitoring of different driving conditions. For example the temperature in the electronic box or the residual voltage of the accumulators could be indicated.

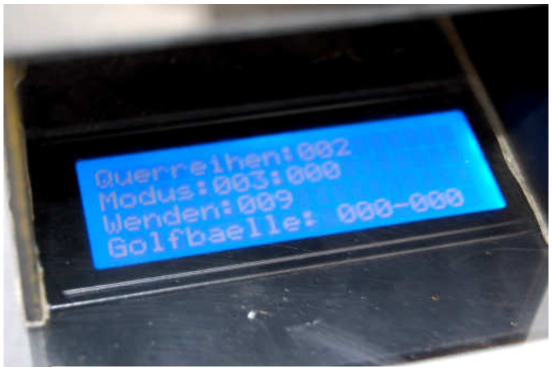


Figure 6: Display

3.4 Sensor technology

3.4.1 Ultrasonic sensors

After a set of tests with various ultrasonic sensors from Devantech we chose the model SRF08. This model had the best fail-safe characteristic and the most exact sensor behaviour.

Fortunately it was possible to merge the components directly into our I2C bus. Due to a relatively strong beam width we installed the sensors „above looking". So unwanted reflections with the soil should be avoided.

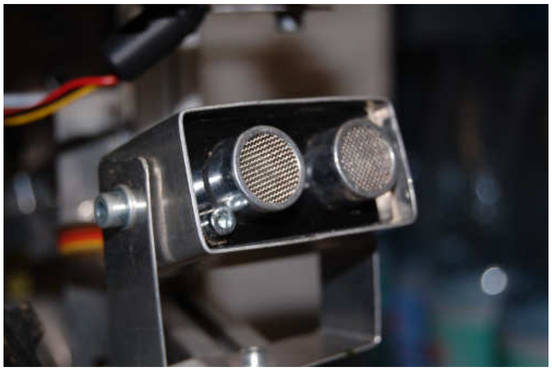


Figure 7: Ultrasonic sensor

3.4.2 Infrared sensors

We used the model GP2Y0D02YK from Sharp. The analog output signal was converted by a ADC and merged into the I2C-Bus. With the infrared sensors a point-exact measuring of the distance was possible which we used in the third task of the competition.

3.4.3 Acceleration sensor

For the measurement of accelerations and vibrations the three-axis sensor MMA7260QT from Freescale has been mounted. However it has not been used during the competition.

3.4.4 Pressure sensor

In order to collect statistic data over the weather, a pressure sensor was planned. It measures the atmospheric pressure, and makes statements about the future weather.

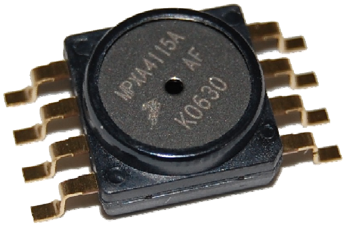


Figure 8: Pressure Sensor

4 Software

4.1 Programming

4.1.1 Controller

Large parts of "Gaias" program code had already been written in C and were adapted for "Helios".

Some of the old libraries, which had at that time already proven as quite reliable, were revised and integrated into the current code, too. The compiler was AVR-GCC.

4.1.2 iPaq

The software for the iPaq has been Visual C# from Microsoft. A graphic user interface served for the expenditure of the current sensor values and for calling the different tasks. Also the WLan communication represented one of the iPaq tasks.

4.2 Navigation

4.2.1 Intra row

For navigation within the rows the ultrasonic modules have been used. With a frequency of approximately 20Hz they determined continuously the distances to the plants on the right and left side. With a simple filter algorithm the data have been weighted and averaged by its topicality. The newest data in the Buffer have been multiplied e.g. by the factor 10, old data only by the factor 2. The arithmetic average finally formed the filtered value. Newer data had thus a stronger influence than older data. „Outliers ", e.g. by hanging leaves, were not so relevant any more.

With further filter algorithms the average width of the row has been determined. If plants were missing on one side, the robot was able to drive the distance to the other side constantly further. If plants were missing on both sides, then simply the original course has been continued.

To detect the missing plants a plausibility check has been inserted. If the measured distance exceeds 50cm, the plants were missing. If on both sides plants were missing for more than approximately one meter, the row end has been recognized and a u-turn initialized.

4.2.2 U-turn

For the u-turn we first wanted to use the compass module CMP03 from Devantech. But the results were very unsuccessful and the robot wasn't able to find the next row. Instead we decided to navigate just by the steering angle and the driven distance (odometry). The parameters have been empirically determined.

In order to keep the slip during the turn small the speed has been reduced. This was maintained up to a meter in the new row. Thus the Intra Row regulation could adjust itself first to the new row, before the original speed was taken up again.

4.2.3 Counting rows

In the second task the infrared sensors at the vehicle front were used for counting the rows.

When the end of the row has been recognized, the robot has already left the row for approximately half a meter. A 90° turn would mean that the robot drives in a large distance to the rows and the infrared sensors would not be able to see the plants.

We solved the problem by driving a 270° curve into the opposite direction. Then „Helios" drove parallel along the corn rows.

When he passed a plant row, a peak in the measured infrared distance showed up. "Helios" counted a passed row. If the correct number of rows has been reached, the robot turned 90° into the row and afte r the short orientation phase of the IntraRow regulation the speed increased again.

4.2.4 Detection

For the detection of the yellow golfballs we used a CMUCAM2. By a small mirror system the rows have been scanned directly with only one camera. The camera searched the yellow colour of the golfballs, so a good contrast between the balls and the soil was very important. Because of bright sunlight it was sometimes very hard to detect the balls correctly.

4.2.5 Killing action

After a golfball has been recognized by the camera at the front of the vehicle, it got a specific number assigned. With the help of the permanently measured distance, conclusions of the current position of each individual golfball were drawn. As soon as the golfball reached the rear end of the vehicle, the killing action took place, connected with a sound effect.

For the killing action we used deodorant bottles. The bottles were fixed with few handles in a mounting plate. Spraying has been controlled with a servo and a simple mechanic construction. Combined with an ignition mechanism also thermal destroying of weeds would be possible.

5 Results and Discussions

At the beginning of our test phases we used a very complex sensor concept. Unfortunately we had to reject the whole concept, because of lack of time and manpower. Finally we adapted the proven system of „Gaia “. After a few test runs the stability of the system was obvious. Principle KISS: Keep it simple, stupid!

During the Event in Wageningen „Helios” showed a great performance. Our tactic to drive with less speed, but more reliably had made itself paid. (Well, we’ve also been very lucky).

Thus we could achieve three first places as well as a second place within the four tasks. Victory was ours!

For the future there is much potential concerning the sensor technology and control strategies. Momentarily new concepts are sketched and realized and will be tested during the next months.

Acknowledgements

Finally FREDT wants to thank our entire sponsors for the financial and material support.

Also special thanks to the “Institut für Landmaschinen und Fluidtechnik” for the excellent assistance, and of course to Prof. Harms, who always believed in us.

Last but not least we thank the organizing team of the Field Robot Event for making all this possible. We spent great days in Wageningen and we hope the public interest for this event will continue to grow in the next years.

References

Knowledge:

[1] Prof. Vörsmann (2005): Regelungstechnik I, lecture notes

[2] U. Tietze, Ch. Schenk (2002): Halbleiterschaltungstechnik

[3] Dubbel (2005) – Taschenbuch des Maschinenbaus

[4] www.datasheetcatalog.com

[5] www.wikipedia.org

[6] MSDN – Microsoft Developers Network

[7] www.roboternetz.de

[8] www.mikrocontroller.net

Shops:

[1] Roboter-Teile, Jörg Pohl: www.roboter-teile.de

[2] Robotik Hardware, Brallsoft: www.robotikhardware.de

[3] Reichelt Elektronik: www.reichelt.de

Sponsors:

[1] Class-Stiftung: www.claas-stiftung.com

[2] Krone Gruppe: www.krone.de

[3] Dunkermotoren: www.dunkermotoren.de

[4] item international: www.item.info

[5] FAG Wälzlager: www.fag.de

[6] mbo: www.mbo-oswald.de

[7] Mädler, Antriebselemente & Normteile: www.maedler.de

[8] Modellbauzentrum Ilsede:www.modellbauzentrum-ilsede.de

Hielke is a Bayesian, balancing field robot

Frits K. van Evert[1], E.J.J. (Arjan) Lamaker[2], Gerrit Polder[3], Arjan de Jong[4], Gerie W.A.M. van der Heijden[1], Eltje Groendijk[1], Lambertus A.P. Lotz[1], Ton van der Zalm[1]

[1]Plant Research International, PO Box 16, 6700 Wageningen, The Netherlands
[2]WUR Facility Services, Wageningen, The Netherlands
[3]Biometris, PO Box 100, 6700 AC Wageningen, The Netherlands
[4]Alterra, PO Box 47, 6700 AA Wageningen, The Netherlands

Abstract

We view the Field Robot Event as an opportunity to have fun building a robot with friends and family while at the same time doing something new and potentially useful. For this year's Event, we decided to have fun by building a two-wheeled balancing robot, and to do something useful by working on probabilistic localization. We constructed a platform of which the main components were two motors taken from power drills, bicycle wheels, a microcontroller and an inclinometer. A simple PD controller sufficed to implement balancing. We added to the platform a webcam and a PC to perform vision-based localization. We implemented a particle filter for localization within the rows and another one for localization with respect to the end of the rows. When travelling within the rows, we segment the image using excessive green and a threshold. We then apply the Hough transform to find crop rows and calculate a measure of the probability distribution of the robot's position. When approaching the end of the rows, we calculate the probability distribution of the robot's distance from the end of the rows by considering the number of green pixels in columns perpendicular to the direction of travel. Time constraints prevented us from implementing vision-based localization on the headland, thus we had to resort to dead reckoning. The robot was able to balance well. The particle filters performed well and turned out to be a major improvement over our earlier localization methods. Unfortunately, poor tuning and software bugs resulted in disappointing performance in detecting end-of-row and on the headland. Passive stability of the robot was insufficient. In spite of these problems, we feel that we now have the knowledge to build a robot capable of running a perfect course in the 2008 Field Robot Event.

1 Introduction

During the plenary session on the last day of the 2007 Field Robot Event it emerged that a number of participants see the FRE primarily as a means for students to acquire basic skills, while others see FRE primarily as a professional challenge and an opportunity to advance the science of agricultural robotics. Our team has clearly defined its goals a number of years ago. We are a team of professionals from a variety of agriculture-related disciplines, including weed science, simulation modelling, image processing, statistics, electronics, software engineering, and mechanical engineering. Our daily work does not involve robotics. FRE provides us with an excellent excuse to pool our knowledge and work on interesting problems outside the scope of our daily work. Because the preparation for FRE takes place after hours, at home, we involve our children as much as possible. This involvement has ranged from designing art work, to functioning as a sparring partner when developing ideas, to assembling a component from Meccano parts and tie-wraps.

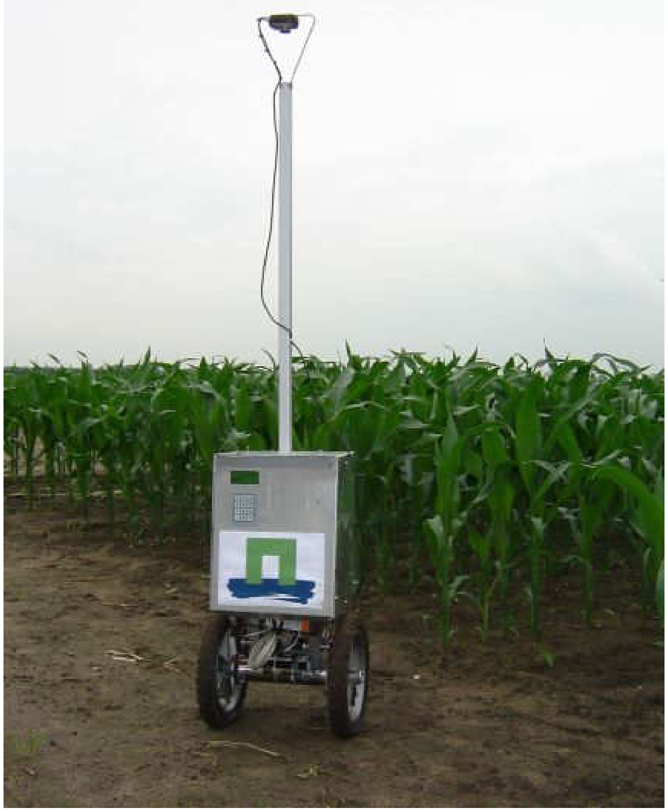


Figure 2. Robot "Hielke" actively balancing in front of some maize

At the same time, because we are professionals, a powerful motivator is the potential to apply in practice the concepts developed during FRE. The perfect example of this is the detection of broad-leaved weeds in grassland that we demonstrated in Hohenheim in 2006. This had led to funding to build a farm-sized prototype of a robot that will detect and eliminate broad-leaved dock (*Rumex obtusifolius L.*).

From the above it should be clear that our goals are, first, to have fun building a robot with friends and family, and second, to take on the challenge to do something new and potentially useful. Below it shall become clear that the fun part of the 2007 robot is building a two-wheeled balancing robot, while the useful part is to develop probabilistic localization.

1.1 Sensor technology and robotics for precision farming

The goal of precision farming is to maximize profitability and minimize environmental damage by managing each part of an agricultural field at just the right time, using just the right amount of fertilizer and/or pesticide. This is at odds with the ongoing trend to minimize the cost of labour by using ever larger machines. Previously, we have built three small, autonomous machines ("robots") that demonstrate the potential to engage in precision-farming without incurring large labour costs. The first of these was built in 2004 (Van Evert et al., 2004). The second robot detects volunteer potato plants in maize fields (Van Evert et al., 2006). The third robot detects broad-leaved weeds in organically farmed grass fields (Polder et al., 2007).

The platform that we used during 2004-2006 had received significant wear; using it for the 2007 Event was not an option. The last three years have shown that we need more robust navigation. For the 2007 Field Robot Event, we decided to take on two challenges: construct a new platform, and implement a radically different navigation strategy.

2 Materials and methods

2.1 Platform

Two constraints with small robot are weight and robustness. A two-wheeled robot can be light and robust because it doesn't need a steering mechanism (O'Halloran et al., 2005). If the center of gravity of such a robot is below the axle, it will act as a tumbler and always stay upright. If the center of gravity is above the axle, the robot must actively balance itself by always driving the wheels in the direction that it is falling. The balancing can be based on a tilt sensor that measures the deviation of the robot's vertical axis with respect to the vertical.

Quite apart from the usefulness of a having a robot balance, it seems to many people an improbable feat and it catches the eye of whoever looks at it. Thus, we decided to build a balancing robot.

We constructed a chassis that houses two Bosch drill motors, one for each wheel. The wheels were regular 12” children's bicycle wheels. The motors were connected to a Roboteq AX3500 motor controller. The sensors were a Creative NX Ultra webcam, a Kübler incremental rotary encoder on each wheel, and a FAS-G inclinometer. Computing power was provided by a PC for image processing and a microcontroller for low-level control. The PC was a mini-ITX PC with a 1.6 GHz Pentium-M, for image processing and state machine. The operating system was Windows XP. Image analysis software was written using VXL image processing library, DSPack wrapper for DirectX, using Borland Delphi and Microsoft Visual Studio. The microcontroller was the Philips LPC2129 on an Olimex board. The microcontroller program was developed using Keil's PK-ARM development environment and the FreeRTOS operating system.

The new robot was named “Hielke”, after Sietse's brother in the popular Dutch series of books by H. de Roos; it is shown in Fig. 1.

2.2 Balancing

Reports of successful two-wheeled balancing robots abound. The nBot (Anderson, 2007) is perhaps the most well-known balancing robot; it certainly seems to have been one of the first in recent years.

The physics of a balancing robot are the same as those of an inverted pendulum. The inverted pendulum is often used as an example in textbooks on control theory (Kwakernaak and Sivan, 1972). It is well-known that a simple PD controller is sufficient to balance an inverted pendulum.

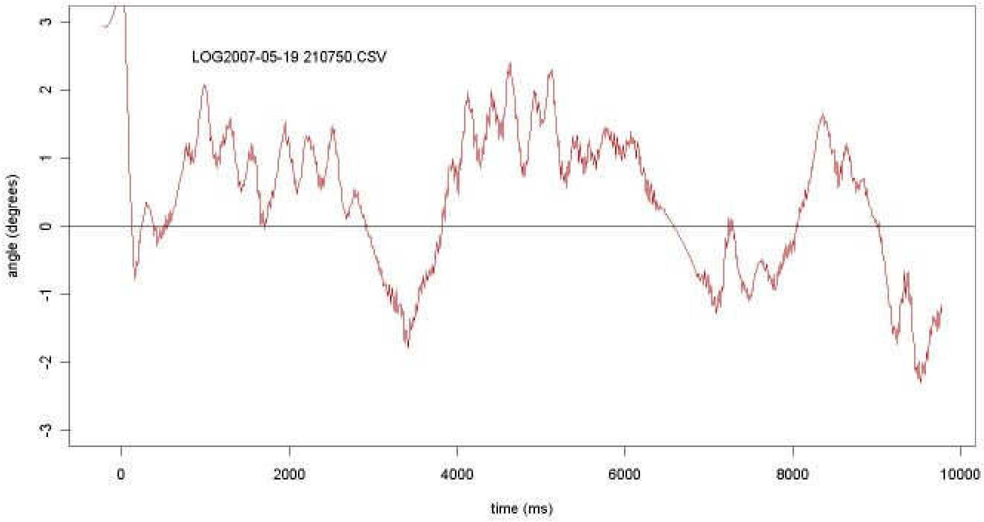


Figure 3. Output from the inclinometer during a test run.

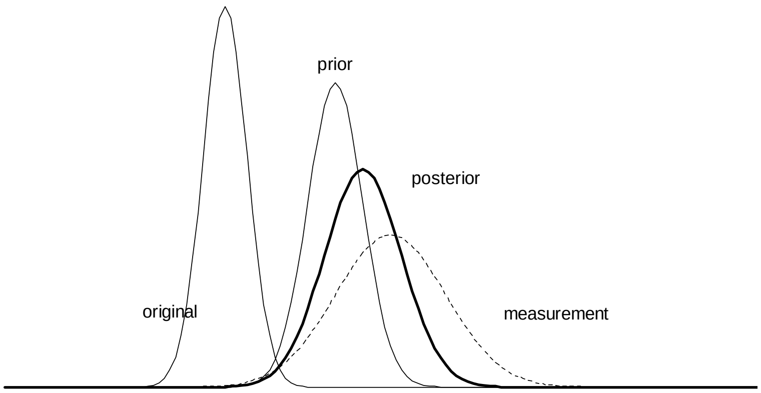


Figure 4. Probabilistic localization of a robot moving in one dimension. For explanation, see text.

We ran into a fair amount of difficulty parameterizing our PD controller. Finally, we wrote a simulation model using the linearized equations of movement for the inverted pendulum as given by (Kwakernaak and Sivan, 1972). A quick exploration of the parameter space showed clearly which P and D factors we should use for our robot.

While the robot was able to balance well, even while driving over rough terrain, we were not able to stabilize the system completely (Fig. 2). This may well be due to imperfections in the hardware, such as backlash in the gears.

2.3 Bayesian localization

Our previous robots have used the following, simple method of localization. A camera placed at the top of a mast in the center of the robot takes a downward-looking image. The Hough transform is used to detect crop rows. We also use the a priori knowledge that one or two rows may be visible, that they are approximately 0.75 m apart, and more or less parallel to each other. Images are acquired at 10-20 Hz, analyzed separately, and translated into navigation commands.

The above method, in which only the most recent image is used to navigate, is sub-optimal. It would be better to use the information from a sequence of images and utilize odometric information as well. Probabilistic (Bayesian) methods offer a framework to combine information from several sources and take into account the uncertainty associated with each information source (Thrun et al., 2005).

Let’s take the example of one-dimensional movement of a robot. When using probabilistic localization, we express the position of the robot not in absolute terms, but rather in terms of a probability distribution. In other words, we represent the robot’s location not by a point x, but by a probability distribution around x (Fig 3, “original” line). If odometric information tells us that the robot has moved a certain distance, this results in a distribution that is displaced and wider than the previous one. This is the prior distribution (Fig 3, “prior”).

When a measurement of the robot’s position is made, it is expressed as a probability distribution (Fig. 3, “measurement”). Combining the prior distribution and the distribution from the measurement results in the posterior distribution, which is an estimate of the robot’s position in which both measurement information and prior information are used (Fig. 3, “posterior”).

Many methods of implementing probabilistic methods are available, including various forms of Kalman filtering. Here, however, we use a particle filter, in which distributions are represented by a finite number of values or “particles”. One important reason to use a particle filter is its ease of implementation. Another important reason to use a particle filter is because the method can easily be applied to localize the robot within the rows, to localize it with respect to the end of the rows, and to localize it on the headland.

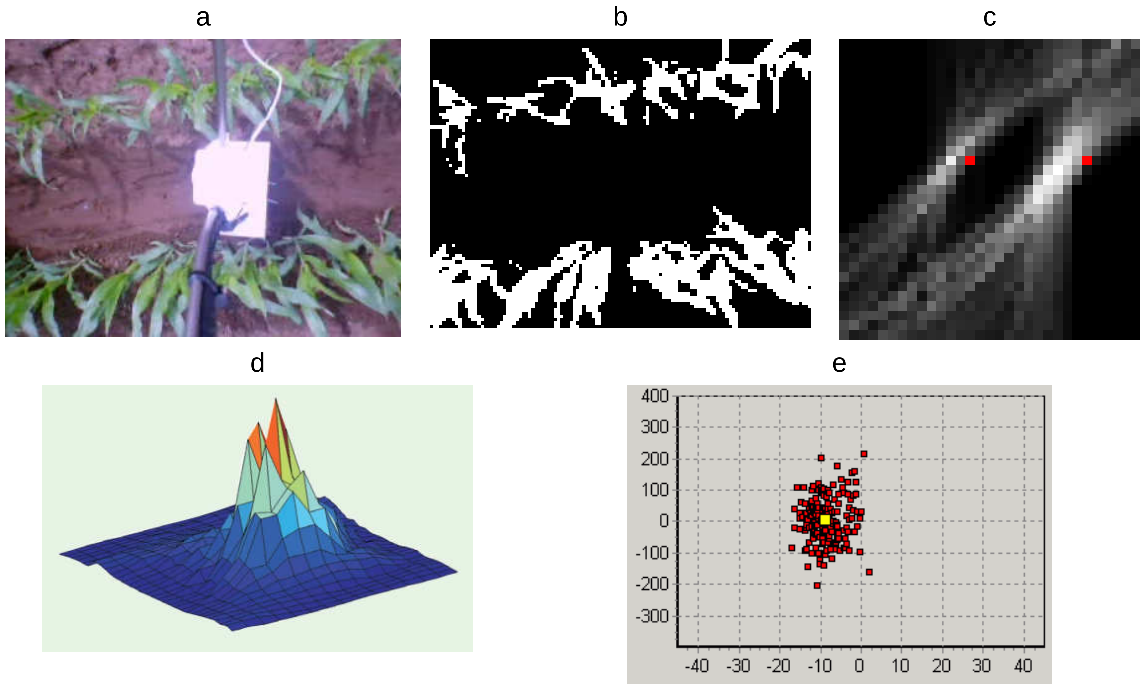


Figure 5. Steps in localization while travelling between rows. (a) original RGB image, (b) segmented image, showing the presence of plants, (c) Hough space, where the brightness of each pixel represents the "greenness" of a line through (b), while the horizontal coordinate represents the distance of that line from the center of the image, and the vertical coordinate represents the heading of that line, (d) measured probability density of the position of the robot, (e) state of the particle filter at a certain point in time, with heading (degrees) on the horizontal axis and lateral deviation (mm) on the vertical axis; each red point represents a particle, and the yellow point represents the mean of the distribution.

While travelling within the rows, the pose of our robot is determined by two state variables, namely the robot's heading relative to the crop rows, and the distance from the center of the robot to the middle of the rows ("lateral deviation"). Note that we measure the speed of the robot with a high degree of confidence, thus the speed does not need to be estimated. When approaching the end of the rows, the robot's pose is given simply by the distance to the end of the rows. Finally, on the headland, the robot's pose is no longer determined by heading and lateral deviation, but by "x" (distance from the end of the rows) and "y" (distance from the center of the robot to the middle of the two rows from which the robot has just emerged).

For each situation (within rows, approaching end-of-rows, and headland), we need to transform the image that the robot sees to a probability density of the pose. Our measurement of the robot's pose is always based on an image taken approximately 1.5 m above the center of the robot, and looking straight down. This image is processed in a different way for each situation.

When travelling within the rows, we segment the image using excessive green and a threshold. We then apply the Hough transform as described earlier (Van Evert et al., 2006) to find crop rows. Projection of an assumed pose into this image leads to two locations where we would expect to find crop rows. We take the brightness around these pixels as a measure of the probability that we would obtain this image given that pose. The various steps of the procedure are shown in Fig. 4. A typical estimate of the robot's pose is shown in Fig. 4e.

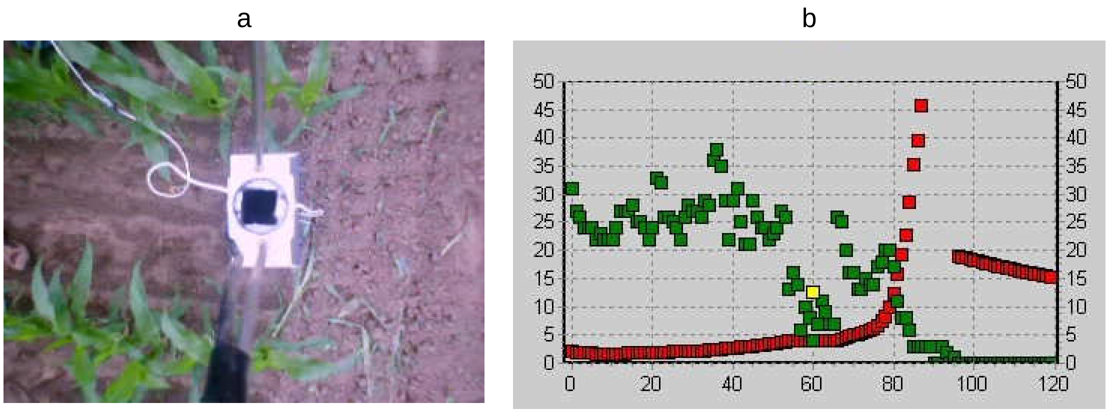


Figure 6. Steps in localization while approaching the end of the rows. (a) original RGB image, (b) green points show the number of green pixels in each column, while red points show the probability density of the location of the end of the rows.

When approaching the end of the rows, we calculate the average green value of the pixels in a column perpendicular to the crop rows. This number will be high and fairly constant for positions in the row, while it will be low and fairly constant for positions outside the row. For each position, we calculate the ratio of the average green value to the left of that point and the average green value to the right. This ratio peaks sharply at the end of the row (Fig. 5) and its value is taken as the probability that we would obtain this image given this distance to the end of the rows.

Time constraints prevented us from implementing localization with the camera on the headland. However, it is easy to see that as long as the row ends are in view of the camera, probability of obtaining a certain image given an assumed pose can be estimated and used in the particle filter. We aim to implement camera-based headland localization in time for the 2008 Field Robot Event.

In controlling the robot, we assume that the robot’s pose is accurately estimated by the mean of the probability distribution. In other words, while the localization is probabilistic, the control is deterministic.

2.4 Headland turn

Given that we did not have sufficient time to implement Bayesian localization on the headland, we use dead reckoning. Once the robot has detected that it is at the end of the rows, it uses odometry to drive to a point that is 100 cm away from the row ends and in the middle of the new row (under the contest rules, this might be any number of rows from the one that it has just left). Once it has reached that point, it makes an appropriate turn and approaches the new row. In this way, we maximize the opportunity for the robot to adjust its course while approaching the new row.

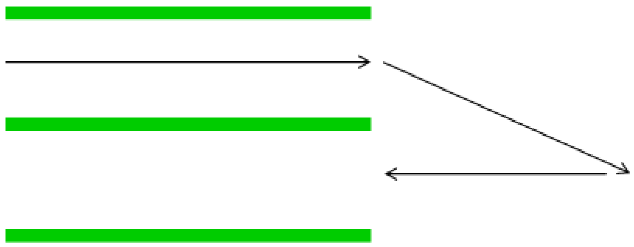


Figure 7. The path that the robot is programmed to follow using dead-reckoning while on the headland. For details, see text.

2.5 Counting golf balls

The images used for row following were also used for detecting golf balls. Golf ball pixels were found by applying a formula of the form rR + gG + bB, where R, G and B are the intensity of pixels in the red, green and blue bands; and r, g and b are parameters. Optimal parameters were found by using a number of training images for which the segmentation had been performed by hand. After segmentation, golf balls were found by thresholding, erosion and dilation, after which each blob was taken to represent a golf ball. The counting algorithm deals with overlap in the images by considering only increases in the number of golf balls.

2.6 Freestyle: The jury deserves a beer

The purpose of our freestyle task was to draw attention to the fact that our robot balances on two wheels. What better way to demonstrate its stability than to drive over rough terrain and carry –without spilling– a glass of beer to the jury? For this task, we controlled the robot remotely using an RS232 cable.

Figure 8. Hielke has dressed up as a waitress to serve the members of the jury a beer.

3 Results and discussion

The robot was able to balance quite well, taking the bumps and other obstacles of the contest field without a problem. We had a couple of anxious moments when heavy rain on the morning of the contest made the field considerably more slick than during the trials. The soggy soil provided less traction and this caused the balancing the fail. Luckily we received some tried-and-true advice from experienced agricultural engineers. Accordingly, we were able to increase traction to the required level simply by deflating the tires. The particle filter performed well and the difference between it and the old, image-by-image method was remarkable. When we switched from the old method to the new one, a hesitatingly, zig-zagging robot was transformed into a smooth and self-confident one. Thus, the robot was able to navigate through the cornfield with a speed of 0.3 m/s and an update rate of 10 images per second.

The maximum speed of 0.3 m/s was determined not by limitations of the row-following algorithm, but by the robot's center of gravity. The center of gravity was relatively high, resulting in lateral instability and sideways falling when serious bumps were taken with speed. A contributing factor was that the wheels were fairly narrow; this caused the robot to dip quite a bit when one wheel hit a depression.

We were plagued by serious lack of time during the last few weeks before the contest, much more so than in previous years. We have already noted that there was no time to implement Bayesian headland localization and dead-reckoning had to be used on the headland. Unfortunately, we had tested left turns only and thus didn't discover the bug that caused right turns to go very wrong.

Another kind of problem caused end-of-row detection to perform less than perfect during the contest. In the weeks leading up to the contest, we had consistently worked with a image acquisition rate of 15 Hz. At the last moment we decided to change this to 10 Hz. The particle filter worked so well that 10 images / sec turned out to be more than enough and we hoped that the lower frame rate would reduce the CPU load to the point where it would perform a little bit more consistently. However, we failed to adjust the parameters of the particle filter for end-of-row detection to the new frame rate.

Golf ball counting was a failure. At the time of the contest, we were completely baffled: the algorithm had worked flawlessly during the 2006 Event. Examination of logged images, however, showed that specular reflection from the robot's housing had caused incorrect exposure of the images, which in turn caused the algorithm to fail.

4 Conclusions

Despite our disappointing sixth place in the overall ranking, we met our twin goals of fun and a useful new development. Until the day of the contest, not a single person outside of our team believed that it would be possible to have a two-wheeled balancing robot in the field. In fact, one of the competitors confided afterwards that he was sure the movies we published before the contest were tricked and meant to confuse the other contestants. It was fun to prove them all wrong!

Probabilistic robotics has taken a huge flight in recent years and forms the basis of some of the worlds most successful robots (Thrun et al., 2006). Using probabilistic methods, we were able to address the localization problems that have bedevilled our robots in the last three Field Robot Events.

For next year's event, it seems likely that we will rebuild our robot to increase its passive stability. We will fix the bugs related to end-of-row detection and turning and we will implement probabilistic localization on the headland. In short, we feel that we now have the knowledge to build a robot capable of running a perfect course in the 2008 Field Robot Event.

Acknowledgments

Jan Bontsema helped us understand the finer points of a PD controller for balancing. We thank our sponsors, foremost of which is Plant Research International's unit Agrosystems Research. We gratefully acknowledge support received from Tamiya (www.tamiya.com), Keil (www.keil.com), Tritec (www.tritec.nl), and Analog Devices (www.analog.com).

References

Anderson, D.P. 2007. nBot Balancing Robot [Online] http://www.geology.smu.edu/~dpa-www/robo/nbot/ (verified 10 October 2007).

Kwakernaak, H., and R. Sivan. 1972. Linear optimal control systems John Wiley & Sons, New York.

O'Halloran, D., A. Wolf, and H. Choset. 2005. Design of a high-impact survivable robot. Mechanism and Machine Theory 40:1345-1366.

Polder, G., F.K. Van Evert, A. Lamaker, A. De Jong, G.W.A.M. Van der Heijden, L.A.P. Lotz, T. Van der Zalm, and C. Kempenaar. 2007. Robot system for weed detection using textural image analysis. 6th Biennial Conference of the European Federation of IT in Agriculture (EFITA), Glasgow.

Thrun, S., W. Burgard, and D. Fox. 2005. Probabilistic robotics MIT Press.

Thrun, S., M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. 2006. Stanley: The robot that won the DARPA Grand Challenge. Journal of Field Robotics 23:661-692.

Van Evert, F., A. Lamaker, K. Van Evert, G. Van der Heijden, A. De Jong, J. Lamaker, B. Lotz, J. Neeteson, and T. Schut. 2004. Een grote toekomst voor kleine robots in open teelten. Agro-Informatica 17:21-26.

Van Evert, F.K., G.W.A.M. Van der Heijden, L.A.P. Lotz, G. Polder, A. Lamaker, A. De Jong, M.C. Kuyper, E.J.K. Groendijk, J. Neeteson, J. , and T. Van der Zalm. 2006. A mobile field robot with vision-based detection of volunteer potato plants in a corn crop. Weed Technology 20:853-861.

Autonomous wheeler “KU-FR01”

Michihisa Iida, Masashi Fukuta, Hiroshi Tamaki
Laboratory of Field Robotics, Graduate School of Agriculture, Kyoto University,
Kitashirakawa-oiwake-cho, Sakyo-ku, Kyoto 606-8502, Japan
E-mail: iida@elam.kais.kyoto-u.ac.jp

Abstract

KU-FR01, which means Kyoto University Field Robot prototype 1, is an autonomous wheel-type robot. It is designed and developed to navigate through maize plants in a field. In this paper, a technical overview of the design and control of the robot is described.

1 Introduction

Members of our team are two graduate students and an associate professor in Kyoto University, Japan. A robot developed to entry the Field Robot Event 2007 is 4-wheel-driven by an electric motor. Dimension of the robot is 360 mm in length, 230mm in width and 3.0 kg in weigh. As we have to carry the robot from Japan to Netherlands by air, the dimension of the robot is limited. The robot consists of a main microprocessor for control, five infrared distance measuring sensors, a gyro sensor, a photo transistor for detecting rotational speed of rear propeller shaft and so on.
Field Robot Event 2007, organized by Wageningen University, has been competed in four tasks as follows:
Task 1. Robust navigation in a maize field with curved rows.
Task 2. Advanced robust navigation in a maize field with straight rows.
Task 3. ‘Weed’ – control in a maize field.
Task 4. Free style.
Our team has competed in the task 1 and 2 as our robot has not completed a vision system for detecting yellow golf balls.

2 Materials and methods

2.1 Hardware

2.1.1 Platform

The base vehicle for our field robot is a 1/10 scaled radio-controlled 4x4 pickup truck produced by Tamiya. This vehicle is electrically driven and features four-wheel-drive, four-independent-suspension, three-range transmission, and serve steering. Dimension of the robot is 360 mm in length, 230mm in width and 3.0 kg in weigh, including batteries. A main electric motor is type RS-540RH (Mabuchi Motor, Japan). Output and torque of the motor are 14.6W and 14.4 mN-m, respectively. A FET amplifier (SCR-6712, Yokomo, Japan) is applied to control the speed of the main motor. Servomotors that are used to operate the steer and transmission are type S3003 (Futaba, Japan). Operating speed and torque of the motor are 316deg/s and 400 mN-m, respectively.

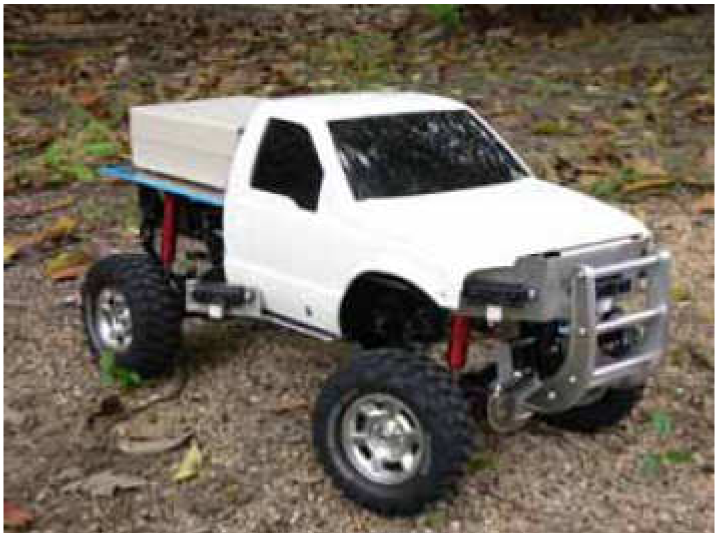


Figure 1: The robot “KU-FR01” developed for Field Robot Event 2007.

2.1.2 Sensors

In order to detect maize plants and some obstacles, we make use of five infrared distance measuring sensors (GP2D12, Sharp, Japan, hereinafter IRDM sensor). The sensors can measure a range of approximately 10 - 80cm. A sensor is mounted on the front of vehicle and is used to avoid collision with obstacles. Two sensors are mounted at the angle of 46 deg from the right front on the left or right front of the vehicle, respectively. The other two sensors are placed at the angle of 51 deg from the right front on both sides of the vehicle. They are used to detect maize plants and to navigate through the plants.

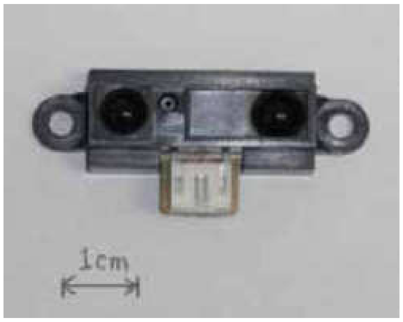


Figure 2: The infrared distance measuring sensor “GP2D12” (Sharp, Japan).

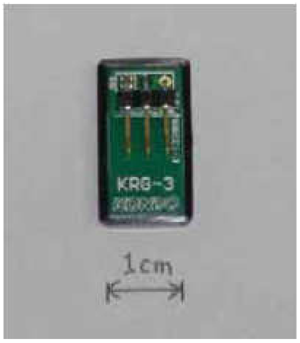


Figure 3: The gyro sensor “KRG-3” (Kondo Kagaku, Japan).

A gyro sensor (KRG-3, Kondo Kagaku, Japan) is employed to detect the yaw rate of the vehicle while turning at the headland. It is installed in a control box for water-proof. Output of the gyro sensor is adapted a high pass filter of 0.4Hz. Additionally, we adapt a low pass filter of 25 Hz and amplify the

output voltage of sensor two times. As it was, however, difficult for the gyro sensor to detect the yaw rate of the vehicle while turning at the headland, the data of the yaw rate was not used for turning in the Field Robot Event 2007.
A Photo transistor that is removed from a broken electric mouse is used to determine the current speed of the vehicle. It is installed at the rear propeller shaft to detect the rotational speed of the shaft. Furthermore, we obtain the travel distance by summing the pulses that the transistor generates.

2.1.3 Main board

A main board is YS7045-3 (YellowSoft, Japan). The microprocessor that is mounted on this board is SH2/7045F, 20MHz (Resesas Technology, Japan). This board is programmed by C language compiler "YCSH" (YellowSoft, Japan). It features a serial connection (RS-232C) with the PC. To communicate between the PC and the microprocessor a wireless communication unit as mentioned in section 2.1.4 is employed in the vehicle.

A steering servomotor, a FET amplifier unit for the electric motor and a transmission servomotor are connected to the PWM output pins. Five infrared sensors and the gyro sensor can be connected to the 10-bit AD converters. The pulse outputs form the photo sensor is connected to the _IRQ0 interrupt pin.

A push switch is set up to command "RUN" or "STOP" for the program.

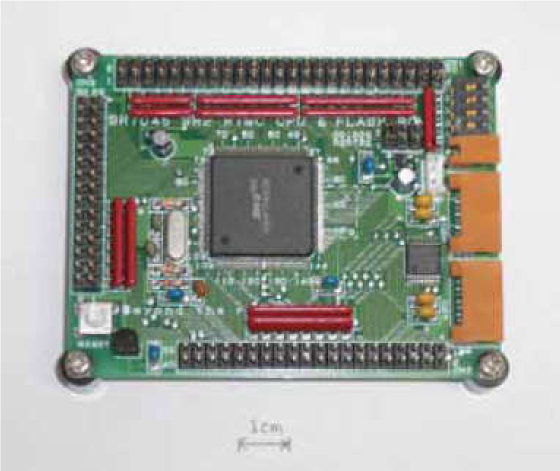


Figure 4: The main board "YS7045-3" (YellowSoft, Japan).

2.1.4 Wireless communication

The wireless communication device features 2.4GHz frequency band and is connected to the serial port. The maximum throughput of the device is 250kbps and the area that it can communicate each other is about 10m. We use the device in the peer-to-peer mode for the communication between the PC and the microprocessor.

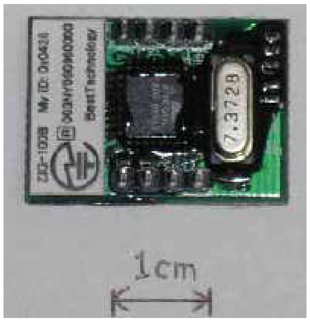


Figure 5: The wireless communication device "ZIG-100 B" (BestTechnology, Japan).

2.1.5 Power supply

The robot has two independent power sources. The control system as the microprocessor and the sensors is supplied directly from four rechargeable Ni-MH batteries of AA-size. The main motor and two servomotor are powered by a model car racing pack 7.2V.

2.2 Software

Figure 6 is a schematic diagram of the software. Some parts of the software are described.

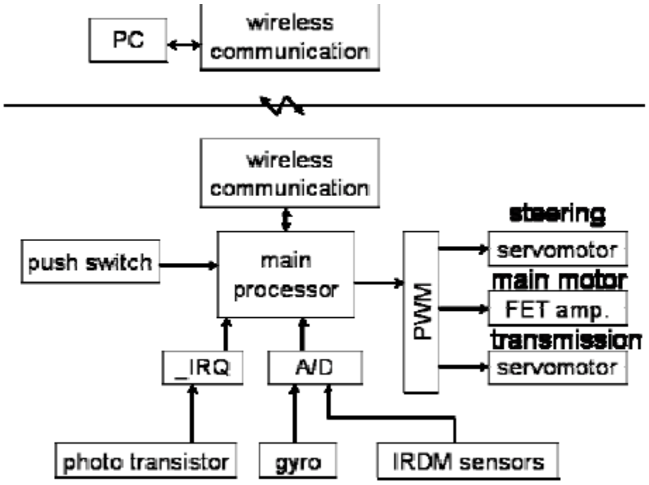


Figure 6: The schematic diagram of the software.

2.2.1 Speed and steering control

The microprocessor SH2/7045F has five channels of the multi-function timer pulse units (hereinafter MTU). MTU can be used for a counter, interval timer, PWM, and so on. We use three channels of the PWM of the MTU. All controls by the PWM are running in the duty ratio range of 6.5 – 23 % at a 100 Hz cycle, depending on the specification of the used servomotors.
The PWM0 controls the steering servomotor proportionally according to the pulse width. The minimal turning radius of the vehicle is about 650mm. The steering angle is determined by the method as mentioned in the next section.
The PWM1 adjusts the speed of the vehicle. The PWM2 move the servomotor to shift the transmission of the vehicle. The maximal speed of the vehicle is 2.7m/s at the high range of the transmission. However, the speed of the vehicle at high range is too fast to control the steer quickly. In the maize field the transmission of the vehicle is kept at the low range. The maximal speed of the vehicle is 1.2 m/s at the low range.

2.2.2 Following control

IRDM sensors are used to navigate the robot through the maize plants. Two sensors equipped on the left of the vehicle can measure the distances $L_1$and L_3 between the robot and the left side plants. The positions $(x_1,y_{1,})$ and (x_3,y_3) of the sensors and the angles $\square_1$ and $\square_3$ of sensors are measured in advance. Then a left relative angle ($\square_L$) between the heading angle of the robot and the direction of left crop row is calculated by these measurements. This calculation is done by the following equation.

$$X = \left(x_1 + L_1 \cos\theta_1\right) - \left(x_3 + L_3 \cos\theta_3\right) \tag{1}$$

$$Y = \left(y_1 + L_1 \sin\theta_1\right) - \left(y_3 + L_3 \sin\theta_3\right) \tag{2}$$

$$\alpha_L = a\tan 2\left(X, Y\right) \tag{3}$$

A right relative angle (□$_R$) is determined by the same way.

The robot controls a steer to avoid collision with crops by reference of left or right relative angle during navigation. If both □L and □R are calculated, an average of both angles is used to control. This enables the robot to follow even the curved rows. If one or a few IRDM sensors don’t measure and only □$_L$ (or □$_R$) is calculated, the robot follows the rows of the left-hand rows by □$_L$. This enables the robot to follow the crop rows with one side measurement alone.

If the measured distance is less than threshold value, the robot steers to the opposite side regardless of relative angle.

While the robot runs between crop rows, it gets travels distance by the photo transistor.

The robot stops at once and begins to turn at the headland if the following conditions are met.

- The travel distance is over the length of crop row.
- All IRDM sensors don’t detect the crops while traveling 10cm.

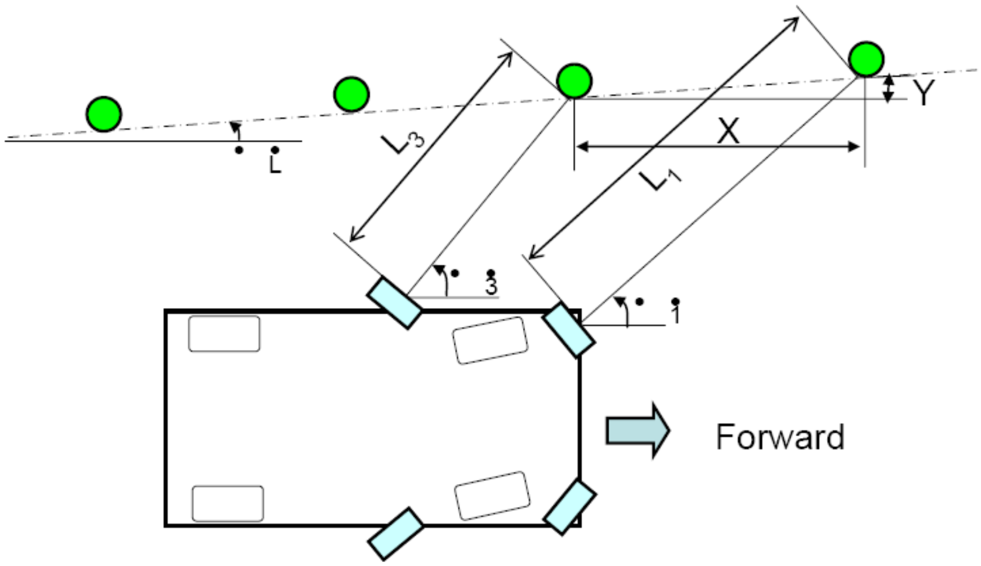


Figure 7: Usage of IRDM sensors for following control.

2.2.3 Turning control

After the robot finishes running through crop rows, it begins to turn at the headland. The robot turns at specified degrees at first, and then it goes straight in setup length L. Moreover, it turns at specified degrees while backing. Finally, it goes straight again to proceed into the crop row avoiding collision with crops.

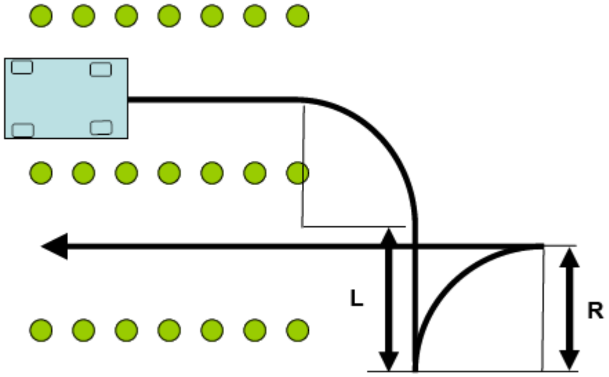


Figure 8: Headland turn.

The turning angle is measured by a gyro sensor. But the data of the yaw rate is so noisy that we don't use the gyro sensor. Instead, we got the turning angle by the traversed distance and turning radius R. The robot can enter the given path by changing setup length L. We set the length L according to inter row spacing in Field Robot Event 2007.

After backing, the robot proceeds into the crop rows at a slow speed. It steers to avoid collision with crops by using all IRDM sensors and begins to follow the new rows by above following control when this proceeding distance become over 2R.

2.2.4 Route map

Figure 9 shows the route map for example. In Field Robot Event 2007, the robot should be able to follow a certain pre-defined pattern over the field, so the robot is set a sequence of path information such as the length of crop rows, the turn direction, the turning angle while advancing, the traverse distance, the turning angle while backing and so on.

The both former and latter angles for turning at the headland can also be set respectively. If the end of the crop rows and the edge of the field are not perpendicular, making only a right-angled turn causes the turning failure at the headland because most suitable angle is not always 90 degrees (See Figure 10). In addition the turning angles are calculated from the traversed distance, so the measurements of angles are susceptible to bad ground condition because of slips, slope or rough. It is useful to adapt the input value for the circumstances. This needs preliminary tests, but it enables the robot to turn more accurately at the headland.

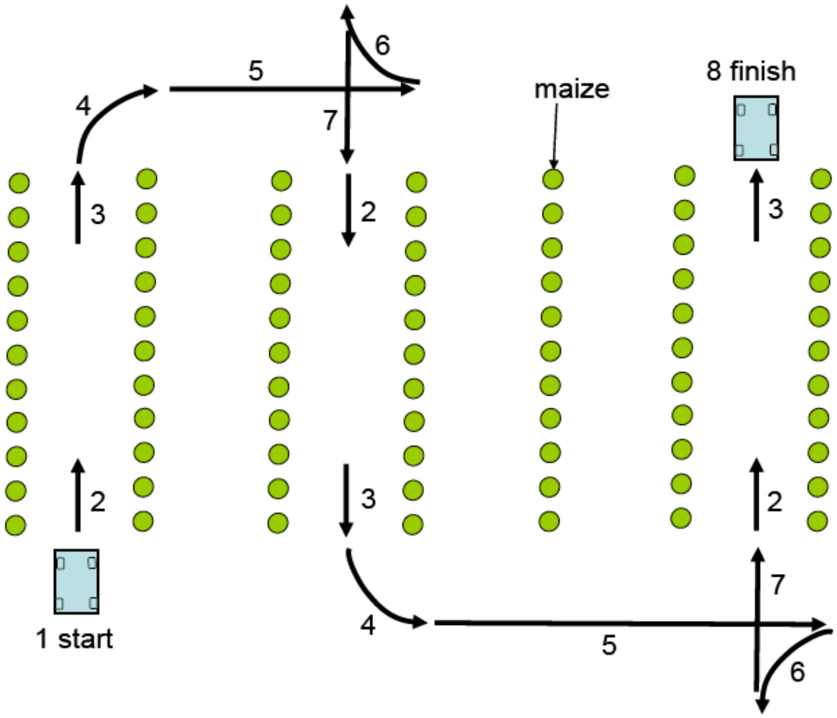


Figure 9: Route map and tactics.

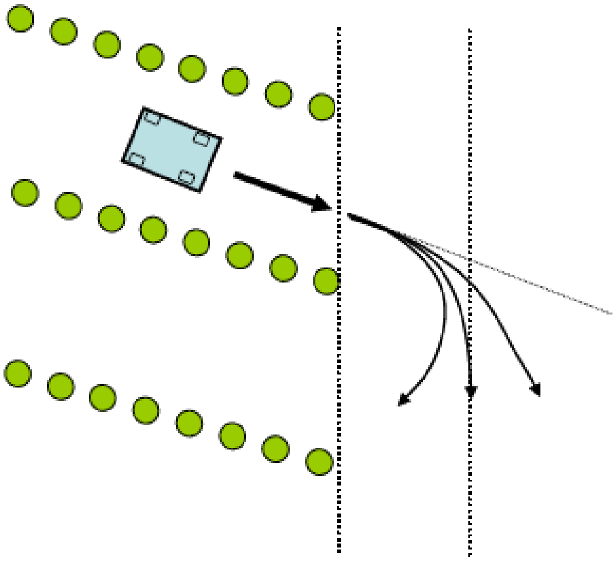


Figure 10: The most suitable angle of traverse.

2.3 Tactics

Procedure of our tactics follows corresponding to the numbers as shown in Fig. 9.

1. At the start of the race, the start button is pushed.
2. The robot runs and follows the crop rows by the following control.
3. When the travel distance of the robot comes close to the length of crop rows, it runs at a slow speed to detect the end of crop rows. If all IRDM sensors don't detect any crops while advancing slowly, it begins to turn at the headland.
4. The robot turns to the right according to the path information. The turn direction and turning angle are obtained from the path information inputted in advance.
5. To approach the next crop rows, the robot goes straight at a right angle to the crop rows. The traversed distance is inputted according to the inter row spacing.
6. In order for the heading angle of the robot to be parallel to the crop rows, the robot turns to the left while backing. The turning angle while backing is pre-set from the traversed distance measured in advance.
7. The robot runs at a slow speed and gets into the crops. All IRDM sensors are used to avoid collisions with the plants.
8. The robot stops at the end of the crop rows when it completes the race, and then it stands ready to restart.

3 Discussion

We took part in Field Robot Event 2007 in 14th – 16th of June. It was a first time to apply our robot to an actual maize field. The wheels of our robot were often slipped because they were too small to run on the playground. Sometimes we had to push it when it slipped and didn't proceed. Since the robot is small and short in height, soil matter which the wheels threw up ruined IRDM sensors. So the robot needs to grow in size and become insusceptible to ground condition.

The function of image processing is not installed, so camera and image processing system are needed for finding yellow golf balls, task 3.

4 Conclusions

At the Field Robot Event 2007 in Wageningen, our team was the eighth place. Our field robot could run in task 1 and task 2 regardless of the small size of wheels and tread.

The development of the robot as a team, doing test drives, and challenge of the overseas competition certainly improve our skills, knowledge, and sprits.

References

Blaanuw, D., Gulik, W. van, Kommer, J., Koning, D. de (2005). “The field robot Conrnhoolio”, Proceedings of the Field Robot Event 2005: 1-9.

Yamamoto, S., Hamada, Y., (2005). “Whirligig beetle – a small scale hovercraft robot”, Proceedings of the Field Robot Event 2005: 91-98.

Autonomous system for **N**avigation **A**nd **D**etection in agricultural environments

Jörg Klever
Email: jklever@gmx.de, a.ruckelshausen@fhos.de
University of Applied Sciences Osnabrück, Faculty of Engineering and Computer Science
Albrechtstr. 30, 49076 Osnabrück, Germany

Abstract

The autonomous field robot NAD has been developed at the University of Applied Sciences Osnabrück as a master thesis. The robot is based on the robotic platform called *ProfiBot* of the Fraunhofer Institute of *Intelligente Analyse- und Informationssysteme.* The centre of the system architecture is a mini PC using a Windows XP operating system combined with an iCONNECT development suite. The iCONNECT software is a comfortable integrated development environment for fast and block oriented programming of industrial applications. The main function of this software is the data acquisition of the sensor information and the calculation of resulting speed and steering commands for the robot.

To give a high robustness to the autonomous row guidiance within the maize rows the robot is equipped with ten distance sensors, used for measuring the distance between the robot and the plants. With the need to make an exact turn at the end of the row it was necessary to integrate a gyroscope to measure and control the turn angle. To detect and calculate the position of the yellow golf balls, simulating dandelions, a smart camera was installed.

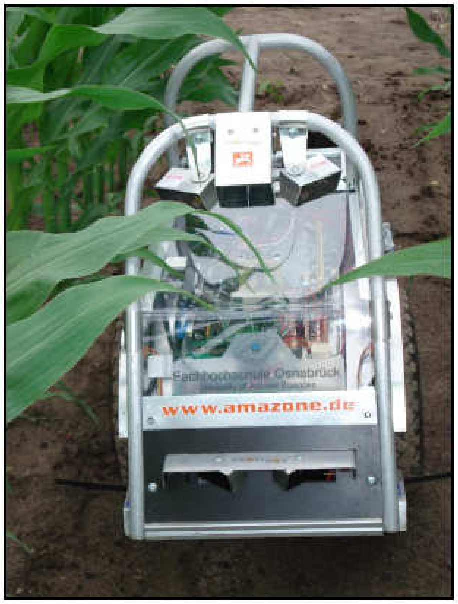


Figure 1:NAD

KEYWORDS

Field Robot, Platform *ProfiBot*, iCONNECT, WLAN Ad-hoc connection

1 Introduction

The robot NAD has been developed for the special tasks of the Filed Robot Event 2007 in Wageningen. The tasks were as described below:

1. The robot should cover as much distance as possible in 3 minutes time while navigating between curved rows of a maize field, making a head-land turn and returning in the adjacent row.
2. The robot should cover as much distance as possible in 3 minutes time while navigating between straight rows of maize plants. The robot should be able to follow a certain pre-defined pattern over the field. At various places in the maize field, plants will be missing in either one or both rows over a length of maximally 1 m. A head-land of only 1.5 m will be available for turning.
3. The robot should cover as much distance within 3 minutes time while navigating between straight rows of maize plants. In the maize field randomly distributed artificial weeds - yellow golf balls have to be detected. Detection of a 'weed' should be demonstrated by producing a clear signal such as a flashlight or a sound. Additionally, a 'weed-killing' operation should be performed on the 'weed'.
4. Robots are invited to perform a free-style operation. Fun is important in this task but agricultural relevance is emphasized. One team member has to inform the public about the idea. **[Fieldrobot]**

2 Hardware

The main advantage of the concept is to use an existing robot platform which has to be modified for the tasks of the Field Robot Event 2007. As this platform a robot, developed by the Fraunhofer Institute of *Intelligente Analyse- und Informationssysteme,* called *ProfiBot* (figure 2.1) was choosen. The robot was selected because of the following features:

- easy to handle lightweight framework
- robust and user-friendly drivetrain
- very high agility

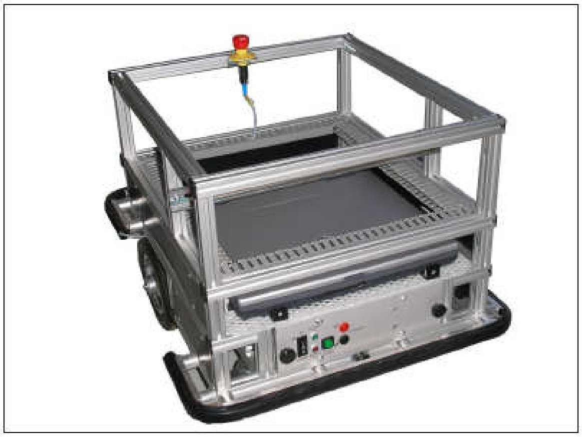


Figure 2.1: ***ProfiBot***

Project Profibot

The *ProfiBot* project has been initialized by the Fraunhofer Institute. It is used as a mechatronic system for training and teaching purposes for industrial education. The main goal of this project is to make the educational fields of mechatronics, mechanics, electronics and computer science more practical and interesting. Additionally the understanding for technical systems will grow which can be an additional qualification for further job-live.

ProfiBot system

Each of the two frontwheels is driven by one of the two maxon dc motors with mounted gearboxes. Steering is automatically done by speed difference between these two front wheels. The engines are equipped with a rotary encoder for the speed and steering control.

The two engines are powered by two 12 V lead accumulators. The robot is also equipped with a sensor which can detect collisions. If a collision is detected, the robot stops immediately.

The centre of the *ProfiBot* system architecture is a laptop mounted onto a drawer (figure 2.2). The controlling of the robot is done by the iCONNECT software.

Additionally, the following equipment belongs to the *ProfiBot*:

- USB Web Cam – Typehoon Easycam 1.3 Mpix
- USB speaker – Sigma SP3000
- USB Experiment Interface Board – Vellman K8055 (2 analog input, 2 PWM output, 5 digital input, 8 digital output)
- USB Joystick – Saitek Cyborg EVO Wireless

In the future, a robot arm will be mounted on top of the ProfiBot (figure 2.3)

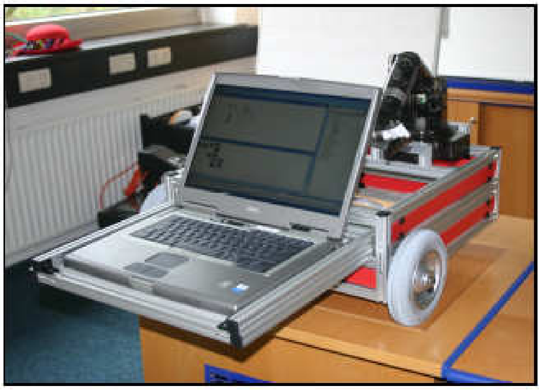


Figure 2.2: *ProfiBot* with laptop [ProfiBot]

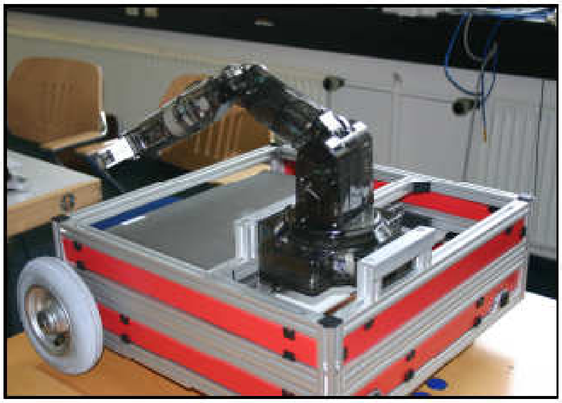


Figure 2.3: *ProfiBot* with robot arm [ProfiBot]

ProfiBot modifications

ProfiBot has been designed for indoor applications. As a consequence the platform has to be changed strongly for the Field Robot Event. The following modifications have been done to the *ProfiBot*:

1. Wheels were changed to bigger ones to be able to drive in rough terrain
2. gear transmission ratio was changed to raise to maximum speed
3. the width of the robot was reduced to make it fit into the maize rows
4. additional sensors were added for the autonomous navigation
5. to reduce the size of the robot a mini PC instead of a laptop was installed
6. an enhanced measuring board was installed to gather the sensor data
7. an additional microcontroller board was used for the analysis of the CMUcam2, the ultrasonic and the gyroscope signals

2.1 Sensors

The robot uses ten distance sensors and two flex sensors combined as redundant sensors systems in a sensor fusion concept to give a higher robustness to the autonomous navigation between the rows. To make a controlled turn at the end of the rows, a gyroscope was integrated. A camera is used for the detection of the yellow golf balls.

Sharp IR distance sensors

The main sensors for the navigation of the robot NAD are eight low-cost IR distance sensors. Four of these sensors are long distance sensors (type GPY0D02Y), appropriate with the direction to the front, are used for the detection of curves.
The other four Sharp IR sensors are short distance sensors (GP2D12). Their main task is the calculation of the position of the robot within the rows.

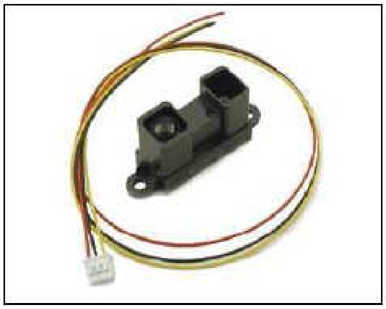


Figure 2.4: IR distance sensor [Acroname]

Ultrasonic Sensors

Additionally two ultrasonic distance sensors (SFR8) are used for the navigation. With their position at the side of the robot, they are able to measure the distance to the maize plants. The main advantage of ultrasonic sensors compared to the IR-distance sensors is the big spot measured.

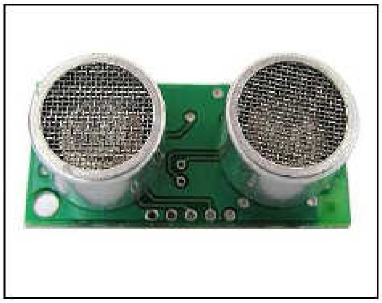


Figure 2.5: Ultrasonic sensor [RT]

Flex sensors

For security reasons, two mechanical sensors have been attached to the front of the robot. These flex sensors consist out of strain gauges which change their electrical resistance when they are touched/bend by the collision with a maize plant.

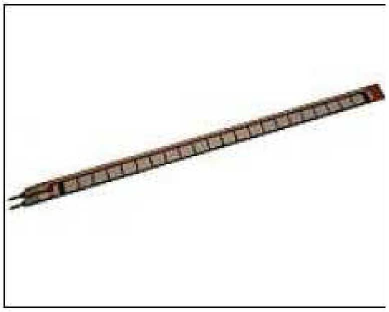


Figure 2.6: Flex sensor [MM]

Gyroscope

The gyroscope is a sensor that can measure the angular velocity of the robot. To calculate the exact direction/angle of the robot, an integration of this angular velocity is done by a microcontroller. The sensor was integrated into a circuit board together with filter circuits to reduce the noise of the signal and to compensate temperature changes. **[optoMAIZER 2005]**

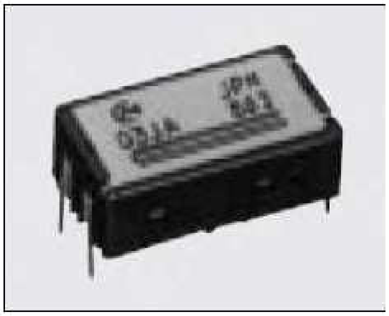


Figure 2.7: Gyroscope

CMUcam2

The CMUcam2 is a smart camera with onboard image processing. Its main feature is to perform tracking of predefined color blobs within different virtual windows and to send the result as reduced data over a serial RS2323 link. It is used to detect the yellow color of the golf balls and to calculate their position within the rows.

The CMUcam2 has already been successfully used for the autonomous robots Eye-Maize, optoMAIZER and Maizerati. **[optoMAIZER 2005] [Maizerati 2006]**

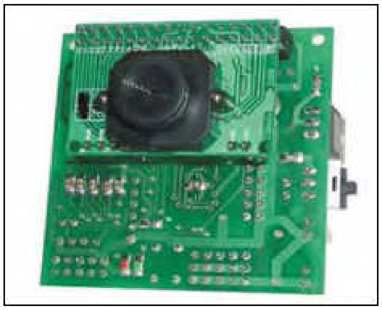


Figure 2.8: CMUcam2

2.2 User interface

Touch display

The touch display is used as an easy way for the user to communicate with the robot without the need to know the whole system in detail. The user can choose different operation modes for the different tasks of the Field Robot Event and have a look at process data. During the navigation the number of the next row to drive into is displayed and parameters can be changed.

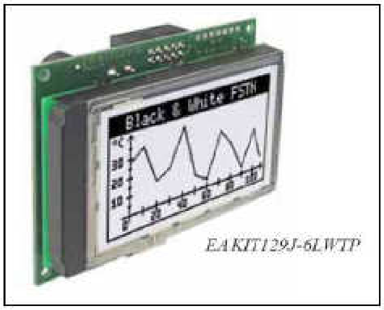


Figure 2.9: Touchdisplay [E.A]

Wireless LAN Dongle

The PC-board which is integrated into the robot system is equipped with a wireless LAN dongle to make it accessible from another computer without the need to use any cables. The WLAN link (ad-hoc) is also used to establish a remote desktop connection using Ultra VNC. The connection is very useful during the test period to make changes to the software without the need to connect a screen, mouse and keyboard to the PC board.

2.3 Controller

Mini ITX motherboard

Because of the reduced size, weight and cost a mini ITX PC motherboard with a VIA EDEN 1,2 GHz CPU was used instead of a Laptop. The supply voltage for the PC board is generated by a Marex Mini ITX 60W power supply and a DC to DC converter which uses the 24 V of the two lead accumulators.

Technical data:

- VIA EDEN 1,2 GHz Prozessor
- 512 MB 533MHz DDRII RAM
- 10/100 LAN on Board
- 8x USB
- 2x COM
- 1x Firewire

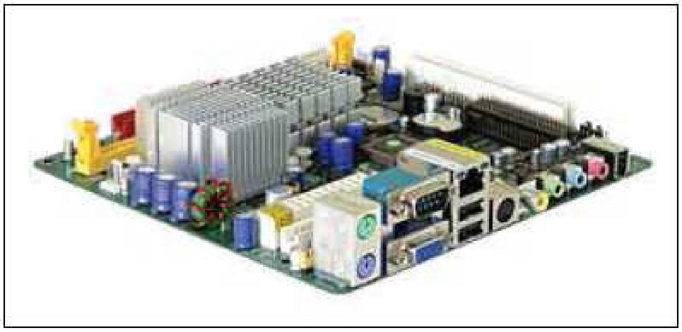


Figure 2.10: Mini ITX Motherboard [Reichelt]

Labjack U12

The measuring board called "*Labjack U12*" produced by Meilhaus performs the data acquisition of the eight IR distance sensors and the two flex sensors. The measuring board can be connected to the PC via USB and it can be easily used with iCONNECT.

I/Os:

- 8 analog inputs (used for 8 IR distance sensors)
- analog outputs (not used)
- 20 digital I/Os (2 used for flex sensors)

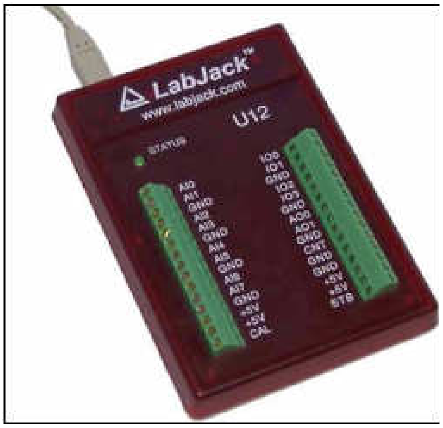


Figure 2.11: *LabjackU12* [Labjack]

Glyn Evaluation Board MB90F340/860

The Glyn Evaluation Board is necessary for the data acquisition of the gyroscope, the CMUcam2 and the ultrasonic sensors. These sensors can't be used with the U12 measuring board because of the need to perform a fast signal preprocessing or the need of the sensors to be connected over a serial RS232 link (CMUcam2) or an i²c link (ultrasonic sensors). The Glyn Evaluation Board was chosen because of the existing experience in working with the Fujitsu MB90F340 microcontroller. The preprocessed data is transferred via RS232 to the PC and can be used for further processes in the iCONNECT environment.

Features of the microcontroller

- 1xCAN Interface
- UART
- 24MHz
- 512K Flash
- I²C (400Kbit/s)
- A/D Converter (24 input channels with 10 or 8-bit resolution)

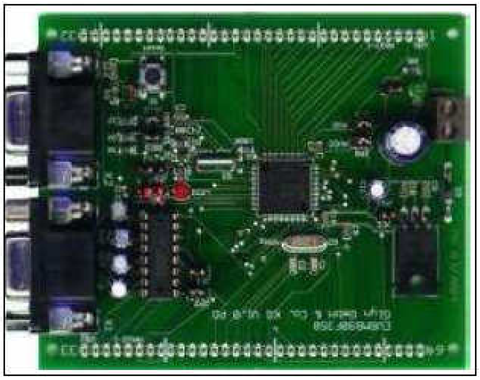


Figure 2.12: Glyn Evaluationboard [Glyn]

3 Software

The operating system installed on the PC is Windows XP. For the data acquisition of the sensor signals the software iCONNECT was installed. The software iCONNECT is a comfortable integrated development environment for a fast block oriented programming. Figure 3.1 shows the block diagram of NADs software.

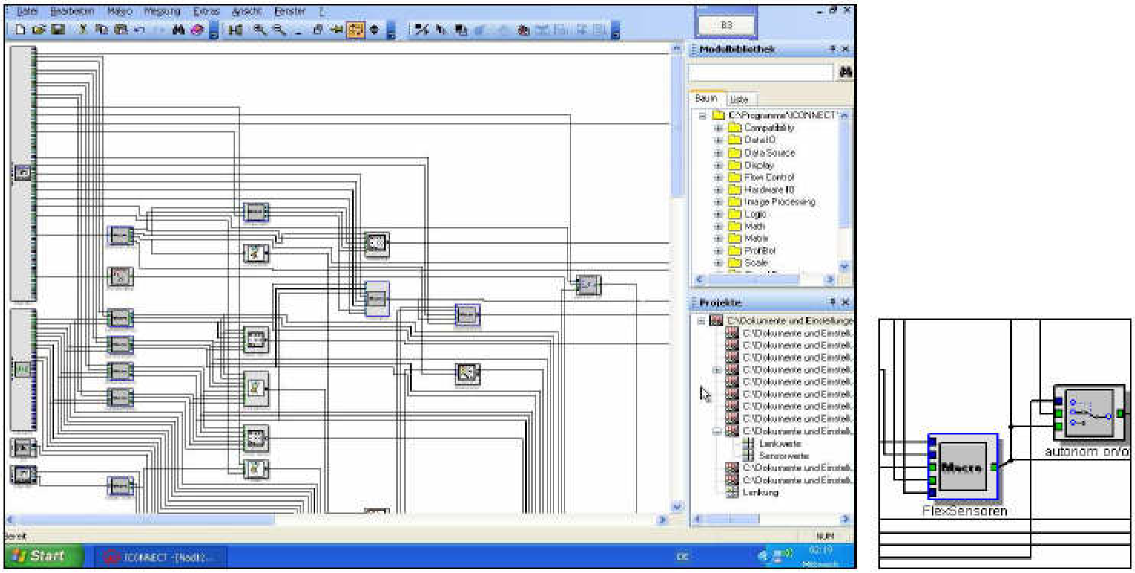
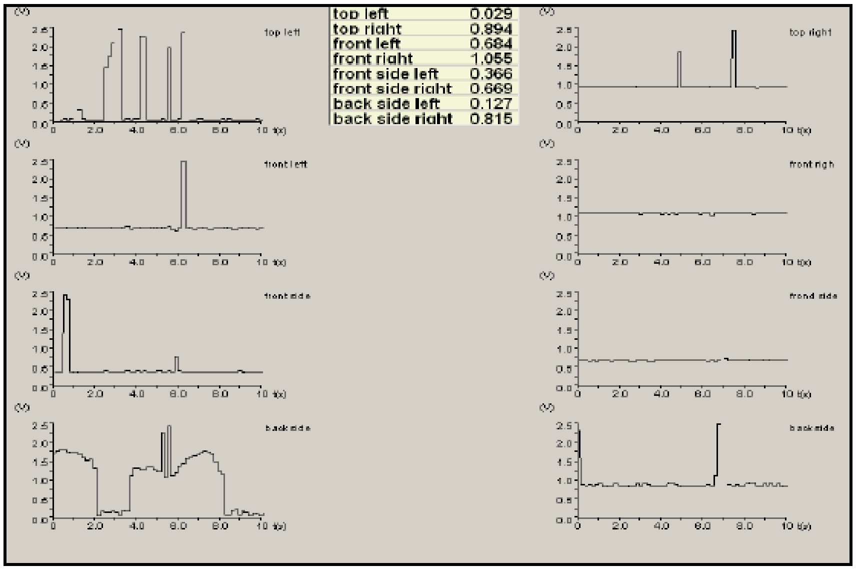


Figure 3.1: Block diagram of NAD in iCONNECT

Figure 3.2 shows the sensor data of the eight IR distance sensors gathered with the Labjack measuring board and visualized with the iCONNECT software.

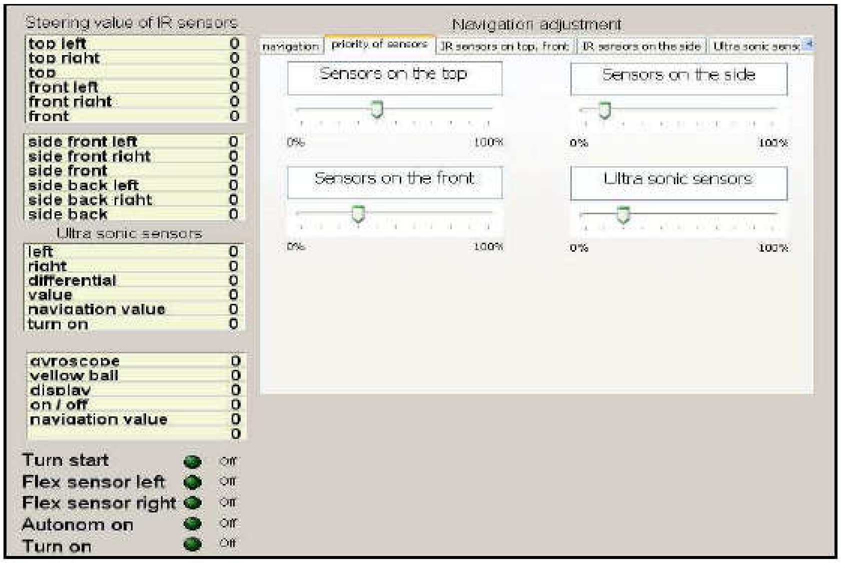


top left	0.029
top right	0.894
front left	0.684
front right	1.055
front side left	0.366
front side right	0.669
back side left	0.127
back side right	0.815

Figure 3.2: Sensor data of the eight IR distance sensors

Graphical user interface (GUI) von NAD

A feature of the iCONNECT software is to create graphical user interfaces in an easy way with the build in control elements. Important parameters can be changed and process values can be displayed. On the left side of the screen the steering and sensor values are shown. On the right side is a tabbed control where algorithm settings and other parameters can be changed.



Steering value of IR sensors

top left	0
top right	0
top	0
front left	0
front right	0
front	0
side front left	0
side front right	0
side front	0
side back left	0
side back right	0
side back	0

Ultra sonic sensors

left	0
right	0
differential	0
value	0
navigation value	0
turn on	0

gyroscope	0
yellow ball	0
display	0
on / off	0
navigation value	0
	0

Turn start	Off
Flex sensor left	Off
Flex sensor right	Off
Autonom on	Off
Turn on	Off

Navigation adjustment

Figure 3.3: Graphical user interfaces (GUI) of NAD

For the navigation the row is divided into five different zones (figure 3.4). Each of these zones stands for a predefined steering direction and strength which can be easily changed with the graphical user interface (figure 3.5). For the calculation of the position of NAD within these zones four sensor pairs are used: the two IR distance sensors mounted to the top of the robot, the two front IR distance sensors, the two side IR distance sensors and the two ultrasonic sensors.

Every pair calculates the current zone which results in a steering direction and strength this pair. Additionally each sensor pair is given a priority (figure 3.6). To calculate the final direction, the results of all the pairs are combined regarding their priority.

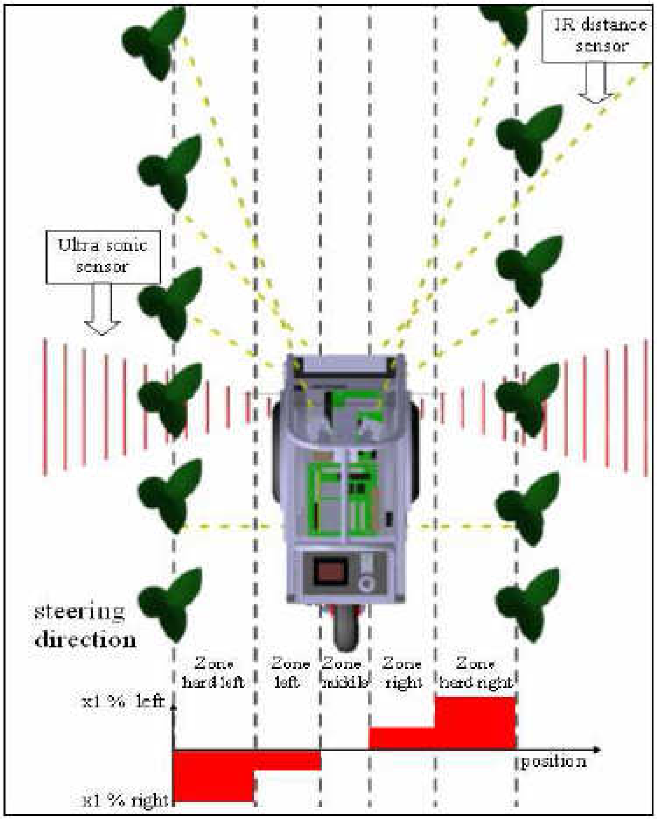


Figure 3.4: The five zones

Another steering decision is done by the flex sensors. They have been given an even higher priority than all other sensors, because of their ability to detect collisions with the plants. The resulting steering direction and strength can be changed in the GUI.

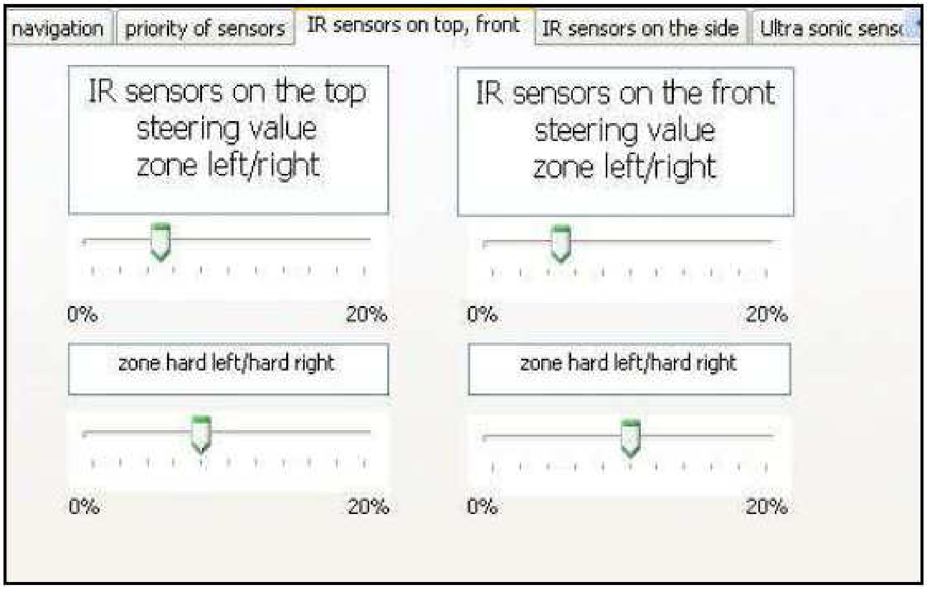

navigation
priority of sensors
IR sensors on top, front
IR sensors on the side
Ultra sonic sens
IR sensors on the top steering value zone left/right
0%
20%
zone hard left/hard right
0%
20%
IR sensors on the front steering value zone left/right
0%
20%
zone hard left/hard right
0%
20%

Figure 3.5: Table steering values

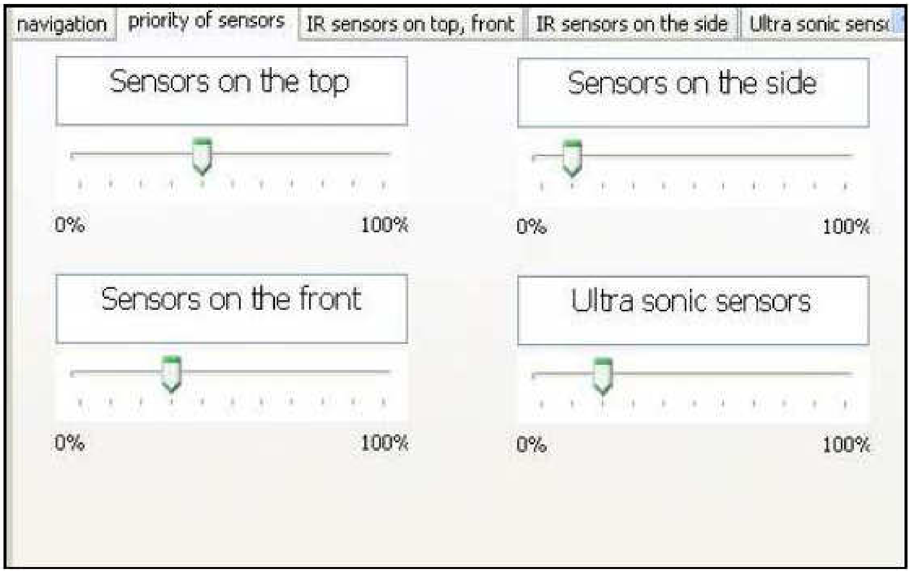


Figure 3.6: Table priority

4 Results and conclusion

At the beginning of the outdoor tests the robot had problems with the roughness of the field which has been solved by increasing the axis-center distance and optimizing the control algorithms. The navigation through the row worked very well during the contest. Another problem was the uncompleted programming of the turn at the end of the rows and the ball detection. Reasons for this are, among other things, the problems which appeared while programming with iCONNECT and its slow performance.

References

[ProfiBot] www.Profibot.de
[Fieldrobot] www.fieldrobot.com
[iCONNECT] www.micro-epsilon.de
[Labjack] www.labjack.com
[Reichelt] www.reichelt.de
[Acroname] www.acroname.com
[Glyn] www.glyn.de
[E.A] www.electronic-assemply.de
[RT] www.roboter-teile.de
[MM] www.micromaus.de

[optoMAIZER 2005] “Field Robot optoMAIZER” ; Ralph KLOSE, Martin MEIER (diploma thesis supervised by A. Ruckelshausen, A. Linz); Field Robot Event 2005, Wageningen / The Netherlands, Proceedings, ISBN 90-6754-969-X, November 2005

[Maizerati 2006] “Field Robot Maizerati” ; Ralph KLOSE, Jörg KLEVER, Hartwig MARKUS, Odo MEYER, Wilfried NIEHAUS, Andreas RAHENBROCK, Vitali SCHWAMM, Mario URRA-SACO, Kai-Uwe WEGNER
supervised by A. Ruckelshausen, A. Linz); Field Robot Event 2006, Hohenheim

Acknowledgment

For the good collaboration and active support i like to thank Prof. Dr. rer. nat. Arno Ruckelshausen, M.Sc. Dipl.-Ing (FH) Marius Thiel and Dipl.-Ing. (FH) / BEng Andreas Linz as well as the employees of the research center ISYS (Intelligent Sensor Systems) Ralph Kose, Wilfried Niehaus, Kai-Uwe Wegner, Christoph Kronsbein, Frank Fender and the whole team AMAIZEING.

In addition to these people my thank goes to all of the generous sponsors:

AMAZONEN-Werke H.Dreyer, Hasbergen-Gaste, Germany

Fraunhofer Gesellschaft, München, Germany
Institut, *Intelligente Analyse- und Informationssysteme*

Glyn Jones GmbH, Idstein, Germany

Wheels of Corntune

T. Maksimow[1], J. Hölttä[1], J. Junkkala[2], P. Koskela[1], E.J. Lämsä[1], M. Posio[2]
T. Oksanen[1] (advisor), J. Tiusanen[2] (advisor)
Helsinki University of Technology, Automation Technology Laboratory[1]
University of Helsinki, Agrotechnology Department[2]

Abstract

Wheels of Corntune was built as a student project for the Field Robot Event 2007 at Wageningen. Robot's main goal was to navigate autonomously between straight and curved rows of maize and turn at the headland to enter the next row. The robot's chassis is based on Demeter, a competitor at the Field Robot Event 2006, fitted with new sensors and improved electronics. The robot is controlled by an onboard laptop that is connected to two microcontrollers. The microcontrollers transfer sensor information to the laptop and control the servos and motors. The operation logic and control design is built with Simulink. In addition to the sensors the robot has a camera that is used in row and weed detection. In the competition the robot finished fifth. In this document the hardware and software implementations of the robot and technologies used are explained.

1 Introduction

The field robot Wheels of Corntune is a result of a student project of four students from the Helsinki University of Technology and two from the University of Helsinki. The project began in October 2006 and we started from where the Demeter team left off after the Field Robot Event 2006. However we redesigned the software and made modifications to chassis. We also added new sensors and built a docking station for the freestyle session.

During the course we experienced a lot of electronic and mechanical problems that delayed the actual testing of the robot until the final stages. Lack of a proper testing environment also made things worse (*Figure 1-1*). Nevertheless we project was very educational in terms of technology and teamwork. In the following we will first discuss the hardware and secondly the software and finally we present some conclusions.

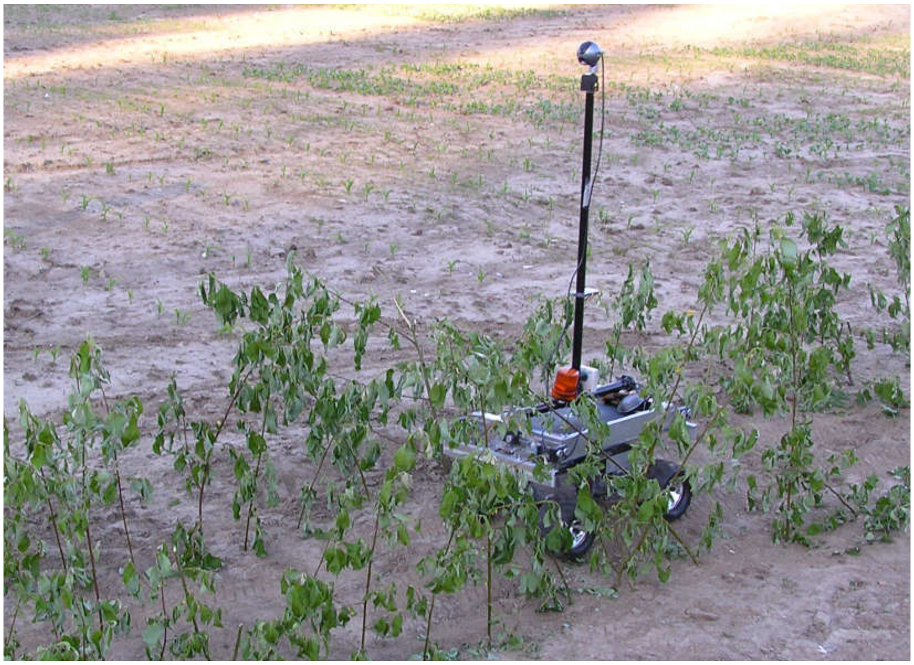


Figure 1-1: Test field in Finland built from birch branches

2 Hardware

2.1 Robot's chassis

The robot's chassis was built by the Demeter team for the Field Robot Event 2006. Their initial plan was to build a chassis that would be reliable, low cost, low power consumption and would have good off-road properties and good accuracy and ability to respond to instructions. They wanted to build and design the chassis from scratch because only then could they achieve the desired results. [3]

We improved the chassis by adding an aluminium framework for the new bumpers which also gave us more room for the electronic components in the middle of the robot. The bumpers were built for our freestyle session. We also repaired the suspension of the front axel. For the weeding task we built a harrowing system for the golf balls.

2.2 Electronics and sensors

The robot has quite a lot of electrical devices most of which are from Demeter: for computing: onboard laptop and two microcontroller boards, sensors: webcam, five ultrasonic distance sensors, actuators: two DC motors, three servos, two for turning the front and rear wheels and one for turning the webcam.

We added two infrared distance sensors to the sides of the chassis, three infrared receivers to the front for our freestyle and two servos for lowering and lifting up the rakes of the weed-killer rake system.

The robot has two Mavric IIB microcontroller boards, which are equipped with AVR Atmega 128 microcontrollers. In Demeter robot there was one Mavric and one PIC, and as it was not possible to extend the number of IO's in PIC, the old PIC was replaced with another Mavric and the whole program was reprogrammed. The first board is for collecting sensor data and the second is for controlling DC motors and servos. Both controllers are connected to the onboard laptop via serial or USB bus. The laptop does all the high level data processing and the microcontrollers are mainly IO devices.

We had to replace the old motor controllers and redo the wiring for the entire robot due to strange electrical problems. We had to completely disassemble the robot to locate the problem (*Figure 2-1*). The main culprit turned out to be the old RC-car motor controllers. We made two new H-bridge type motor controllers, which were connected to the other microcontroller and driven by PWM signal. Servos were connected directly to the PWM outputs of the controllers.

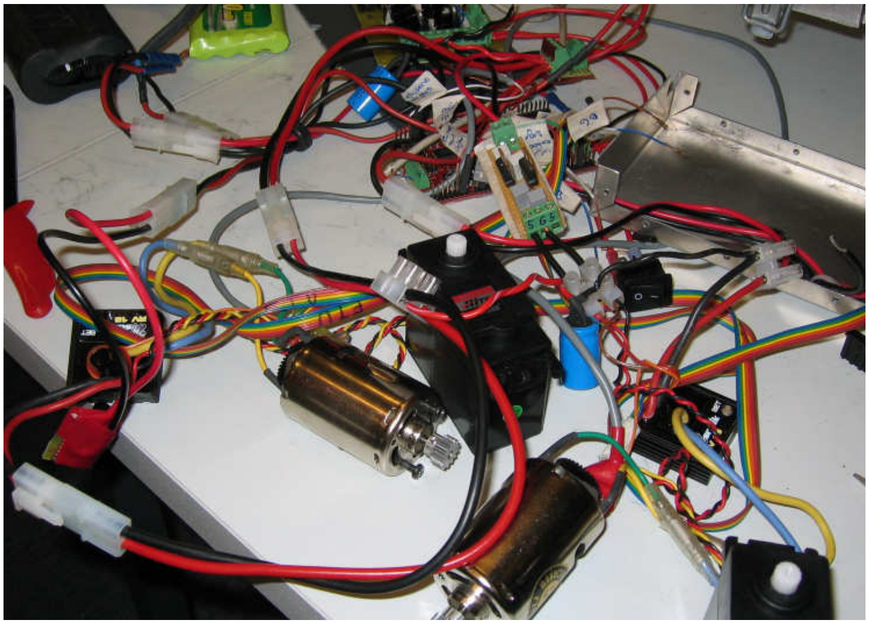


Figure 2-1: Robot electronics attached one by one to find the problems

Ultrasonic sensors and compass were connected to I2C bus and other sensors were connected directly to the I/O pins of the controller. The robot has a top part that can be detached from the chassis and most of the sensors and the camera and rake servos were mounted to the top part. For this reason we made also an adapter circuit board, which connects numerous I/O and PWM signals from microcontroller board to a single flat cable. The upper part has smaller adapter that connects the flat cable to the wires of sensors and servos.

2.3 Docking station

As a part of the freestyle session we want to demonstrate docking to a station (*Figure 2-2*). After completing a task in the field robots need to recharge. It is easy to navigate to three meter proximity of the station with a cheap GPS, but more accurate local navigation needs to be done with other methods. We have used camera and infrared sensors. When the robot locates the hoods with the camera or sees the infrared beam, it starts to drive towards the station.

The station is made of Finnish birch plywood, second-hand drainpipe and two round lamp hoods (red and blue). There are two copper cables, with spring suspension, in the station to make a connection with the surfaces in the front bumper of the robot for charging the batteries.

Throughout the procedure the remote control interface communicates with the robot to synchronize and control the operation. The remote control interface drives a USB-port attached data acquisition device. The DAQ-device is set to measure current in the copper cables and therefore it can sense when the robot is attached to the docking station, the resistance of closed circuit is known. The interface drives the relay to output 12 V from the station's battery. The orange flashing light on the top of the robot starts to flash in order to demonstrate charging the batteries, power for that is coming from the station. When the charging is done the robot leaves the dock and the relay cuts down the power. After recharging the robot can return to the field with new orders.

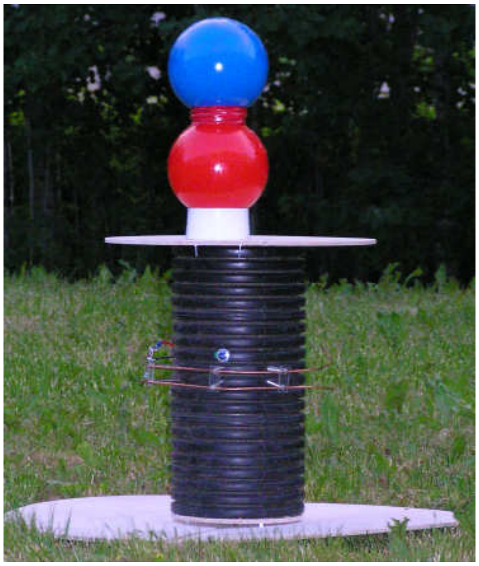


Figure 2-2: The Docking station

3 Software

The robot's software was built using several tools for different purposes. The machine vision was built using C++ and OpenCV [1] image processing libraries. Robot's operation logic and control design was done with Matlab/Simulink. The main program is written with C# and the Remote Graphical User Interface (RGUI) is implemented with National Instruments LabView 8.

3.1 Machine vision

A Logitech web camera was situated on top of the mast looking slightly downward. Image processing was done with EGRBI transform (Excessive Green, Red-Blue, Intensity) using OpenCV image processing functions. Also HSV transform was tried, but EGRBI was found to be better and easier to tune than HSV. At first algorithm takes a RGB image and splits it to 3 images: Red image, Green image and Blue image. Each image is used to make 3 new images: Excessive Green image, Red-Blue image and Intensity image.

$$\underset{[320*240*3]}{\begin{bmatrix} EG \\ RB \\ I \end{bmatrix}} = \underset{[320*240*3]}{\begin{bmatrix} R \\ G \\ B \end{bmatrix}} * \underset{[3*3]}{mask} \tag{1}$$

$$EG=[2G-R-B]$$

$$I=\frac{1}{\sqrt{3}}[R\ G\ B]$$

$$RB=EG\times I$$

We extended the algorithm a bit, we modified the basic equations (1) to support the detection of any color by clicking the desired pixel in the camera image. This way any color can be changed by equation 2 to Excessive Color (EC).

$$EC=\frac{1}{2*(R^2-(B+G)*R+B^2-B*G+G^2)}\begin{bmatrix} 2R-B-G \\ 2G-R-B \\ 2B-R-G \end{bmatrix}\quad , \tag{2}$$

where I is same as on EGRBI transform and the third component will be cross product, EC × I. Let's call this ECCI transform later on. After the transform 3 binary images are made from these 3 new images with thresholds and these 3 binary images are merged to one binary image.

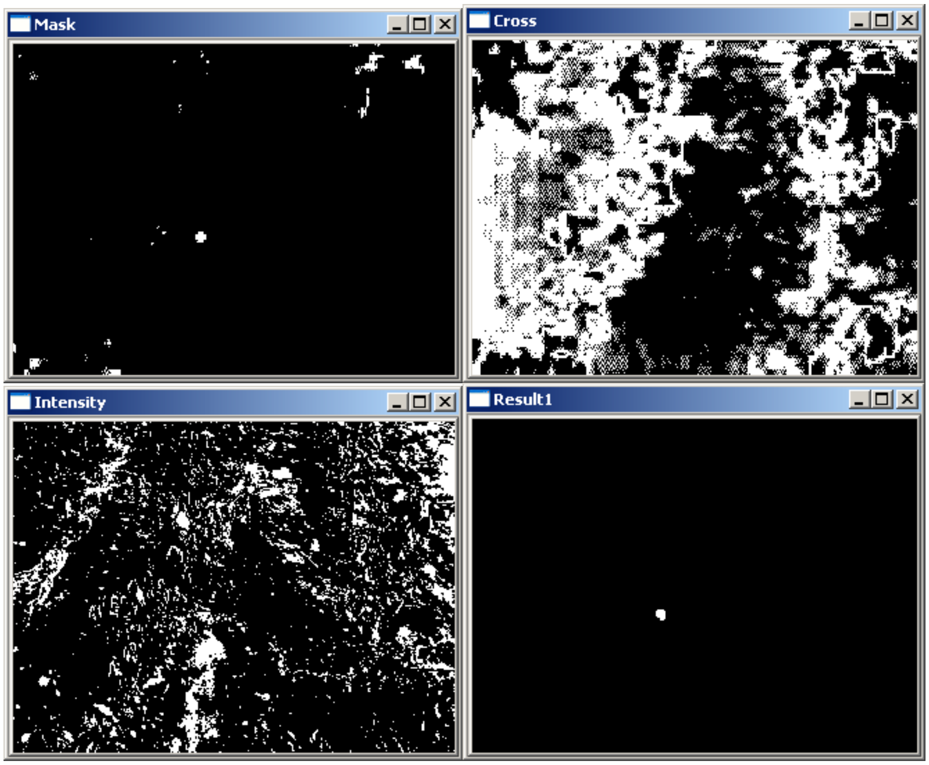


Figure 3-1: Images created by the ECCI transform and the result image for weed detection

3.1.1 Row detection

We used two methods for row detection, Hough transform and Histogram row detector, but in the end we found the latter to more reliable.

In the first method rows can be search with Hough transform based on the binary image. Place of robot can be estimated from rows with simple trigonometric calculations (*Figure 3-2*). This is the same method that Demeter team used. [3]

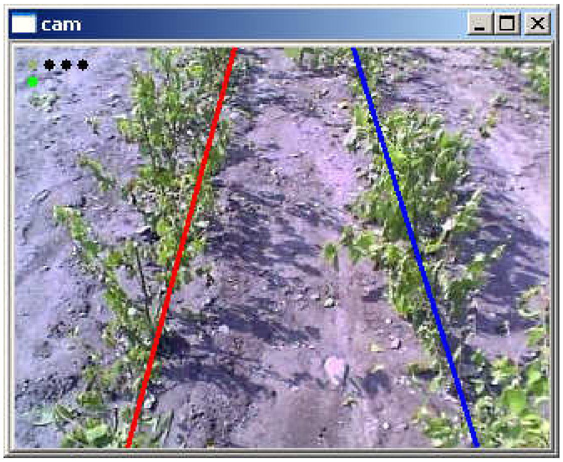


Figure 3-2: Hough transform for detecting lines

The second method is based on an ASAE Publication *Automatic Guidance System With Crop Row Sensor* by H. Okamoto et al [2]. First it does a perspective transform to the binary image and then it slices the picture to six slices and calculates their histograms. From these slices peaks are detected, shifted and combined several times. From the combined histogram the highest peaks determine the necessary shift. Using trigonometry the robot's lateral displacement and angular error can be resolved (*Figure 3-3*).

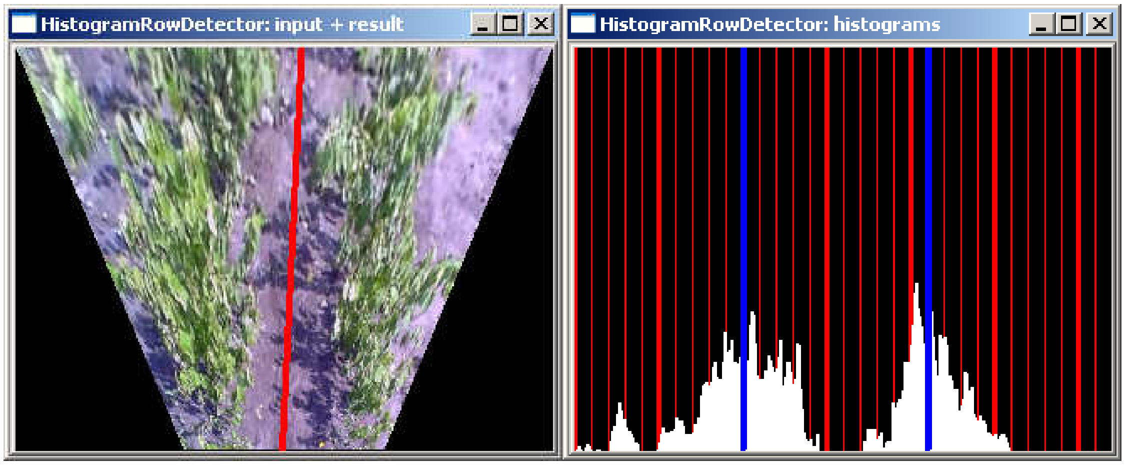

HistogramRowDetector: input + result
HistogramRowDetector: histograms

Figure 3-3: Histogram row detector in action: result image, combined histograms and detected peaks

3.1.2 Weed detection

Weeds were not only detected from the binary image (*Figure 3-1*), but they were also tracked. This means that separate objects that are visible in the result image are detected one by one and then marked into a list where their coordinates are maintained. As the balls move the list is updated and new balls are being searched for. The detection of new balls was implemented with OpenCV function GoodFeaturesToTrack and the updating of their coordinates was implemented with OpenCV function

CalcOpticalFlowPyrLK. When a tracked balls y-coordinate is above a certain threshold, it is removed from tracking and an event to lower the harrow is given. This implementation worked well in our tests but for some reason in the competition the harrows did not lower at all. Perhaps the balls were so close to the stalks that they were not visible in camera because of the leaves.

3.1.3 Dock detection

Dock detection was made by detecting red circle and blue circle from images. Firstly two binary images are created, one with for the red balls and another for the blue. To avoid detection of wrong shapes and sizes, we used OpenCv function HoughCircles to detect the mast of the station. From both of the images circles were searched separately. All of the detected circles were matched with each other to see if there is circle in both images that are of the right size and close enough to each other. If the dock was seen, the direction and distance of dock can be estimated with simple algorithms. This method worked pretty well but was a bit slow and was not very robust.

3.2 Sensor Fusion and navigation

In addition to the camera, the robot has 4 ultrasonic and 2 IR-sensors for navigating between maize rows, a gyroscope and compass for turning at the headland and three infrared receivers at the front for docking.

3.2.1 Navigating between rows

The primary sensors used for navigation are the four ultrasonic sensors located on the upper hull at each corner of the robot. Two IR-sensors were added this year to the lower hull to increase the accuracy of driving in the middle of the row.

The Demeter team used two PID-controllers for controlling the lateral position and angle of the robot [3]. We decided to use a bit more direct approach and control the front and rear of the robot directly with PID-controllers. This gave us the possibility to move more smoothly between the rows by limiting the steering of the rear end of the robot and using the front steering servo primarily for controlling the direction of the robot.

We have two different methods for using the ultrasonic and IR-sensors for navigating between the rows. The first approach is similar than used by the Demeter team, the position of the robot is simulated backwards for a while using robot's kinematic model and the measurements are projected into cartesian space and lines are fitted to them using least square method on both sides separately. With simulator this worked well but at first we couldn't make it work with the robot and the reason was that the odometry measurements were not correctly calibrated. After calibration this method seemed to work well. One thing that we didn't have time to do, was a dynamic modeling for steering servo behaviour, wheels do not turn instantly, and the real steering angle should be estimated.

The other method to estimate position in a row was pure static geometric calculation, the ultrasonic sensor data was used directly to measure the front and rear lateral errors. With this method the robot started to drive logically but ended up being a more or less stable oscillator. With the limited time we had to test the robot in an actual maize field we were unable to tune either ultrasonic method to perfection but we noticed that they seemed to work quite well together.

The fusion of all the sensors for driving between the rows is done with a simple weighted average of the two ultrasonic sensor methods (37.5% + 37.5%) and the camera (25%).

3.2.2 Row end detection

Row end detection is made to be very robust and reliable, but still quite fast at the real row end. The ultrasonic sensors seemed to be very reliable from the start, they worked 100% of the time indoors with solid walls and even flowerpot rows, but when we moved outdoors to our makeshift birch branch testing field, we noticed that the ultrasonic sensors weren't reliable enough alone because we got some random row end detections in the middle of the rows.

We tried adding the IR-sensors to the row end logic and managed to get it more reliable results while on the rows but it had a negative effect on the actual row end detection. The IR-sensors often give false signals even when driving on an empty field so we decided not to use them. The camera information turned out to be the ideal pair for the ultrasonic sensors since the actual row end detection was extremely reliable with the ultrasonic sensors alone and with the camera data we could remove the false row end detections when the robot is actually still between the rows.

3.2.3 Headland turn

The headland turn is based on odometry and measured angle. The compass and gyroscope are both quite reliable alone, but neither is perfect alone. The compass gives some random spikes and isn't accurate if the robot is tilted and the integrated gyroscope angle drifts even though the bias is calibrated when the robot is started. A Kalman filter was implemented to combine these two. The filter was heavily simplified because we only needed to estimate one scalar variable, the angle.

Prediction:
$X_{pri} = X_{pos} + \omega * dt$
$P = P + Q * dt$
where ω and Q are the gyro measurement and its variance

Update:
$e = z - X_{pri}$
$S = P + R$
$K = P/S$
$P = P - K * S * K$
$X_{pos} = X_{pri} + K * e$
where z and R are the compass measurement and its variance.

These equations are the same as used in the Personal Navigation System by Saarinen et al. [5]
Unfortunately the last two nights of coding and testing somehow broke the filter so it was abandoned and the compass data was used directly and we found it to be quite accurate outdoors when the compass is properly calibrated.
After the competition, a thorough analysis of the logs collected during the competition indicated that the gyro may have been malfunctioning because its bias drifts so heavily in many of the logs that the gyroscope data would’ve been unusable anyway. The gyroscope was however in earlier tests quite reliable and it could be used for smoothing random spikes from the compass data because of its good short term accuracy.

3.2.4 Navigating to the docking station

The robot has an array of three IR-receivers on top of the front buffer. The receivers are mounted so that the angle between them is 45 degrees (left, front, right). This allowed us to approximate the direction of the dock with just one IR-transmitter by calculating the vector sum of the voltages given by the receivers (length being 1/voltage). We also used this to roughly estimate the distance to the dock. Since the receivers were mounted at the front of the robot it could only find the dock with them if it approached so that it crossed the cone of the IR-transmitter at some point.

By using the camera and its turning servo we managed to increase the robots field of view and the dock could be found from a much bigger area around it.

The main problem with the camera was the lack of robustness. The Hough circles were sometimes lost even when the robot wasn't moving and it could clearly still see both of the colors. The situation gets much worse when the robot starts moving because the camera is mounted on the mast so high. Even when moving on a flat floor the robot often wobbles enough to cause the camera to swing so much to the side that it loses sight of the dock.

The IR-receivers worked well indoors when we were testing during the winter but they became completely useless when we started testing outside, even when they were not directly exposed to sunlight. The camera however worked well enough alone for the robot to drive to the dock, but not very smoothly.

3.3 Controller & operation logic

The control algorithms of the robot were implemented in Matlab/Simulink. Controllers were developed using Simulink's blocks and embedded functions. We used Matlab's Real time workshop to generate C-code from Simulink models. [4] The generated source code was then used in the main program, which runs in the onboard laptop. In this way the need of writing code by hand was greatly decreased.

The controllers used while driving between the rows are two PID-controllers with anti-windup and filtered derivative. The front and rear errors given by the sensor fusion block is used to control front and rear wheels separately.

Operation logic of the robot was developed using Simulink's Stateflow extension (*Figure 3-4*). It can be used to create state machines and use them in connection with normal Simulink models. Our state machine consisted of states like "between rows", "at the end of row", "turning" etc. Turnings were implemented completely in Stateflow.

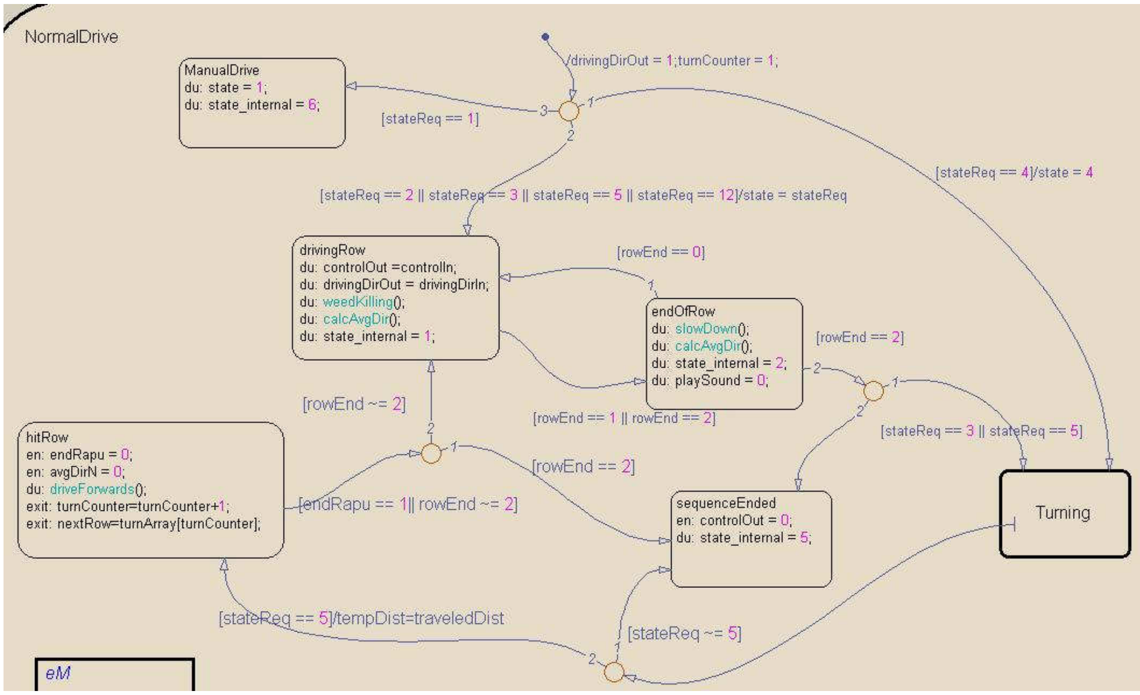


Figure 3-4: Sample of robot's state machine chart

Another advantage of using Simulink's code generation was that the models could be simulated using Matlab. We had a self made simulator which basically simulates the ultrasonic sensors and robot's movement in test tracks (*Figure 3-5*). The simulator was of great help especially in testing stateflow logic and overall functionality of the whole model. We did also some PID parameter approximation with it, but fine tuning of course had to be done in a field. The body of simulator code (laying maizes, kinematic model and simulating ultrasonic sensors) was got from last years team and we extended it to support docking, infrared sensors and plenty of small thing were improved.

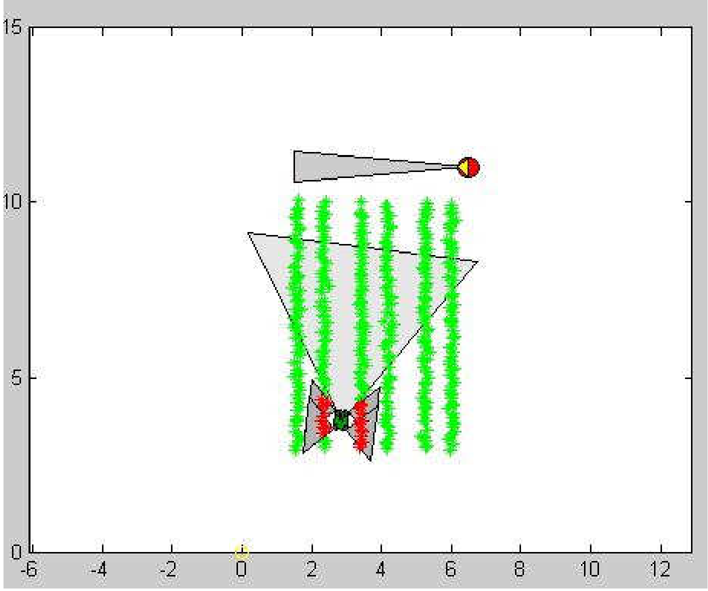


Figure 3-5: The simulator in action

3.4 Remote GUI

The robot has a remote graphical user interface on another laptop (*Figure 3-6*). The laptop uses its own WLAN in order to communicate with robot's laptop. The laptops communicate by sending each other data strings in UDP packages. The interface is made with National Instruments LabView 8. The main purpose is to be able set the robot to do specific tasks, drive manually with the joystick or to stop the robot in an emergency. Robots sensor data can also be viewed from the laptop and different variables can be adjusted.

The remote GUI communicates with robot by sending commands as strings. The strings are orders to change the state of the robot, for example "Drive automatically" or "Stop". These strings are then parsed by the robot and passed to the state machine of the robot. The robot also sends all information as strings to the robot. Different values are separated with a semicolon for splitting the string.

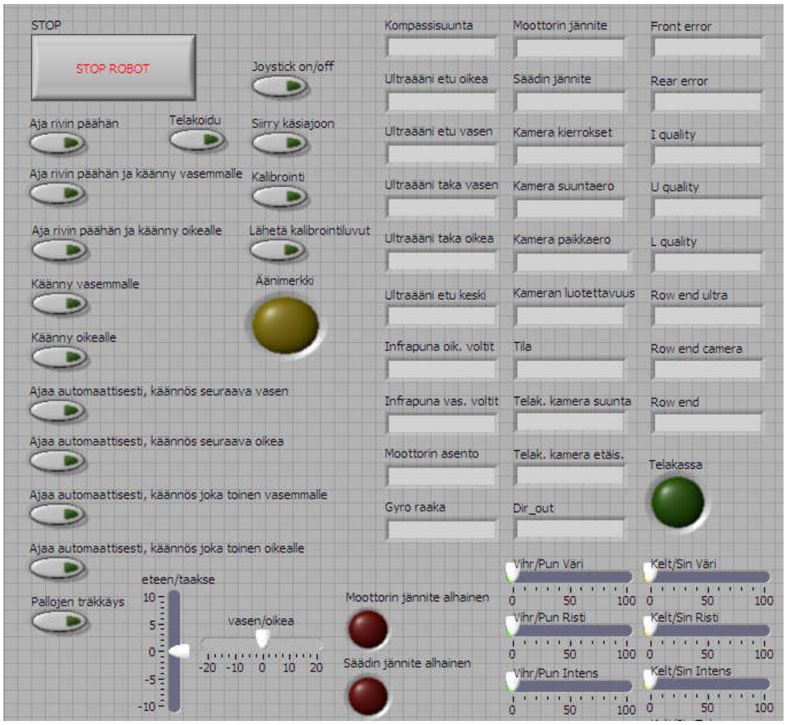


Figure 3-6: Remote Graphical User Interface

3.5 Main program

The main program is written with C# and is a multithreaded program. Although it synchronizes the operation of the camera, the robot controller and the Remote GUI interface threads, it mostly acts as a junction between the other components. It passes sensor information to the controller and pushes the controls to the robots microcontrollers.

In the main program it is possible to load and save parameters that are used. For instance the camera and odometry can be precalibrated and then the calibration values can be saved. Parameters are stored in an XML serializable class.

The main program also writes a log of every input and output the microcontrollers and the robot controller have. The log can then be analyzed later back in the lab.

4 Conclusions

Starting from Demeter's chassis and other components that seemed to work somewhat was a mistake for us. We should have built the robot from scratch. Most of the problems we had during this project were due to the old hardware.

Concerning the software, the user unfriendly interface of Simulink also caused some grief, especially when the models started to get complicated and contained several self-made library blocks. Matlab itself is a great tool for data analysis but it is not well suited for a long term software development project. The simulator built with Simulink was a nice tool for validating the basic logic of the controller, but we relied too much on it and did not start testing with the robot early enough. Also the simulator's model needs to be very accurate in order be useful in real testing. Especially if model based control is used, it is very important that the model used is as accurate as possible.

The camera worked well after we decided to change the Hough transformation to the histogram row detector. The Hough transform is not robust enough for the environment. For detection of the weeds the camera should have been lower. This we did not realise before seeing the real corn field as our test field was not that good a representation (*Figure 1-1*).

In long term software development projects proper version management is very important. This is something we learned in the final coding sessions before the competition.

On the whole the performance of the robot was still decent even though many things went wrong.

Acknowledgments

Thanks to our sponsors:

Ford koneviesti

References

[1] OpenCV, Open Source Computer Vision Library, Intel Corporation, http://opencvlibrary.sourceforge.net/

[2] Okamoto, H., Hamada, K., Kataoka, T., Terawaki, M. and Hata, S. (2002). "Automatic Guidance System With Crop Row Sensor" Automation Technology for Off-Road Equipment, Proceedings of the July 26-27, 2002 Conference (Chicago, Illinois, USA), ASAE Publication Number 701P0502, Pp. 307-316

[3] Telama, M., Turtiainen, J., Viinanen, P., Kostamo, J., Mussalo, V., Virtanen, T., Oksanen, T. and Tiusanen, J. (2005) Field Robot Event 2006 Proceedings, http://automation.tkk.fi/FieldRobot2006/FieldRobot2006?action=download&upname=FRE2006_demeter.pdf, (accessed 30.6.2007)

[4] The MathWorks, Inc, Real-Time workshop http://www.mathworks.com/products/rtw/

[5] J.Saarinen, J.Suomela, S.Heikkilä, M.Elomaa and A. Halme, ”Personal Navigation System”, Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan

WURking - An autonomous robot for agriculture applications

J.W. Hofstee, R. Jansen, A.T. Nieuwenhuizen, M.H.A.M. Govers, S.K. Blaauw, L.G. Van Willigenburg, J.D. Stigter
Farm Technology & Systems and Control Group, Wageningen University, Bornsesteeg 59, 6708 PD Wageningen, The Netherlands

J.D. Stigter; L.G. van Willigenburg, T. Bakker, C.J. van Asselt, G. van Straten, S.L. Speetjens
Systems and Control Group, Wageningen University, Bornsesteeg 59, 6708 PD Wageningen, The Netherlands

Abstract

Autonomous robots for agricultural practices will become reality soon. These mobile robots could take over regular task such as weeding. Other new applications can be plant specific spraying and the release of infochemicals for attracting predators of pests. Therefore the Farm Technology Group and the Systems and Control Group of Wageningen University decided to develop a robot for agriculture applications. This field robot has to navigate and steer autonomously on the field. This task is performed using a BasicATOM micro-controller coupled to a PC based system programmed mainly using Labview. Data from ultrasonic and infrared sensors, a camera, a gyroscope and a guidance rail are used to control the robot. A kinematic vehicle model in combination with a feedback linearizing controller are used to calculate the setpoints for the motor controllers, based on the readings of different sensors.

1 Introduction

Robotic systems for agricultural applications are not new. Tillet *et al.* (1997) already discussed a robot for plant scale husbandry. Robots in agriculture can be used for several applications. Examples are robotic weed control (Astrand and Baerveldt, 2002), mapping in-field variability (Godwin and Miller, 2003) or detection of volunteer potatoes (Evert *et al*., 2006).

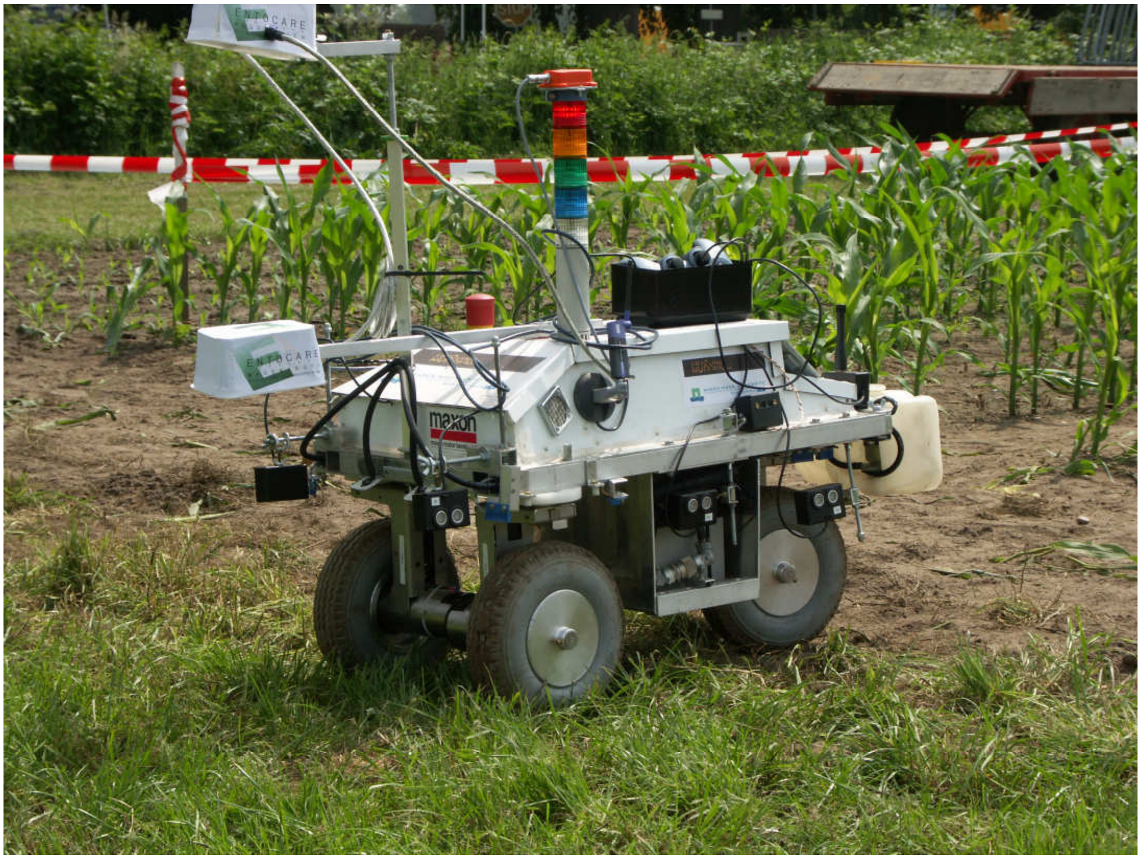
Wageningen University started in 2003 with the organisation of the Wageningen UR Field Robot Event (Van Straten, 2004). An important reason is to stimulate the further development of robots in agriculture, The organisation of the event in 2006 by the University of Hohenheim made initiated the joint development of a small robot for field applications by the the System and Control Group and the Farm Technology Group. There were two objectives: (1) to have a robot to participate in the 4th Field Robot in Germany and (2) to have universal platform to be used for further research and education on small robots for agricultural applications. The robot had to be able to fulfil the following tasks: (1) drive through maize rows and count dandelions (yellow golf balls), (2) detect holes in a grass field of 10 x 10 m, (3) detect a corner flag and drive to it, (4) drive as fast as possible through maize rows, and (5) give a good performance in the free style session.

2 The fieldrobot

2.1 Overall design

The fieldrobot (Figure 7) is based on an aluminium frame and three independent wheel units. Two wheel units are placed in the front and one wheel unit in the rear of the frame. Each wheel unit is equipped with two DC motors, one motor for steering the unit and one motor for driving the wheel. The robot is equipped with six ultrasonic and two infrared sensors for row detection. Furthermore the robot is able to follow a row based on a guidance rail with two potentiometers. Additionally a camera is attached to the robot for detection of golf balls, lines, and the corner flag. A gyroscope is used to determine the orientation of the robot.

Figure 7: WURking.



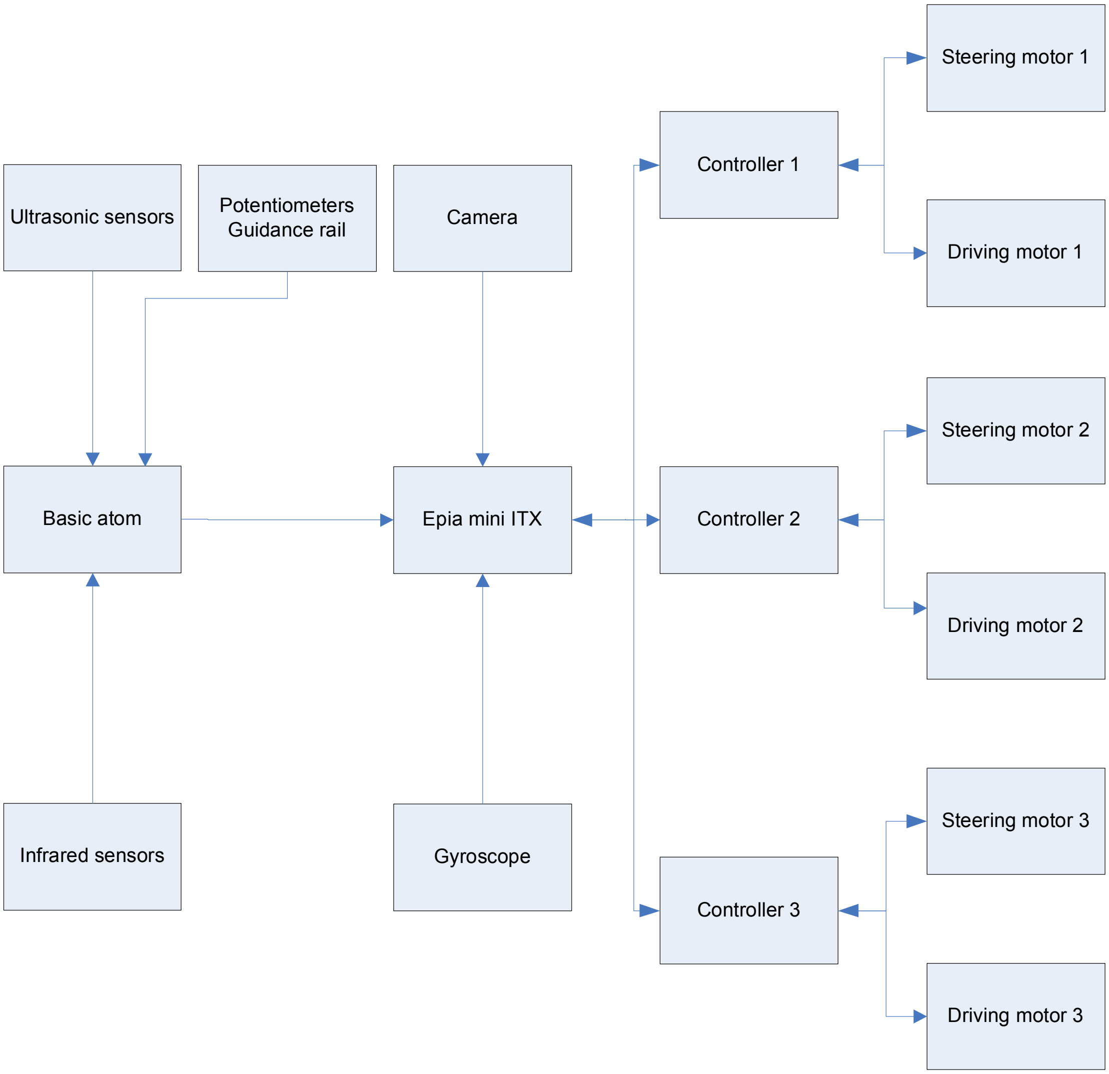


Figure 8: Overview of the different components of WURking.

The electronic signals from the potentiometers, the ultrasonic and the infrared sensors are processed by a microcontroller. These signals are, together with camera and gyroscope signals, processed by high-level software. Part of the high level software is a kinematic vehicle model. This vehicle model calculates the control signal for each motor based on the output signals of all different sensors.
An overview of components of the data acquisition and the motor control system is given in Figure 8.

2.2 Platform

The platform consists of an aluminium frame, a cover to protect the electronics, a battery pack, several sensors and three independent wheel units. The wheelbase is 500 mm and the track width between the front wheels is 320 mm. The total width of the platform is 400 mm and the total length is 700 mm. Each wheel unit is able to steer left (+135^o) and right (-135^o). The individual wheel units are independently driven by a DC motor connected to a planetary gearhead. A battery pack is attached to power the motors, the personal computer, the controllers and the sensors. This battery pack is located in between the wheel units to realize a low centre of gravitation. The clearance of the platform is 120 mm. The weight of the platform including battery pack is 39 kg.

2.2.1 Wheel units

The wheel units (Figure 9) are designed in cooperation with the Mechatronics Department of the Kverneland Group in Nieuw Vennep, The Netherlands. Each wheel unit is equipped with two DC motors. The wheel is powered by a single 150 W motor at 24 volts (Maxon Precision Motors, brushed DC motor, model RE40). The maximal torque delivered is 181 mNm at 7580 rpm. This motor is connected to a planetary gearhead with a reduction of 15:1 and a maximum efficiency of 83%. (Maxon Precion Motors, model GP52C). The drive motor is equipped with an encoder to measure the speed in counts per turn (Maxon Precision Motors, encoder, model HEDS 5540). Steering is realised by a 20 W motor at 24 volts (Maxon Precision Motors, DC motor, model RE25) connected to a planetary gearhead with a reduction of 66:1 and a maximum efficiency of 70% (Maxon Precision Motors, model GP32C). The wheel units are equipped with conventional tubed tyres (Ø 250 mm, 80 mm). During the FRE 2006 the tubes were inflated to approximately 3 bar for optimal traction. The maximum steering velocity of the unit is approximately 115 deg/sec. The weight of the wheel unit including the motors is 4.1 kg.

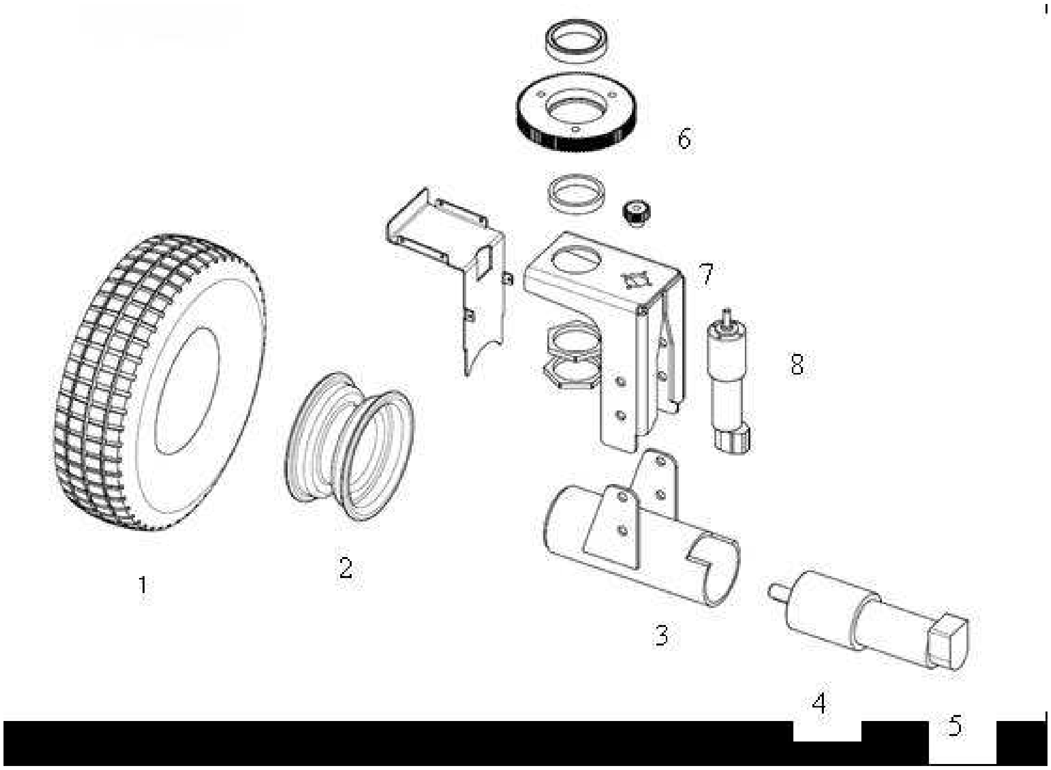


Figure 9: Details of the wheel unit: (1) Tyre, (2) wheel rim, (3) frame for drive motor, (4) drive motor, (5) encoder, (6) steer transmission, (7) frame for steering motor and (8) steer motor (Drawing: Kverneland Group).

2.2.2 Motor controllers

Each wheel unit is controlled by a Motion Mind DC motor controller. Each controller is capable of controlling two brushed motors. In this case one motor for driving and one for steering.

2.2.3 Battery pack

The platform is equipped with a battery pack containing three batteries. Two 12V, 7Ah batteries are used to power the driving and steering motors. Additionally there is one 12V, 12 Ah battery to power the PC platform, controllers and the sensors. The total weight of this battery pack is 10 kg. This battery pack is mounted to the platform in such a way that it can be easily exchanged with a spare battery pack for continuous operation.

2.3 Sensors

2.3.1 Ultrasonic

The robot has six Devantech SRF08 ultrasonic sensors, three at each side. These sensors measure the distance from the robot to the crop row. The range of the ultrasonic sensors is 3 cm to 6 m; the frequency is 40 kHz. The ultrasonic sensors are connected to the I^2C bus of the BasicATOM microcontroller.

2.3.2 Infrared

The robot has two Sharp GP2D12 infrared sensors. These are until now not used.

2.3.3 Camera

The camera is a Unibrain Fire-i firewire camera. The camera has a ¼" CCD (659 x 494 pixels); the pixel size is 5.6 µm in both horizontal and vertical direction. The frame rate is up to 30 frames per second (uncompressed VGA picture).

2.3.4 Gyroscope

The robot is also equipped with a XSens MT9-B gyroscope. This gyroscope has a 3D compass, 3D accelerometer and 3D gyro's and yields by integration very precise values for yaw, roll, and pitch. The angular resolution is 0.05º, the static accuracy is <1º, and the dynamic accuracy is 3º RMS (XSens, 2006).

2.4 Controllers

2.4.1 Basic ATOM

Part of the data acquisition is realised with a BasicATOM40 microcontroller. Inputs for the microcontroller are the two potentiometers, the two infrared sensors and the six ultrasonic sensors. The microcontroller processes the raw sensor signals and creates a message with the calibrated values which is send to high level control program.

2.4.2 PC

WURking has a VIA EPIA SP13000 PC for high level control. This is a low power compact motherboard with built in CPU, graphics, audio and network. The PC has 512 MB RAM and a 40GB hard disk with a Windows XP operating system. The PC also has a WiFi connection for remote control and monitoring purposes.

2.5 Robot position, orientation and end of row detection

A VI (Virtual Instrument) was designed to compute the position and orientation of the robot relative to the row from six ultrasonic sensor measurements. The VI basically employed a linear regression technique in combination with information about the row width and robot dimensions. Also it provided a signal when one or several sensor measurements are inconsistent with these dimensions. This signal was used for the end of row detection.

2.6 Vehicle model and on-line control by means of feedback linearization

Navigation and control design issues were addressed in an advanced model-based design using a kinematic mathematical model of the three-wheel vehicle. The kinematic model is a dynamic state-space model as in (Campion et al., 1996) that is integrated using standard numerical integration techniques. In this way the next state, that is the next position and orientation of the robot is calculated, knowing the current control inputs being the wheel velocities and wheel angles. A so-called Kalman filter combines these computations with on-line measurements obtained from the ultrasonic sensors to estimate on-line the future state, that is the next position and orientation of the robot relative to the row. These estimates together with the state-space model are used to compute on-line the next control values, that is the next values of the wheel angles and velocities. The computation is performed by applying a feedback linearization scheme (Campion et al., 1996). Application of this scheme reduces the controller design to a simple linear controller design that is performed by means of pole placement. The state estimates, generated by the Kalman filter, can also be used for higher level decision making such as what to do if the end of the row is reached.

2.7 End of row detection and the headland turn

Turning by means of feedback linearization was unsuccessful due to the total absence of feedback from the ultrasonic sensors. Therefore a very simple proportional controller was used for turning at the headland. It used the X-sense absolute angle measurement for feedback and benefited from the fact that setting the three robot wheels at appropriate angles allows the robot to fully turn around its midpoint. The output of the proportional controller was clipped to limit the angular velocity of the robot. Filtering the last couple of measurements taken from X-sense while driving through the previous row provided a good estimate of the orientation of the end of the previous row. Relative to this orientation the robot turned successfully into the next row using the clipped proportional controller. The turn was performed in three stages. First a 90 degree turn was made. Next the wheels were set straight to drive straight over a fixed distance coinciding with the row distance. This distance was measured by the wheel rotations and regulated also by a clipped proportional controller. Then a 90 degree turn was made again. The end of row detection was performed using the VI (Virtual Instrument) build around the six ultrasonic sensors. The VI provided a signal that indicates whether measurements occur that are incompatible with driving through the row. If this signal was received for a certain amount of time this indicated the end of row.

2.8 High level control

The high level control of the robot is realised by a LabVIEW program. The program consists of several processes that run independently from each other. Each process is represented by a VI (Virtual Instrument). and several sub VI’s and realises a specific task. There are VI’s for initialisation, motor control, the kinematic vehicle model, camera control, communication with the BasicATOM, communication with the gyroscope, and sensor fusion, A state machine controls the activation and de-activation of processes. The VI’s exchange data with each other via global variables.

3 Field tests

Field tests indicated that driving through straight and curved rows by means of feedback linearization performed well since no plants were damaged while the driving speed was reasonable up to 0.75 m/sec. In roughly 75% of the times the headland turn was successful meaning that no plants were damaged and no human interference was necessary. Recognising golf balls was successful only when the light is sufficiently constant. Our camera and image tools were not sophisticated enough to do anything about this. Our "water canons" meant to hit the golf balls responded well but often missed since they could not be steered. Moreover the moment of golf ball detection by the camera (if it was detected at all) was rather unpredictable. Still, during the competition, the robot managed to hit some.

4 Conclusions and discussion: kiss Bill, kill Bill

Last year we had the ideas but we greatly underestimated the time needed to properly implement them. Compared to last year many technical shortcomings and programming errors have been overcome and remedied. Moreover to deal with the task of detecting and "killing" golf balls the robot was equipped with "water canons". The basic idea of how to control the robot has not changed although we abandoned the use of one important particular sensor to detect the position and orientation of the robot when driving through a row. This detection is now performed entirely by six ultrasonic sensors coupled to a VI. The VI, the Kalman filter and the feedback linearizing controller worked well. It was great fun being the captain of the "Wurking team" where everybody contributed in good spirit and often with success. My scepticism concerning the heavy weight and large dimensions of the robot, compared to those of a row, turned out to be largely unjustified thanks to our clever engineering and Labview programming and our intelligent VI and feedback linearizing controller. All this lead to an overall fourth position out of ten. We were especially happy that during the contest we were able to reproduce our best results found during several tests. The last position of last year has been forgotten …

Compared to the winners we suffer from one major drawback. Some of them had more intelligent sensors but mainly they did not suffer from Bill Gates who interfered at irregular times with our so called "real time control system" by suspending it for several seconds! But as our main programmer who we called "Magic" put it: "I would never have started the job without Bill, because of all the benefits Bill and Labview offer when developing software cheaply and quickly". On the other hand the hardware of the winners contained no PC's, so no Bill Gates. They had to program at a lower level, with more advanced and expensive equipment and software, which is mostly available at the institutes where they work. Also none of the winners used a model! It seems that using rather simple steering strategies is already sufficient to drive properly and fast through curved rows. For us model based control was a must to keep the project interesting. But looking closely into the control problem we also discovered that simple steering strategies can be very effective as well. The headland turn of the winners was a bit of luck, as they told me, although they had a clever sensor installed for it, that the others did not have. Our headland turn strategy was also amazingly simple and rather successful.

In summary we very much enjoyed the event and the preparation. However we do feel that to continue joining the event we must be challenged with a task that does require sophistication and intelligence which would justify scientific publication. That would also justify additional budget and time spent on it. If the contest keeps its current character that suits very well educational purposes, as expressed by many of the competitors, I think we will remember this last robot event gracefully. Maybe an educational and a scientific competition should be organised in future robot events.

Acknowledgments

The authors would also like to acknowledge Mechatronics Department of the Kverneland in Nieuw Vennep for the cooperation in the design of the wheel units, XSens Motion Technologies for providing the gyroscope, Priva for some financial support and the colleagues of the other robots teams of Wageningen UR for the cooperation and discussions during the pre field robot event era...

References

Astrand, B. and Baerveldt, A. J. (2002). "An Agricultural Mobile Robot with Vision-Based Perception for Mechanical Weed Control." Autonomous Robots 13: 21-35.

Campion G., Bastin G., d'Andrea-Novel B., 1996, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots", IEEE Transactions on Automatic Control, 12, 1, 47-62.

Evert, F. van, Heijden, G. W. A. M. van der, Lotz, L. A. P., Polder, G., Lamaker, A., Jong, A. de, Kuyper, M. C., Groendijk, E. J. K., Neeteson, J. J. and Zalm, T. van der (2006). "A mobile field robot with vision-based detection of volunteer potato plants in a corn crop." Weed Technology (in press).

Godwin, R. J. and Miller, P. C. H. (2003). "A Review of the Technologies for Mapping Within-field Variability." Biosystems Engineering 84: 393-407.

Straten, G. van (2004). "Field Robot Event, Wageningen 5-6 June 2003." Computers and Electronics in Agriculture 42: 51-58.

Tillett, N. D., Hague, T. and Marchant, J. A. (1997). "A Robotic System for Plant-Scale Husbandry." Journal of Agricultural Engineering Research 69(2): 169-178.

XSens (2006) http://www.xsens.com/download/MT9_brochure.pdf (18-08-2006)