

AUTONOMOUS SERVICE ROBOTS FOR ORCHARDS AND VINEYARDS: 3D SIMULATION ENVIRONMENT OF MULTI SENSOR-BASED NAVIGATION AND APPLICATIONS

A. Linz, A. Ruckelshausen and E. Wunder

University of Applied Sciences Osnabrueck and Competence Center of Applied Agricultural Engineering COALA, Osnabrueck, Germany

J. Hertzberg

Osnabrueck University, Institute for Computer Science, Osnabrueck, Germany and DFKI-RIC Osnabrueck Branch, Osnabrueck, Germany

ABSTRACT

In order to fulfill economical as well as ecological boundary conditions information technologies and sensor are increasingly gaining importance in horticulture. In combination with the reduced availability of human workers automation technologies thus play a key role in the international competition in vinicultures and orchards and have the potential to reduce the costs as well as environmental impacts.

The authors are working in the fields of unmanned or remote controlled autonomous field robots, navigation, image-based sensor fusion as well as agricultural applications. In particular field robots have been applied for a few years in outdoor agricultural field applications. Within an interdisciplinary research group these technologies are transferred to robot applications in vineyards and orchards. The goal is the availability of an autonomous service robot, whereas first applications are site-specific plant protection (e.g. precise spraying), mulching and picking up fruit boxes. A first version of the robot with electrical drives and precise sprayers has already been developed. The applications, however, show a large range of field conditions which have to be considered for the vehicle application design. Thus the authors have developed a 3D simulation environment which allows the virtual test of the robot platform prior to its application. Moreover, the software algorithms can be directly transferred to the robot and thus allow iterative optimizations of the development process. The generation and first applications of the 3D simulation environment of multi sensor-based navigation and applications in vinicultures and orchard is the focus of this work.

Robot Operating System (ROS) has been chosen as software framework for integrating the autonomous vehicle, the sensors and the environment for navigation and application processes. ROS supplies the 3D simulation environment Gazebo using physical engines (e.g. ODE – Open Dynamic Engine)

in order to simulate the robots behavior as close as possible to reality. Moreover, the software tool Rviz is used for visualization of the sensor data (as for example) for optimization of navigation algorithms. Since the navigation in vine and fruit rows, the various applications as well as safety issues require sensor based solutions. The navigation itself is performed by image based sensors, since GPS based systems do not fulfill the requested functionality. In order to compensate for varying selectivities of different sensors, concepts of sensor fusion are applied. Sensor data in ROS is exchanged by so called messages, which can easily be logged to a database. For processing this data ROS integrated tools like NumPy (matrices and mathematical function) or OpenCV (image processing) are used. An interface from the database to MATLAB is also a powerful tool for evaluating the sensor data offline and testing first algorithms.

In practice color cameras (for documentation purposes), 3D cameras, laser range finders as well as ultrasonic multi reflectance sensors are used. In addition a priori data (such as maps or row distances) or GPS sensor information can be included and thereby increase the robustness of the navigation or the safety level.

Within ROS plugins for different sensors have been generated (color camera, 2D laser scanner Sick LMS511; 3D laser scanner Nippon FX-8; ToF camera Mesa SR4500). Together with environmental data of crop plants (or obstacles) the robot behavior with respect to the navigation and the application can be evaluated prior to field tests. As for example the leaf wall area for controlling precise sprayer can be virtually measured and the reduction of chemicals can be evaluated. ROS enables the usage of the same control software for the simulation and the hardware (robot, actuators), thereby strongly reducing the development times.

As a result the simulation environment has been developed and the results of first reactive row navigation algorithms are evaluated and compared to dynamic tests with real robots.

Keywords: Service robot, navigation, Robotic Operating System ROS, sensor fusion, orchard, vineyard

INTRODUCTION

First autonomous fieldrobots, running in outdoor areas, are performing services like phenotyping in corn (Ruckelshausen et al. 2009) or plant protection in orchards (Raussendorf 2013). Navigation in these environments has a high complexity and leads to new kinds of development strategies.

With increasing computational- and graphical power, 3D robot simulators are getting more and more comfortable in practice. There are many frequently maintained open source simulators available (e.g. ARS (ARS), LpzRobots (University of Leipzig)) as well as commercial ones, like “Microsoft Robotics Developer Studio” (Tsukor et al. 2012) or “Webots” (cyberbotics). Gazebo (Koenig and Howard) is probably the most complete open source 3D simulator obtainable at this time and is well supported by the Robot Operating System framework ROS (Quigley et al. 2009).

The combination of ROS and Gazebo is a powerful tool for running, developing and testing navigation software for any kind of autonomous systems.

Sensors and sensor systems plays an important role for controlling a robot in a mainly unpredictable environment. In our case an electrical driven vehicle has to navigate in between fruit trees and perform some actions like plant protection or mulching. 2D and 3D laser range finder, Time of Flight cameras and ultrasonic range finder are the preferred sensors to locate the trees and potential obstacles in the row. Moreover, these scanners can be used to detect the leaf wall of the trees and control the plant protection application, which leads to saving chemicals and reducing the impact on the environment.

Using 2D laser range finder for row navigation is quite popular and has been shown by other research teams (Hamner et al. 2010) (Hansen et al. 2011). Here the focus is set on the navigation. Safe obstacle detection is neglected. One possibility to find obstacles in the row is to use an inclined 2D laser scanner and generate 3D data by combining the scans with the covered distance (Freitas et al. 2012). This assumes a moving vehicle and precise distance information. For our vehicle a nonrotating 3D laser scanner or a Time of Flight camera will be used.

Within ROS and Gazebo a plugin model is supported to program simulated sensors and actuators. These sensor drivers generate data in the same format as the driver for the real sensor does. Thus, a navigation algorithm cannot distinguish between the real world sensor data and the simulated sensor data. Furthermore, ROS has the ability to save all data with time stamps to a file and play it later on back to the system. So, algorithms can be optimized offline. Figure 1 shows the development process of the navigation software including the simulation environment, an indoor simulation platform and the service robot elWObot (Linz et al. 2014).

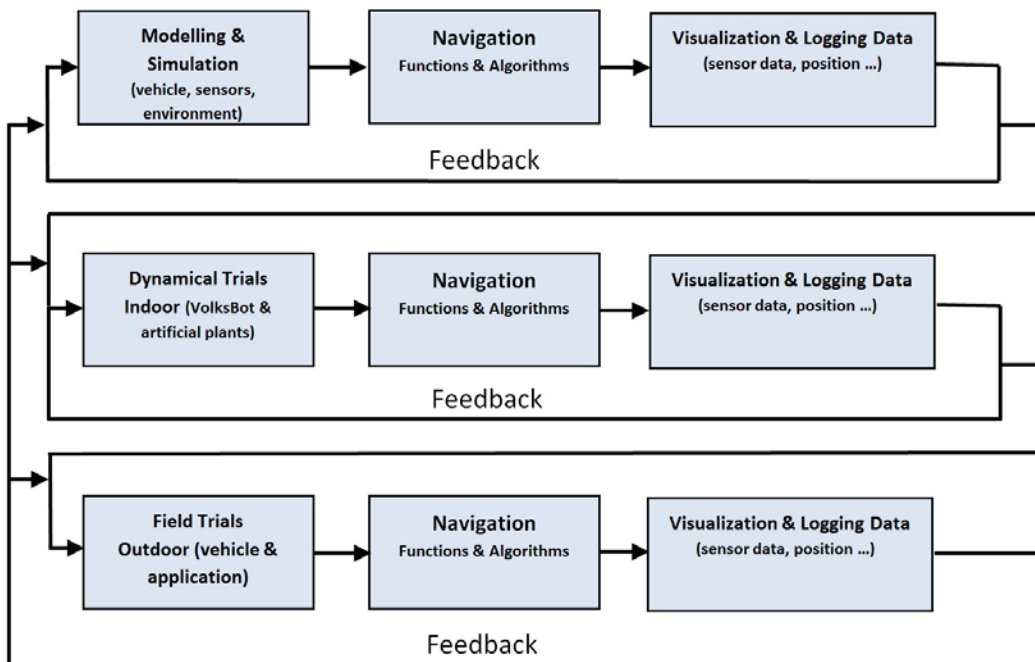
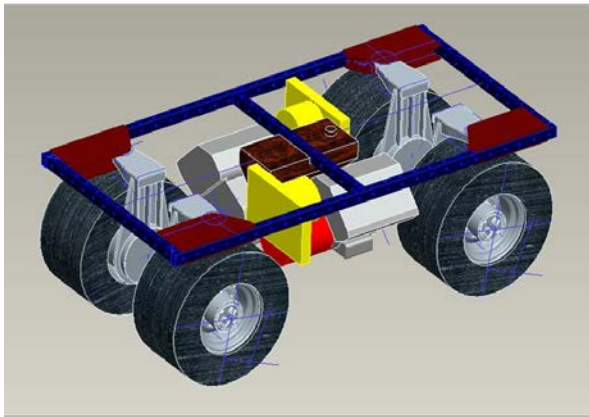


Fig. 1. Development process for the navigation.



*Prototyp vehicle orcharding
Realization of a platform strategy*

*dimensions (l x w x h):
2600 mm x 1300 mm x 930 mm*

*drive train: 30 kW
Working elements: 30 kW*

*Electrical single wheel drives,
Individually steerable*

voltage 600 V DC

Fig. 2. Layout of the vehicle.

SERVICE ROBOT CONCEPT

Robot platform

For the robot platform, a diesel electric concept is chosen. The main components are two 30kW diesel motors with generators and four independent steered wheels with integrated electric hub motors (Fig. 1.). Each wheel is driven by 7kW power, thus one diesel motor can be used for the drives, the other one for the equipment e.g. the electrical driven sprayer. The advantage over the common used hydraulic concept is a better controllability of the robot and possible energy conservation. One of the first applications is plant protection in plane orchards. So, the overall payload of the vehicle will be 1.5t, which corresponds to a sprayer plus 1000 liters of chemical spray. The modular and scalable concept makes it possible to adapt the vehicle to the different requirements in orchards and vineyards.

Plant protection equipment

A new type of sprayer for vineyards and orchards was constructed at the Institute of Viticulture Engineering in Geisenheim. Objectives have been, to reduce the power consumption and optimizing the air distribution and speed. The result is a sprayer having 8 blowers, one for each nozzle, and two pumps, all of which are electrically driven. Using frequency converters allows the adaption of the air flow by regulating the rotation speed, according to the height and depth of the rows, considering cross wind effects as well. In addition, pressure control of the spraying system also works with rotation speed control of the pumps, thus avoiding pressure regulating with a waste gate. Compared to reference spraying machinery, test results showed considerably less power demand, simultaneously gaining application quality.



Fig. 3. Test sprayer.

Sensors

To navigate the service robot in an orchard or vineyard several sensors are needed for detecting the trees and avoiding obstacles. The main navigation task is conducted by the Sick outdoor laser range finder LMS511. The choice was this scanner, because of its robustness and reliability. This type of sensor typically misses obstacles over and under its scan field. To overcome this problem a 3D Time of Flight camera (sr4500 from Mesa) and a 3D laser range finder (FX8 from nippon signals) is applied. Four ultrasonic range finders, an IMU, a GPS and a network camera for documentation purposes complete the list of deployed sensors. The table below shows the sensors with their tasks and options.

Table 1. Applied sensors

Sensor	Type of Measurement	Detection Range	Tasks and Options
1x Sick LMS511	Laser Range Finder Time Of Flight	Field of View: 190° Angular Res.: 1.66° Max. Distance: 80m	- Navigation - Obstacle Detection
1x Nippon Signal FX8	3D Laser Range Finder Time Of Flight	Resolution: 65 x 40 Field of View: 50° x 60° Max. Distance: 12m	- Obstacle Detection - Leaf Wall Detection - Navigation
2x Mesa SR4500	3D Time Of Flight Camera	Resolution: 176 x 144 Field of View: 69° x 55° Max. Distance: 9m	- Obstacle Detection - Leaf Wall Detection - Navigation
4x Microsonic wms-340	Ultrasonic Range Finder	Max. Distance: 5m	- Obstacle Detection - Leaf Wall Detection

1x IMU Razor 9DOF	Inertial Measurement Unit	ITG-3200-triple-axis digital-output gyroscope. ADXL345-13-bit resolution, $\pm 16g$, triple-axis accelerometer HMC5883L-triple-axis, digital magnetometer.	- Decline Detection - Calculate Absolute Position - Velocity
1x Navilock NEO 6P	GPS-Receiver	Positioning accuracy: GPS: 2.5 m SBAS: 2.0 m SBAS + PPP: < 1 m (2D, R50) SBAS + PPP: < 2 m (3D, R50)	- Calculate Absolute Position - Velocity
1x Levelone FCS-122	Network Web-Cam	Resolution: 1280 x 800 Framerate : 10fps	- Documentation - Surveillance
4x Incremental Encoder	Wheel Position	---	- Velocity - Calculate Absolute Position - Odometry
4x Rotary Sensor	Absolut Wheel Angle	---	- Calculate Absolute Position - Odometry

VIRTUAL AND EXPERIMENTAL SIMULATION METHODS

Indoor simulation platform

Outdoor trials are only feasibly in spring, summer and in good weather conditions. It is also a time consuming process to prepare such trials. To have some hardware for testing first row navigation algorithms as well as the system technology, a small indoor robot platform is sufficient. The VolksBot from the Fraunhofer Institute is equipped with two 150W motors, two 12V batteries and a power control unit. Figure 4 shows the indoor robot platform with an industrial PC on the top, a wireless access point, a LIDAR and a GPS, which is exactly the same hardware used later on, on the service robot elWObot.



Fig. 4. The simulation platform VolksBot, running on an indoor test track.

Robot Operating System (ROS)

The Robot Operation System is a framework with a valuable collection of libraries and tools especially for programming autonomous robots. The main task of the framework is the data exchange between different processes. The processes in ROS are called “nodes” and the data for exchange are called “messages”. A simple example with three nodes is a driver for the laser scanner, a navigation task and a motor controller driver. The scanner generates sensor data, the navigation task processes this data and sends some commands to the motor controller to move the robot. The management of these data exchange is being done in the background by the “roscore”.

One helpful and important tool for the development of the navigation stack is Rviz. This tool is be used for 3D visualization of different data types in ROS. Typical data for visualization are 2D laser scans, 3D point clouds, camera images or position data. Further on, additional data can be made visible by markers.

Gazebo

Gazebo is an open source 3D simulation environment for robots, which is built on the top of the **Open source Dynamics Engine ODE** and the **Open source 3D Graphical Rendering Engine OGRE**. ODE is a very fast, powerful, robust and flexible physical engine especially for simulating vehicles and objects in virtual reality environments. It also has integrated collision detection with friction.

The vehicle and other objects are described with the **Unified Robot Description Format URDF**, which uses the XML-Format. In this description, friction of the wheels or effort and limits of a joint can be defined. This is important to design a model as close as possible to reality.

ROS supports this simulator very well by having a bridge for sending and receiving messages between a ROS-node and the simulation environment. There is also a plugin system applied, where new sensors and actuators can be integrated. By using a standard message format, data from the simulated sensors and the position of the robot can be visualized by Rviz.

SIMULATION RESULTS AND EXPERIMENTAL TEST

Absolute position data

To get absolute position information from Gazebo or from the real robot, odometric vehicle data is needed. The VolksBot simulation platform has a differential drive. From both, the simulated robot and the VolksBot we get the velocity of the left and the right wheels. If the length of the axis is known, we can calculate the position after every sample by the formula in Figure 5.

The formula is an approximation to the real position, but with a high sample rate of the velocity it will be accurate enough. In the real world as well as in the simulated world we have friction between the wheels and the ground, which generates slip. This slip gives an integral error to our calculation. One possibility to overcome this is to measure the slip factor and include it in the calculation.

$$P_n \equiv \begin{pmatrix} x_n \\ z_n \\ \theta_n \end{pmatrix} = \begin{pmatrix} x_{n-1} \\ z_{n-1} \\ \theta_{n-1} \end{pmatrix} + \begin{pmatrix} v \cdot \Delta t \cdot \sin(\theta_{n-1} + \frac{1}{2} \Delta \theta \Delta t) \\ v \cdot \Delta t \cdot \cos(\theta_{n-1} + \frac{1}{2} \Delta \theta \Delta t) \\ \Delta \theta \cdot \Delta t \end{pmatrix} \quad \begin{array}{l} \theta = \text{robot orientation} \\ P = \text{robot position} \end{array}$$

Fig. 5. Position calculation with a differential drive (Hertzberg et al. 2012).

That's not very practical, because usually you don't know the factor for all surfaces. A better solution would be to use other sensors like a GPS and an IMU to correct the position. This for example can be done by fusing them with an Extended Kalman Filter.

The position calculation for elWObot and its simulated model is more complicated, because we have four independent steered wheels. If a virtual axis is used for the back and the front wheels, the "bicycle model" can be conducted (Fig 6). Because of the approximated model, the slip and the side drift of the not steered wheels, we have again an integral error. Here once more, additional sensors can overcome this problem.

A model with sensors for the simulation platform VolksBot

A model for the simulation platform has been designed and tested in a virtual environment. The model has a laser range finder, a Time of Flight camera and a video camera on board (Fig. 7). The drivers for these sensors generate standard data, like scan points, a point cloud or an image (Fig. 8). This data is subscribed and processed by the ROS node "toolbox", which includes some helpful functions and the row navigation stack.

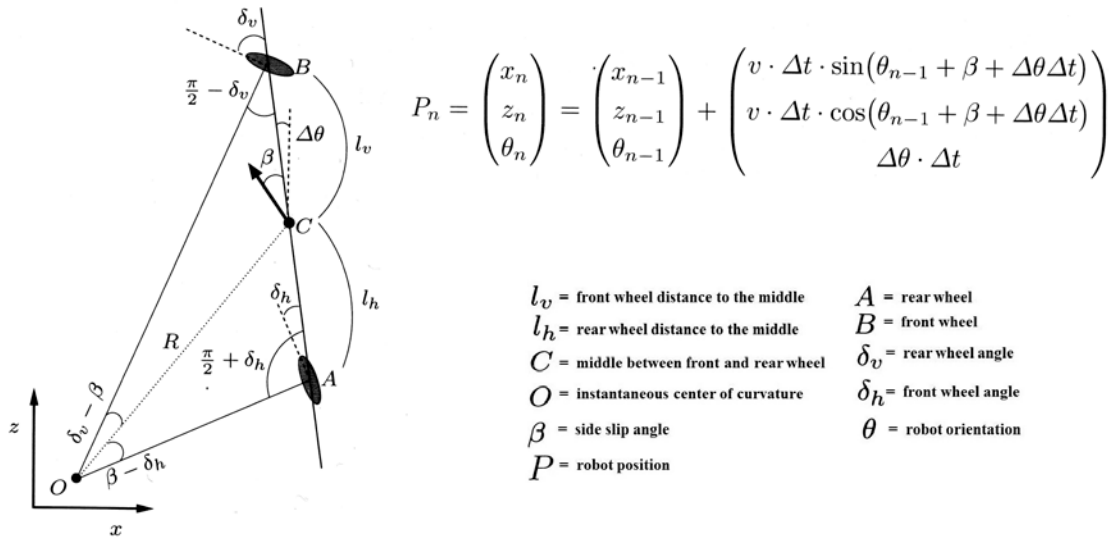


Fig. 6. The "bicycle model" for one or two steered axis and the calculation of the position (Hertzberg et al. 2012).

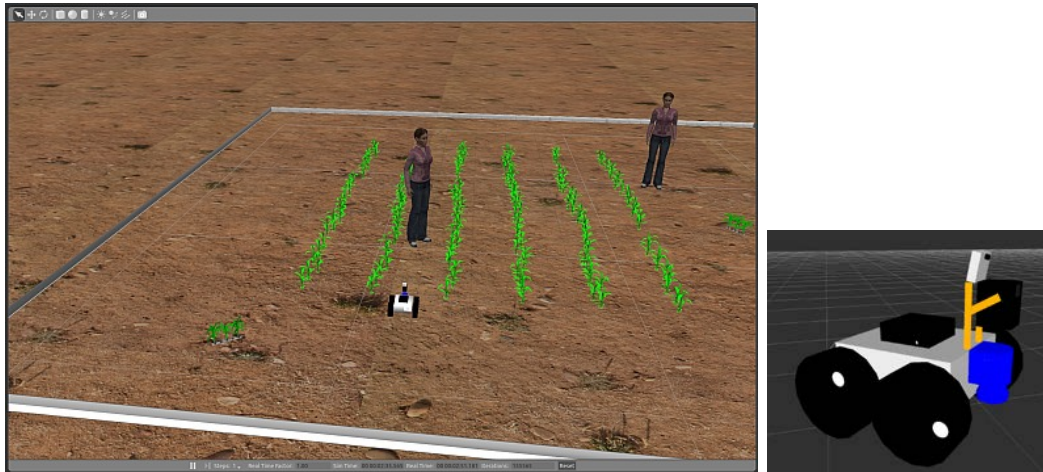


Fig. 7. The VolkBot model shown in the virtual world (Gazebo) and in Rviz with a laser range finder, a Time of Flight camera and a video camera.

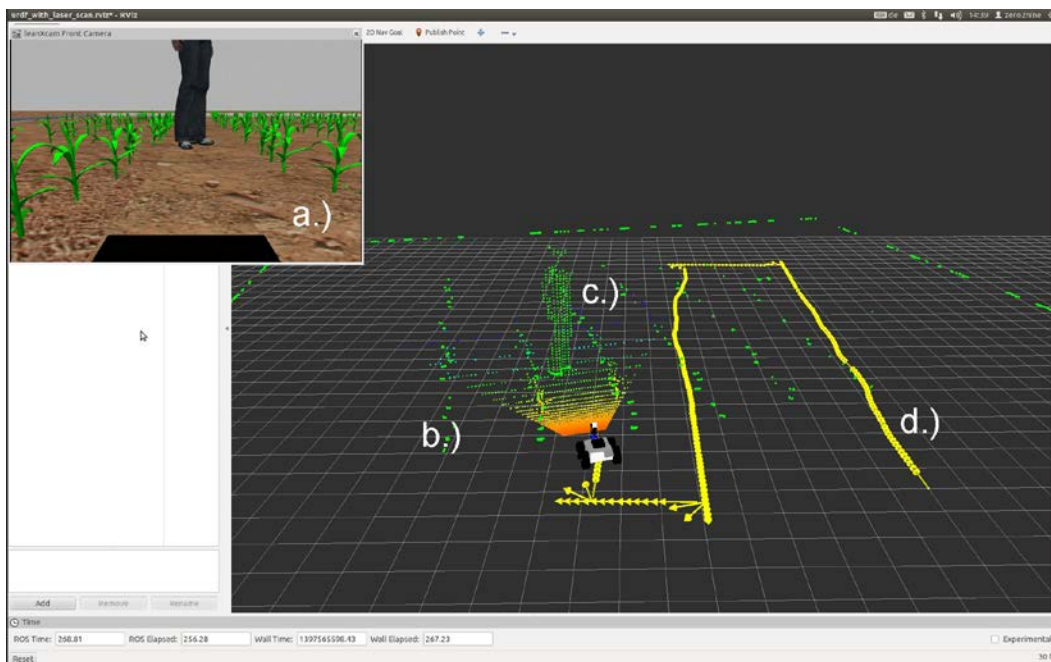


Fig. 8. Sensor data shown in Rviz. a.) Video camera Image b.) Laser scan points c.) Point cloud (Time of Flight camera) d.) Position data (odometry)

A reactive row navigation algorithm

To reduce the laser scan data, a matrix with a field of interest was generated (Fig.9). If a laser point touches a field, the field is set to 1 (red). Depending on the distance to the middle of the matrix and the y-distance away from the robot the steering angle is set.

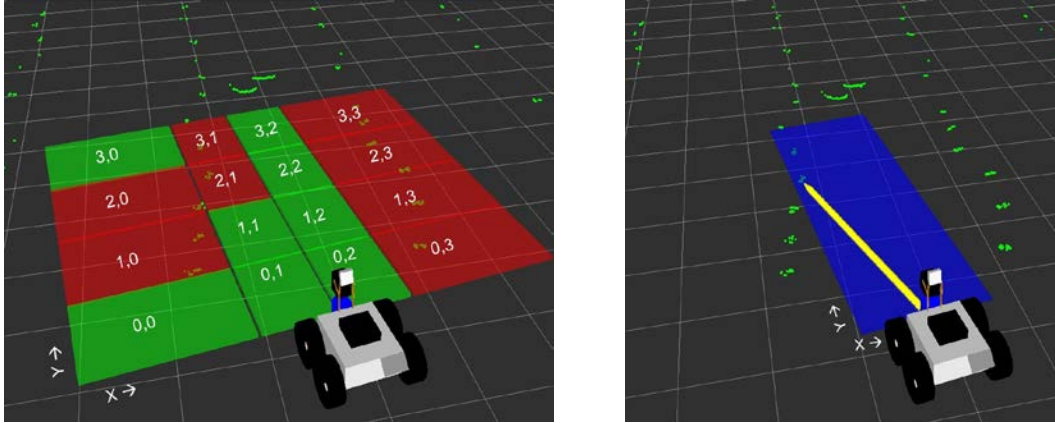


Fig. 9. Reactive navigation method. Matrix for steering and speed control.

The velocity is controlled by one field (corridor). The closest scan point to the robot in y-direction is evaluated. For calculating the velocity a linear function between a maximum speed and a minimum speed is conducted. If the y-distance of the scan point falls below a defined threshold, the robot stops.

The navigation stack was also tested with the VolksBot in an artificial indoor environment and in real corn fields. The result was a robust navigation in the row. The Time of Flight camera can be used to detect obstacles in the row which will be missed by the laser scanner.

A model with sensors for eIWObot

Figure 10 shows a model of eIWObot with several sensors mounted at the front. There is also a GPS and an IMU on the top of the video camera. In practice only 4 steering modes for the service robot makes sense. These steering modes are shown in Figure 11.

Front and rear wheels steering is initiated at the end of the row for the turn. In this mode a smaller steering angle can be reached.

Standard PID control algorithms in ROS are used to drive the wheels and the steering. Together with the simulated inertia and friction, the behavior of the model gets more close to the behavior of the real service robot.

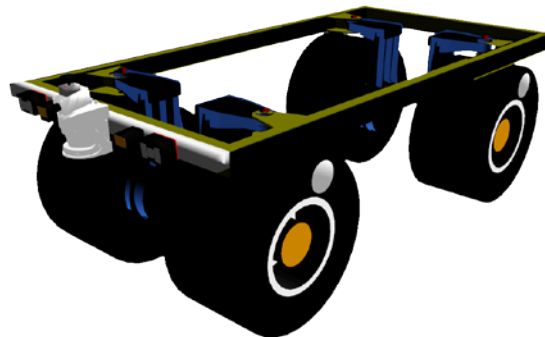


Fig.10. Gazebo model of eIWObot.

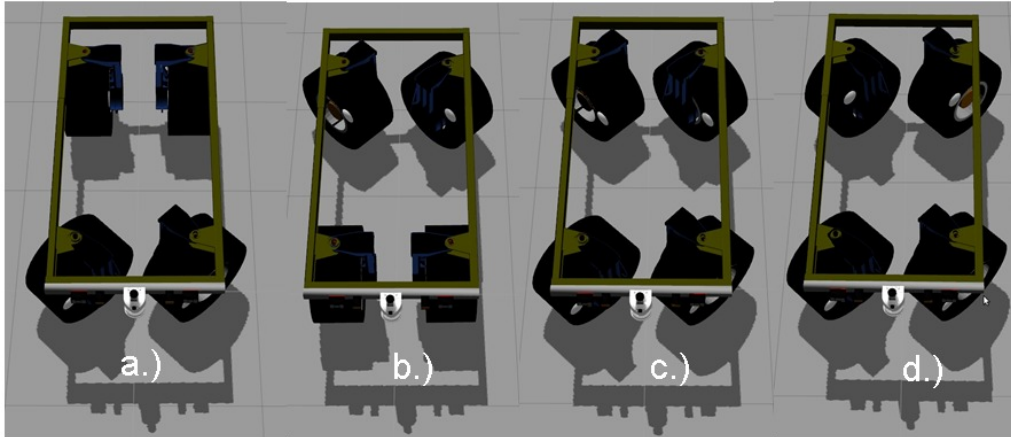


Fig.11. eWObot steering modes: a.) Front wheels steering, b.) Rear wheels steering, c.) Front and rear wheel steering, d.) Crab mode steering.

Outdoor sensor data collection

Within the use of a remote controlled vehicle from the company Raussendorf, first sensor data have been collected in orchards (apple, pear and cherry). ROS gives the ability to record all distributed messages of the system to one file. This file can then offline be used for screening with Rviz (Fig. 12). Moreover, the data can be processed by ROS integrated tools like NumPy or openCV to develop or improve the navigation algorithm.

CONCLUSION AND OUTLOOK

First reactive navigation algorithms have been tested in Gazebo as well as on a simulation platform in indoor and outdoor test tracks. A model for eWObot is completed in Gazebo, including physical characteristics like friction and inertia. PID controllers for the wheels and the steering are conducted, to have more parameters for the description of a realistic model. Real sensor data are collected in orchards (Fig. 12) and can be used to set the parameters for the navigation stack. This will be tested in a virtual environment, on the simulation platform and later on, on the service robot. Sensor information from the Time of Flight cameras will be evaluated and be used to detect obstacles and the leaf wall density. This information can be used to design a realistic simulation environment with orchards or vineyards in Gazebo. All the results, collected with the simulations will be applied to eWObot on the outdoor tests.



Fig.12. Remote controlled vehicle with mounted sensors and sensor data in Rviz.

ACKNOWLEDGEMENTS

The project is conducted in cooperation between University of Applied Sciences Osnabrueck, Technical University Dresden, Geisenheim University and the companies Karl E. Brinkmann GmbH, Raussendorf Maschinen- und Gerätebau GmbH, Obstland Dürreweitzschen AG and Weingut Schloss Proschwitz. The project is supported by funds of the Federal Ministry of Food, Agriculture (BMEL) based on a decision of the Parliament of the Federal Republic of Germany via the Federal Office for Agriculture and Food (BLE) under the innovation support program.

REFERENCES

- ARS. *Physically-accurate robotics simulator*. <https://pypi.python.org/pypi/ARS>. Accessed April 2014.
- CYBERBOTICS. *Webots*. <http://www.cyberbotics.com/overview>. Accessed April 2014.
- FREITAS, G., HAMNER, B., BERGERMAN, M., AND SINGH, S. 2012. A Practical Obstacle Detection System for Autonomous Orchard Vehicles. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Proceedings of a meeting held 7-12 October 2012, Vilamoura-Algarve, Portugal.*, 3391–3398.
- HAMNER, B., SINGH, S., AND BERGERMAN, M. 2010. Improving Orchard Efficiency with Autonomous Utility Vehicles. In *American Society of Agricultural and Biological Engineers (ASABE), An ASABE Meeting Presentation*.
- HANSEN, S., BAYRAMOGLU, E., AND ANDERSEN, J.C. 2011. Orchard navigation using derivative free Kalman filtering. In *American Control Conference (ACC), 2011*. IEEE, Piscataway, NJ, 4679–4684.

- HERTZBERG, J., LINGEMANN, K., AND NÜCHTER, A. 2012. *Mobile Roboter. Eine Einführung aus Sicht der Informatik*. eXamen.press. Springer Vieweg, Berlin.
- KOENIG, N., AND HOWARD, A. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, 2149–2154.
- LINZ, A., BRUNNER, D., FEHRMAN, J., GRIMSEL, M., HERLITZIUS, T., KEICHER, R., RUCKELSHAUSEN, A., SCHWARZ, H.-P., AND WUNDER, E. 2014. "elWObot"- A Diesel-Electric Autonomous Sensor Controlled Service-Robot for Orchards and Vineyards. In *Proceedings MCG 2014 4th International Conference on Machine Control & Guidance*, Braunschweig.
- QUIGLEY, M., CONLEY, K., GERKEY, B., FAUST, J., FOOTE, T., LEIBS, J., WHEELER, R., AND NG, A.Y. 2009. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*.
- RAUSSENDORF. 2013. *Raussendorf Obstroboter Cäsar*. <http://www.raussendorf.de/obstroboter.html>. Accessed September 2013.
- RUCKELSHAUSEN, A., BIBER P., DORNA M., GREMMES H., KLOSE R., LINZ A., RAHE F., RESCH R., THIEL M., TRAUTZ D., AND WEISS U. 2009. BoniRob – an autonomous field robot platform for individual plant phenotyping. In *Proceedings of the 7th European Conference on Precision Agriculture (ECPA), Wageningen Academic Publishers*, pp. 317–318.
- TSUKOR, V., KLOSE, R., AND RUCKELSHAUSEN, A. 2012. Multi-sensor simulation method for outdoor plant phenotyping based on autonomous field robots. In *Proceedings International Conference of Agricultural Engineering-CIGR-AgEng 2012*, pp 55-60.
- UNIVERSITY OF LEIPZIG. *LpzRobots*. <http://robot.informatik.uni-leipzig.de/software/>. Accessed April 2014.