Evaluating the PORTABILITY of the FROBOMIND ROBOT SOFTWARE architecture to new AUTONOMOUS PLATFORM

Jæger, Claes Lund Dühring; Jensen, Kjeld; Larsen, Morten; Hundevadt, Søren; Griepentrog, Hans W.; Jørgensen, Rasmus N.

Publication date: 2014

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Evaluating the PORTABILITY of the FROBOMIND ROBOT SOFTWARE architecture to new AUTONOMOUS PLATFORM

Claes Jaeger-Hansen1, Kjeld Jensen1, Morten Larsen2, Sren Hundevadt2, Hans W. Griepentrog3, Rasmus N. Jørgensen4

1Institute of Applied Eng. Bio/Environmental Technology, University of Southern Denmark, Campusvej 55, DK-5230 Odense M, Denmark
2Department of Engineering, Aarhus University, Finnåsvej 22, DK-8200 Aarhus N, Denmark
3Aarhus University, Finlandsgade 22, DK-8200 Aarhus N, Denmark
4Institute of Chemical Eng., Biotechnology and Environmental Technology, University of Southern Denmark, Campusvej 55, DK-5230 Odense M, Denmark

Introduction
The University of Hohenheim (Unihh) is conducting research in novel precision agriculture production methods which involve crop scoping and scouting using machine learning applications and are collaborating with the University of Southern Denmark on utilizing autonomous field robots in those projects. Unihh has a field robot platform Autonomous Mechanisation System (AMS) consisting of the Hako tractor retrofitted with sensors, actuators and the MobotWare framework (Reske-Nielsen et al., 2006; Griepentrog et al., 2011; Griepentrog et al., 2009) allowing autonomous navigation in the field. Recently Unihh acquired a new robot platform called Armadillo Scout which is a modular platform designed for precision agriculture research. Armadillo Scout runs MobotWare (Back et al., 2010) and the open source FroboMind architecture (Jensen et al., 2012) based on Robot Operating System (ROS) (Quigley et al., 2009). MobotWare has successfully been ported to both the AMS and the Armadillo Scout. To be able to use the same software solution on both platforms Unihh set out to implement FroboMind on the AMS in this project. The purpose is to have FroboMind and MobotWare run on both platforms which makes it possible to choose the best suitable solution for a given research project based on the qualities of MobotWare and FroboMind. The aim of this work is to evaluate the portability of FroboMind to a new robot platform with respect to the implementation process and performance of the robot when operating autonomously in a precision agriculture environment.

Materials and methods

Autonomous Mechanisation System - The Hako tractor

The AMS (Fig. 1) is the Hako tractor that was used as the target platform is a part of an advanced robotic crop-establishment and control system based on Precision Agriculture principles, and was developed by University of Copenhagen in cooperation with the Department of Automation of the Technical University of Denmark (DTU). The AMS is currently being further developed at the Department of Instrumentation and Test Engineering at University of Hohenheim. AMS is based on a conventional 20 kW tractor (Hakotrac) 3980) which was retrofitted with several sensors for navigation along with the AMS Electronic Control Unit (ECU), a custom ECU for the AMS which was designed and fabricated by Safe and Reliable project (Griepentrog et al., 2009). The tractor navigation controller was designed to follow a predetermined route plan accurately and repeatedly across a field with planned action points for implement control (Blackmore et al., 2007), (Blackmore et al., 2004, Griepentrog et al., 2009).

Fig. 2. Autonomous Mechanisation System (AMS).

FroboMind

FroboMind is a conceptual architecture for field robot control software. The FroboMind architecture (Fig. 3) is optimized for precision agriculture and similar field robotics research projects with respect to the parameters modularity, reliability, extensibility, scalability and code reuse between different robot platforms and applications. At the current development state FroboMind is not considered to be stable or reliable enough for production use. In this project we have used the latest software submitted to the FroboMind repository at 2012-03-23. In this project FroboMind interfaces to the AMS through the ECU. The FroboMind computer communicates with the ECU through a CAN bus.

Fig. 3. FroboMind architecture.

Results

AMS position and orientation estimation

The FroboMind components which are required to perform simple navigation through row crops environments like orchards and maize fields were implemented on the AMS. Fig. 4 shows a ROS-RVIZ simulation of this simple navigation in the orchard. The green rectangle defines the estimated rows. An extended Kalman filter fuses the estimated output with the IMU data. The blue lines visualize the estimated rows. An extended Kalman filter fuses the estimated output with the IMU data.

Fig. 4. Simulation of robot orientation and shape in a crop environment.

FroboMind portability

In total the AMS was able to execute FroboMind with respect to modularity, extensibility, code reuse and plan a mission was implemented Frobomind on the AMS. The following table shows the predicted assignments and times. The estimation was based on a team of four persons implementing the new architecture.

<table>
<thead>
<tr>
<th>Task</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementing working software interface</td>
<td>10</td>
</tr>
<tr>
<td>Drawing the ECU to Enode interface in the ECU</td>
<td>4</td>
</tr>
<tr>
<td>Prepare hardware for the ECU</td>
<td>2</td>
</tr>
<tr>
<td>Implementing VME interface for the ECU</td>
<td>2</td>
</tr>
<tr>
<td>Exploring and adding new nodes to FroboMind</td>
<td>6</td>
</tr>
<tr>
<td>Intermediate verification of the AMS in FroboMind</td>
<td>10</td>
</tr>
<tr>
<td>Testing functionality in an orchard</td>
<td>50</td>
</tr>
</tbody>
</table>

AMS and FroboMind interoperability in an orchard environment

In order to test the FroboMind portability with respect to reliability and precision agriculture applications it was decided to perform a field test in an apple tree orchard at Unihh. Fig. 5 shows the approximate orchard tree sizes. The tree rows are 100 meter long and interspaced by 4 meter. In some rows a few trees are missing and/or replaced by younger and smaller trees. During the FroboMind integration and the first tests it was discovered that when requesting the AMS to drive straight for a given research project based on the qualities of MobotWare and FroboMind, performance in an orchard environment it would drift slowly to the left. It is unknown what the cause of this problem is, however based on the performed tests it is likely that the problem is with the steering hardware for a given research project based on the qualities of MobotWare and FroboMind which makes it possible to choose the best suitable solution

Fig. 5. Apple tree orchard used for implementation and testing.

Testing FroboMind portability

FroboMind was ported to the Hako tractor during a one week workshop. The duration of individual working tasks was estimated.

Testing AMS and FroboMind performance in an orchard environment

The test was stopped after completing 7 rows.

- 1. round was completed successfully.
- 2. round was completed successfully.
- During this round a video recording was made while walking behind the robot to document the orchard environment. During the 3. round the robot navigation failed one time causing the robot to drive into the trees. This happened at a location where apple trees were replanted by much younger trees at the left row. During the 4. round the robot navigation failed one time causing the robot to drive into the trees. This happened at another location where apple trees were missing for more than 5 meters at the left row. 5. round was completed successfully. During the 6. round the robot navigation failed one time causing the robot to drive into the trees. This happened at the same location as the 4. round, but during the 5. round the navigation failures occurred at locations where the apple trees did not provide sufficient data for the implemented laser range scanner to detect apple trees and among the described problems the robot navigated reliably through the orchard.

Fig. 6. Navigating through the same route between two orchard rows six times.

Preparing and testing FroboMind portability

The test was stopped after completing 7 rows.

- Due to the problems with Turning to the right was achieved by hard coding a maximum turn.
- The robot failed a right turn two times. One time due to poor matching of the inter-row width and another time because one row was longer than the rest.
- Aside from these the robot performed very reliable autonomous navigation through the orchard.
- The test was stopped after completing 7 rows.

Fig. 7. Navigation through a section of the orchard.

Conclusions

For a 4 man team it is possible to implement and test FroboMind with 5 working days, though the days are longer than normal working days. The implementation is only possible as long as the hardware interfaces are well documented. It is not possible to optimize the behavior algorithms or low level control algorithms nor is it possible to debug and troubleshoot small problems on the platform within the time frame.

Acknowledgements

This research is linked to and partially funded from the Danish Ministry for Food, Agriculture and Fisheries project: FruitGrowth and the Danish National Advanced Technology Foundation project: The Intelligent Sprayer Boom.