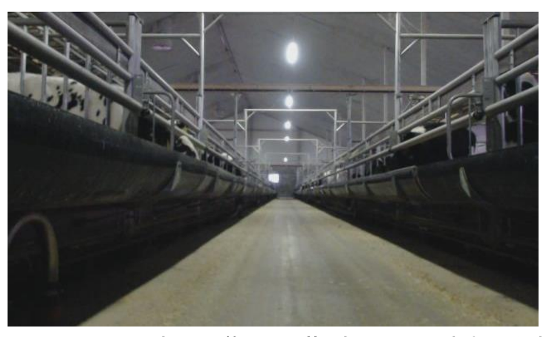
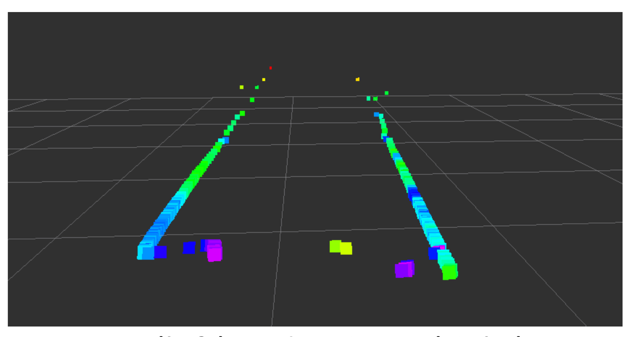
The Path to Autonomous Navigation on Industrial Robots with Robot Operating System (ROS)

W.J. (Wilco) Bonestroo, H.G. (Henk) Kortier, S. (Stan) Dekker, M. (Mick) Boe, A.Y. (Abeje) Mersha
Saxion University of Applied Sciences, Research Group Mechatronics, w.j.bonestroo@saxion.nl

1: Motivation

Traditionally, Automated Guided Vehicles (AGVs) keep track of their position using installed and known infrastructure, such as an inductive cable or RFID tags in the floor. Installing and tuning such systems is costly and the system is not flexible. Current state-of-the-art and open-source navigation algorithms allow robots to navigate based on sensors such as encoders, LiDAR and IMU and do not require installed infrastructure anymore.

How a robot "sees" the world with a camera (left) and a LiDAR (right)

2: Observations

In the robotics research community ROS has grown very quickly over the last few years. Currently, it is the main robotic framework used by the research community. Although the framework could also be used in industrial applications, the adaptation in that area is much slower. We also see that the role of software in robotics is growing. Accordingly, software could be a bottleneck for future development in robotics.

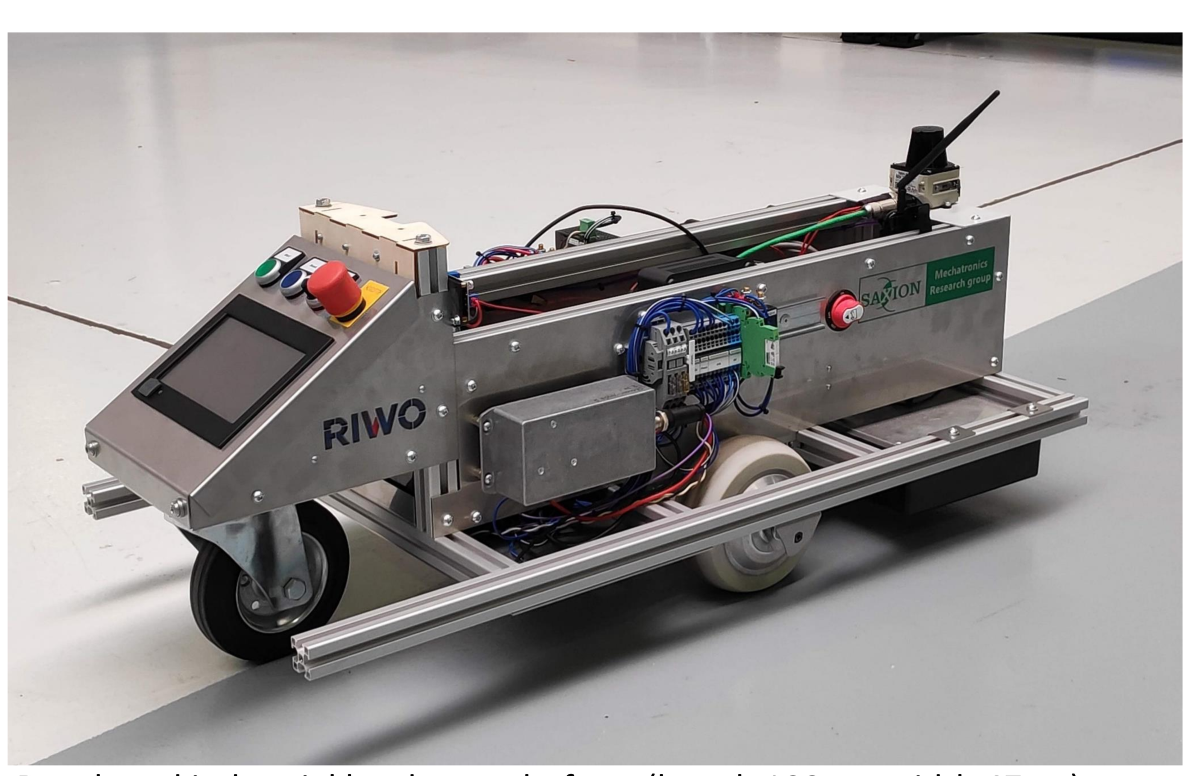
This project investigated whether open-source components can be reused to reduce engineering time. This approach is fundamentally different from a typical industrial systems engineering approach, where functionality is often developed from scratch and in-house. The current approach borrows concepts such a reusability and hardware abstraction from the domain of software engineering.

The resulting system is a modular. With this approach, a technology provider can simply reuse generic advanced software building blocks and focus on the specific aspects or hardware of the robot. In this project we focused on navigation, but the principles could also be applied to other generic tasks, such as manipulation with robot arms and advanced perception.

3: Goal & Research Question

The goal was to research whether open source ROS software could be used on an industrial robot platform with a real time industrial computer. We wanted to know if we could combine systems to apply non-real-time and nondeterministic algorithms, while adhering to real-time and safety constraints of the resulting system.

We were especially interested in the procedure to characterize the mobile robot.



Developed industrial hardware platform (length 100cm, width 47cm)

4: Methods

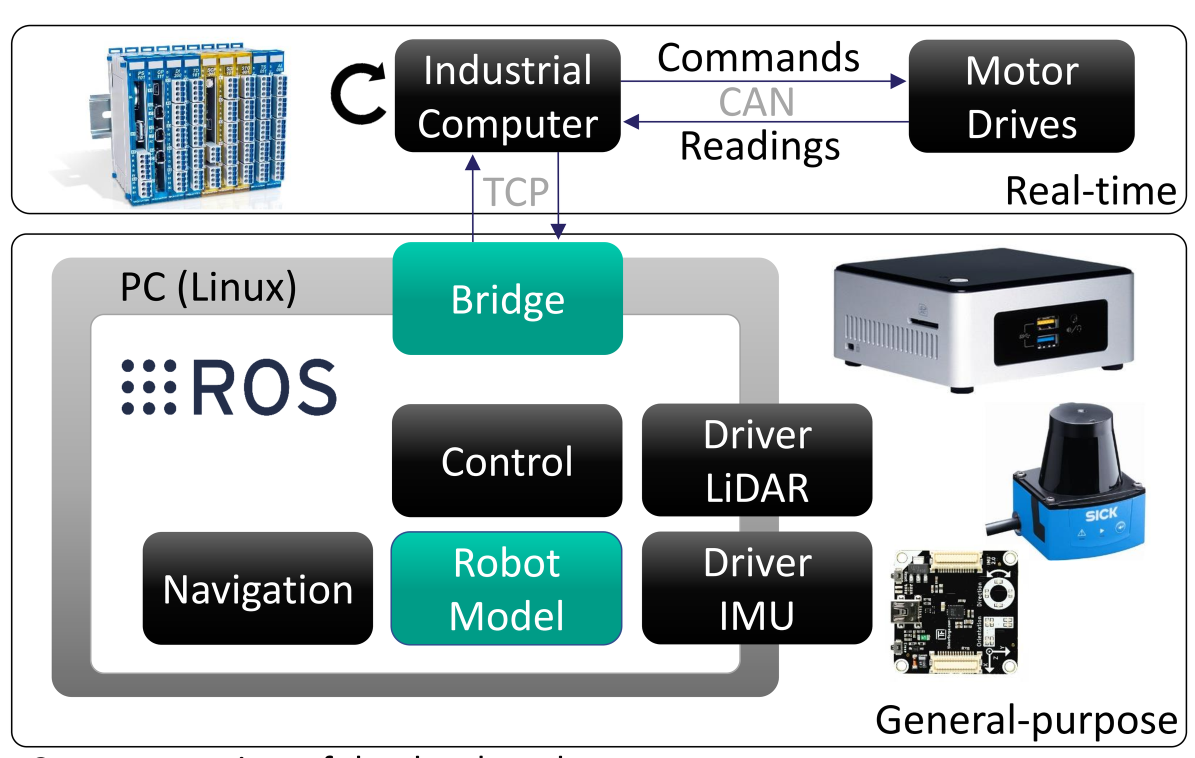
A prototype was developed using the Design Science Research method. Knowledge was developed by designing and implementing the prototype. The prototype was build with industrial hardware, an industrial PC (IPC) with a real-time operating system, combined with a regular PC with ROS.

Once the system was build, system identification was performed to characterise the system. Measurements with an optical tracking system (VICON) were performed to compare pose estimations from the robot with the actual position.

5: Results

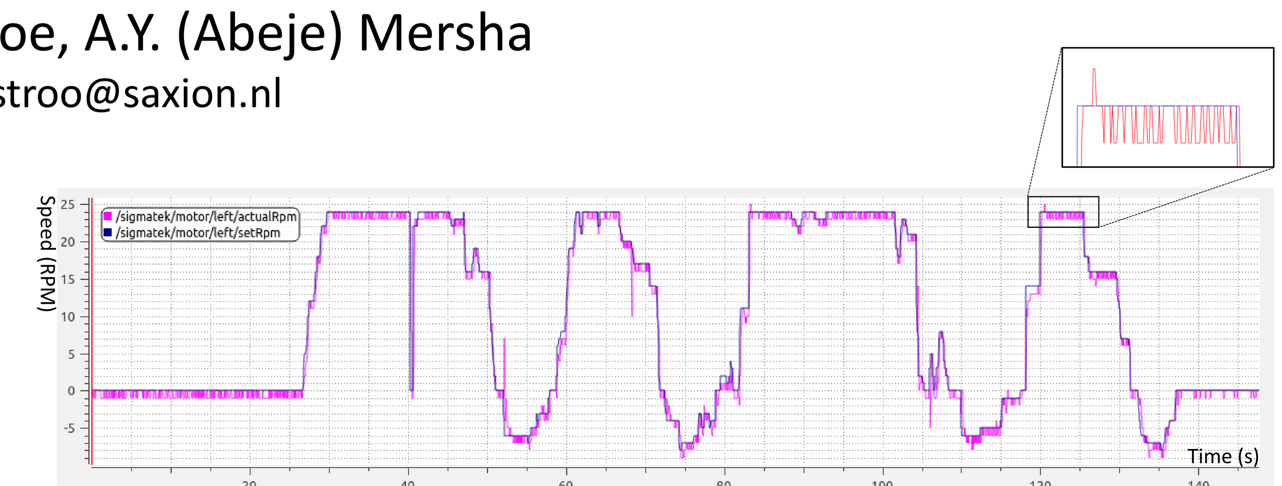
The prototype was equipped with an industrial computer for low level control and safety functions. It also had a general purpose computer for performing complex tasks such as mapping and localisation. The high-level planning software uses non-real-time software to create maps based on different sensor readings that have noise and other errors. The industrial computer continuously checks whether the computer with ROS keeps sending commands.

Once the industrial hardware abstraction is complete (bridge) all ROS tools become available to steer the robot, perform mapping, localisation and navigation, record sensor data, etc. Recorded data can be played back to optimize the robot model and the algorithms used.

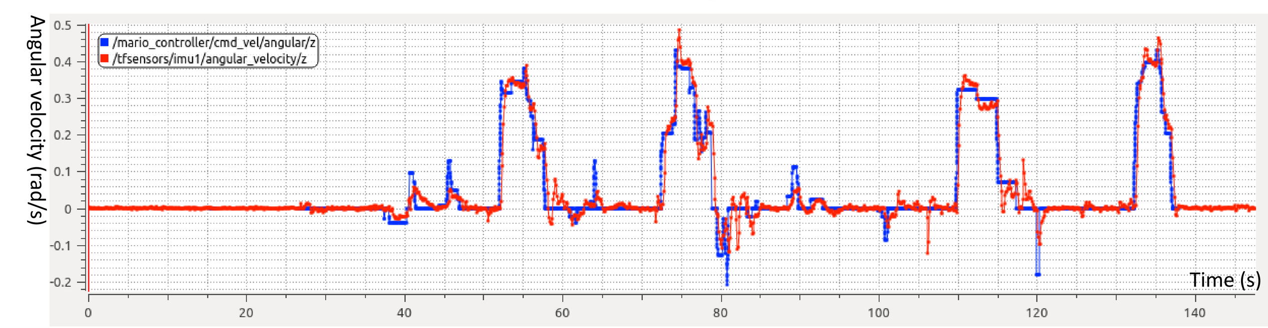


Component view of the developed system

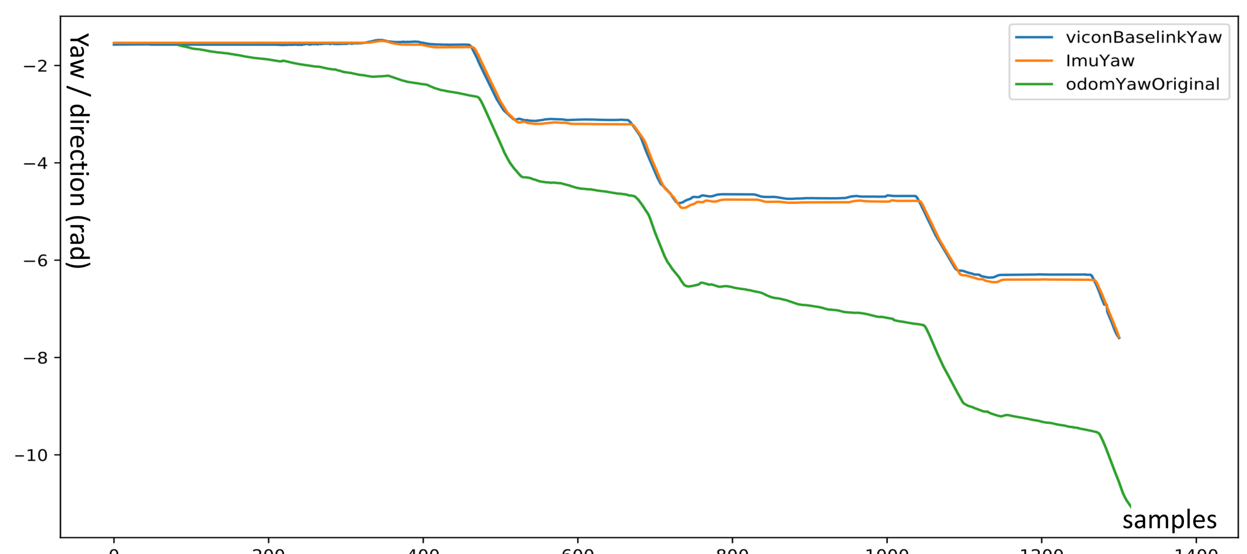
Generated map by the robot on top of actual blueprint of the mechatronics lab in the Saxion building.



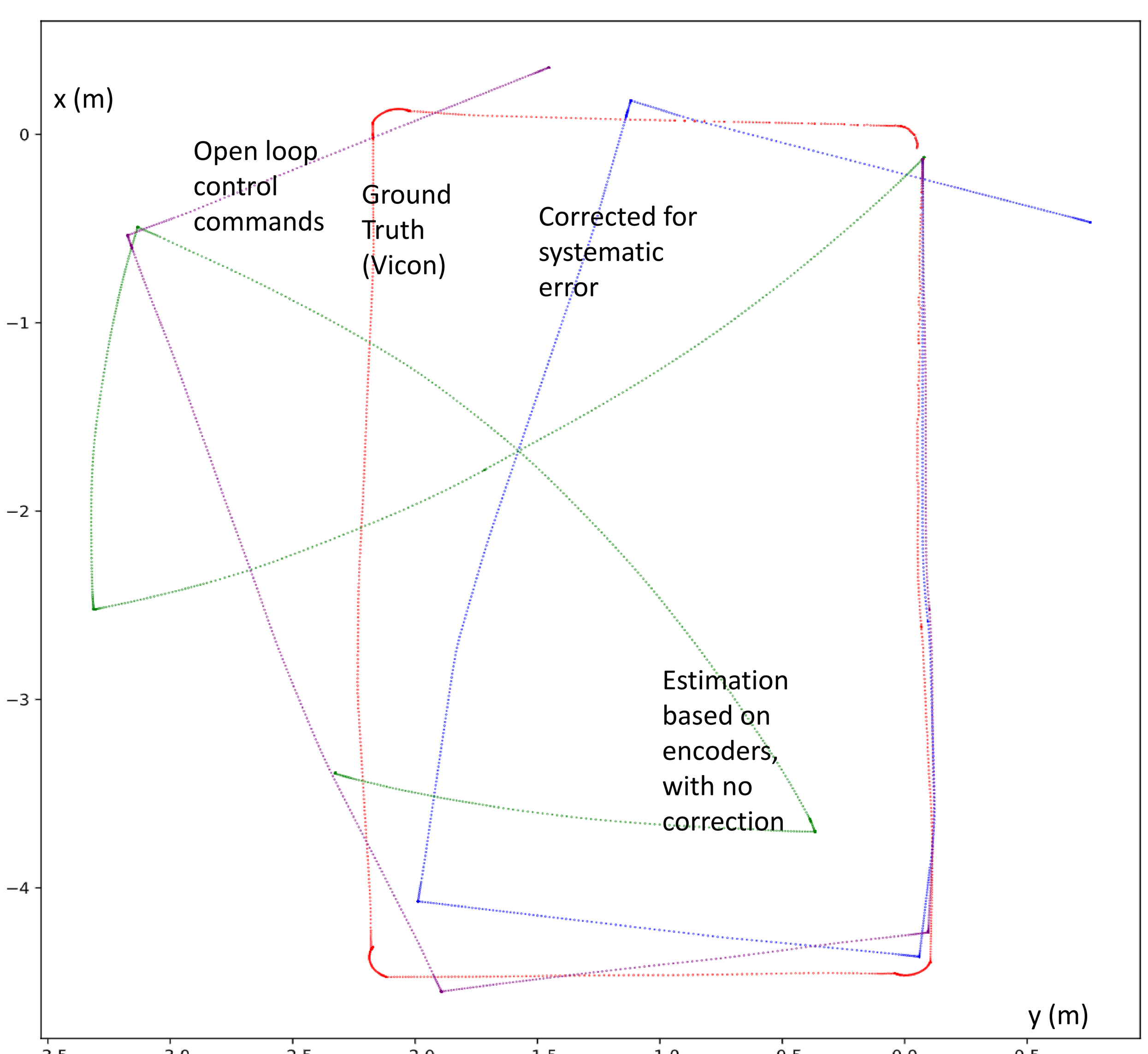
Noise and offset of wheel encoder readings (red) vs control command (blue)



Control command vs angular velocity reported by IMU



Estimated yaw based on encoders (green) and IMU (red). Encoders show large drift over time compared to ground truth (blue). IMU is more accurate.



Robot position based on several versions of odometry estimations based on encoder readings only, compared with Vicon measurements

6: Conclusions

Although the project is still work in progress, the following conclusions can be drawn.

The prototype shows an industrial mobile robot that reuses state-of-the-art and open-source algorithms on custom and industrial hardware. It is possible to interface a general purpose computer (PC) with ROS and a real-time industrial computer (IPC). The resulting system allows complex algorithms on the PC and hard real-time functionality (control and safety) on the IPC.

Based on the log data, we could conclude that encoder readings alone were too unreliable. Although the LIDAR corrects the pose, resulting in an acceptable map, integration of IMU data seems promising for more stable and robust system. The next steps are integrating the IMU in the sensor fusion and using the generated maps to perform path planning and navigation.

Moreover, we want to generalize, standardize and describe the procedure for characterizing a mobile robot. In this way, other groups and companies can also use the available ROS packages to create their own robotic platform with autonomous navigation.

SAXION UNIVERSITY OF APPLIED SCIENCES

TECH FOR FUTURE

Centre of Expertise HTSM Oost
Een initiatief van Saxion en Windesheim
mede mogelijk gemaakt door provincie Overijssel

RIWO

This research was funded by Tech for Future (www.techforfuture.nl) and was co-funded by Riwo Engineering B.V. (www.riwo.eu)