

Kalmtree used for Mobile Robot Navigation

Lars V. Mogensen * Nils A. Andersen * Ole Ravn * Niels K. Poulsen **

* *Department of Electric Engineering, Automation, Tech. Univ. of Denmark,
Build. 326, DK-2800, Kgs. Lyngby, Denmark, lvm, naa,*

or@oersted.dtu.dk

** *Department of Informatics and Math. Modelling, Tech. Univ. of Denmark,
Build. 321, DK-2800, Kgs. Lyngby, Denmark, nkp@imm.dtu.dk*

Abstract:

This paper presents an application of a simulation platform for sensor fusion in mobile robotics. The platform is based on the Kalmtree toolbox which is a set of MATLAB tools for state estimation of nonlinear systems. The toolbox contains functions for extended Kalman filtering as well as several other state of the art filters. Two robotic platforms are considered, a Medium-size Mobile Robot and a Hako tractor. The system models for the vehicles are derived and by using Kalmtree suitable filter coefficients are found.

1. INTRODUCTION

When designing and building complex systems good tools are essential for success. Good tools supports the user on the different levels of abstraction. Typically this ranges from mathematical formulation and simulation of the algorithms over numerical implementation to verification and validation of the actual device in real-time. In this paper it is demonstrated how the tool, **Kalmtree**, is used in connection to sensor fusion. The algorithms are based on the Kalman filter technique and are applied on models of two different outdoor robotic platforms.

Kalmtree is a collection of Matlab implementations for simulation and estimation in connection with nonlinear dynamic systems. The development of the toolbox has been driven by the application which is navigation of mobile robots. In this context location and mapping are corner stones.

Since it was suggested, the extended Kalman filter (EKF) has undoubtedly been the dominating technique for nonlinear state estimation. Nevertheless, the EKF is known to have several drawbacks. These are mainly due to the Taylor linearization of the nonlinear transformations around the current state estimate. The linearization requires that Jacobian of state transition and observation equations are derived. This is often a quite complex task. Moreover, sometimes there are points in which the Jacobians are not defined. In addition to the difficulties with implementation, convergence problems are often encountered due to the fact that the linearized models describe the system poorly.

There has been significant focus on this area recently, and previous work includes several toolboxes and other platforms. ReBEL (Recursive Bayesian Estimation Library) van der Merwe (2004) is a Matlab toolkit of functions and scripts, designed to facilitate sequential Bayesian inference (estimation) in general state space models. The CAS Robot Navigation Toolbox Arras (2004) is a tool for doing off-line off-board localization and SLAM on mobile robots. The design of the CAS toolbox decouples robot model, sensor models, features and algorithms used giving the user ability to adapt the toolbox by just modifying or adding the pieces in question. The toolbox

does not in its present form support the generation of real-time code for use on the robot.

The implemented methods in **Kalmtree** are described in more detail in Nørgaard et al. (2003), Sejerøe et al. (2005) and in Sejerøe et al. (2005).

The paper is organized as follows: Firstly, the overall design philosophy behind the simulation platform is described and a brief outline of the robotic platforms used is given. In section 4 a description of the models of the vehicles and sensors is given. Section 5 gives an extensive example study as well as a demonstration of the different scenarios for navigation of the two mobile robots.

2. THE PLATFORMS AND THE TOOLBOX

The overall design philosophy has been to focus on the creation of a simple, transparent, yet powerful platform making life easier for the application developer as well as the algorithm developer.

Transparency overcomes the barrier effect that is often experienced when using tools that at first sight seem very user-friendly, but when used on real problems become difficult to handle due to the inherent complexity.

The approach taken uses MATLAB as a numerical and graphical basis for developing the platform. The platform is driven from Simulink as this provides a shorter path to implementation using for instance Realtime Workshop and makes it simple to use real data for comparison.

The philosophy of the Kalmtree toolbox is to provide an open structure (see figure 1), which is easy to use and which enables the user to investigate the inner workings of the estimation algorithms. With this in mind, the structure of the Kalmtree functions is open and a large selection of functions made available. The functions are made to work in an online setting, one time-step - one update, though this does not prohibit its use in offline environments. The main changes are the breakup of the estimation loop and the introduction of an evaluator function.

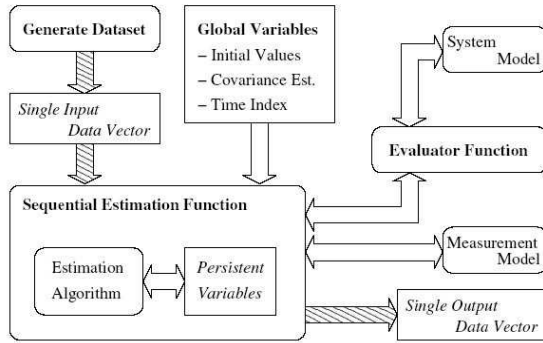


Fig. 1. The structure of Kalmtool 3.

Changing the loop means that the functions can now be used directly in an online setting provided that sufficient resources are available.

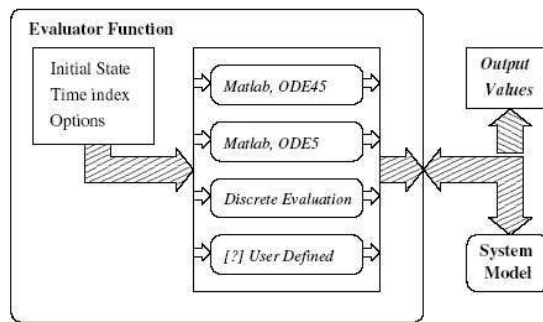


Fig. 2. The evaluator function.

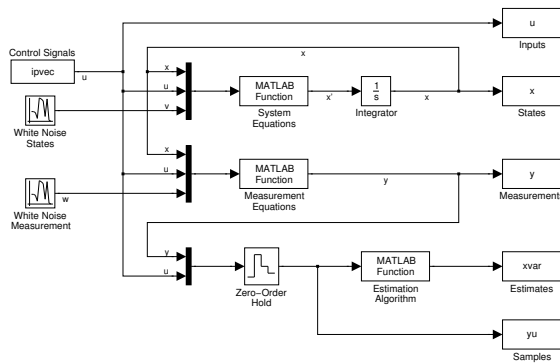


Fig. 3. The Simulink layout of a continuous system.

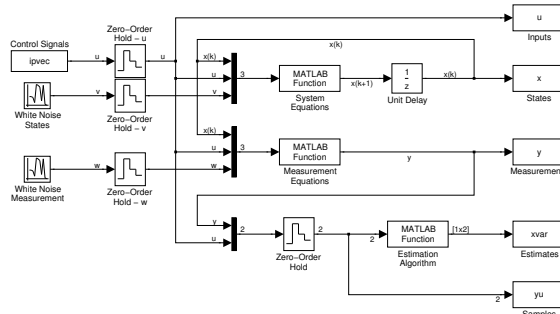


Fig. 4. The Simulink layout of a discrete time system

As seen in the above figures the user can easily add new algorithm into the platform by modifying the MATLAB function in

the Estimation block and change the system by modifying the system and measurement MATLAB blocks.

2.1 Algorithms

The Kalmtool toolbox Nørgaard et al. (2003), Sejerø et al. (2005) and Sejerø et al. (2005) is a nonlinear system parameter estimation toolbox. It is at the moment implemented in MATLAB/Simulink which does not comply with the idea of keeping the solution implemented in C and independent of other programs (as previous versions).

The Kalmtool toolbox is interesting for evaluation of sensor fusion algorithms, due to the advanced algorithms it provides, which includes:

- Stationary Kalman filter
- Kalman filter
- Extended Kalman filter
- Unscented Kalman filter
- Divided difference filter, first order
- Divided difference filter, second order
- Sequential filter (projection Theorem)
- Sequential filter (Bayes Theorem)

This toolbox makes together with the simulation platform a basis for navigation of mobile robots, where one of the key issues is fusion of the results from various types of sensor with quite different properties.

3. HARDWARE PLATFORMS

The Kalmtool is applied on simulation models of two platforms. The two platforms are The Medium Mobile Robot (MMR, see Figure 5) and the HAKO tractor (see Figure 6):



Fig. 5. Medium Mobile Robot.

The MMR robot The Medium Mobile Robot is an outdoor robot, built after the differential motion principle. The vehicle is mainly used for outdoor navigation and mapping purposes, but can also be used indoor. The vehicle deploys a GPS (Global Positioning System) and Odometry (Measurement of motion

based on wheel sensors). Besides these sensors the robot platform is also equipped with a Laser Scanner, a Gyro and a control system.



Fig. 6. Hako tractor.

HAKO Tractor The HAKO tractor is an agricultural research platform used for development of autonomous farming techniques at The Royal Veterinary and Agricultural University. Due to the diesel engine and the size of the vehicle the HAKO tractor is only used for outside experiments. In its current form the HAKO tractor and its security system needs constant supervision. The vehicle deploys GPS and Odometry for navigation purpose.

4. MODELS

The models used in connection to Kalman filtering are the process equations, which related to the vehicles, and the measurements equations, which is models of the sensors.

4.1 Vehicles

In connection to mobile robots there exists two major type steering. There is the Ackerman steering, which is known from automobiles and most tractors, and there is the differential steering which is mostly known from wheel-chairs, tanks and larger tractors. The HAKO tractor has an Ackerman steering whereas the MMR is controlled by means of a differential steering.

HAKO tractor model The odometric model is based on the geometry of the vehicle motion and a simplification of the movement by the uni-cycle principle. The main property of the Ackerman vehicle movement in a circle is, that the center points on the front and rear axle are covering concentric arches, see figure 7. The three parameters that mainly decide the odometric motion of the vehicle is L the distance between the axles, ϕ the steering angle and d_k driven distance. R is the radius of the vehicle movement.

Let x and y denote the coordinates for the center point between the non steering back wheels, θ is the heading of the vehicle and let d_k be the driven distance of the vehicle.

The odometric model of the vehicle is given by the following model:

$$\begin{aligned} x_{k+1} &= x_k + \delta d_k \cos\left(\theta_k + \frac{\delta\theta_k}{2}\right) \\ y_{k+1} &= y_k + \delta d_k \sin\left(\theta_k + \frac{\delta\theta_k}{2}\right) \\ \theta_{k+1} &= \theta_k + \delta\theta_k \end{aligned} \quad (1)$$

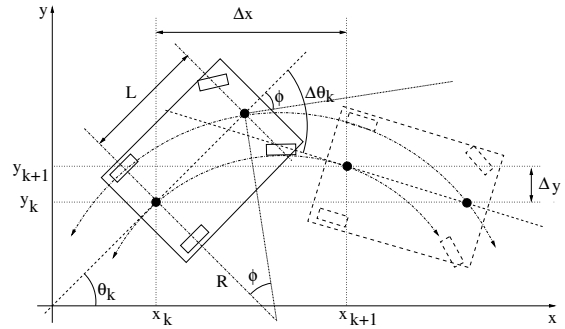


Fig. 7. Ackerman model in discrete time.

where the control input contains the covered distance δd_k and the steering angle ϕ_k .

$$\bar{u}_k = \begin{bmatrix} \delta d_k \\ \phi_k \end{bmatrix} \quad (2)$$

The relation between the steering angle ϕ_k and the change in vehicle heading $\delta\theta_k$ is given by:

$$\delta\theta_k = \frac{\delta d_k}{L} \cdot \tan(\phi_k) \quad (3)$$

MMR model The discrete model of the differentially steered vehicle is depicted in figure 8, in order to present the variables and geometry. The parameters that mainly describe the odometric motion of the vehicle are the driven distances of the two wheels and B the distance between them. The driven distance of the two wheels is proportional to the diameter of the wheel, represented by d_{wr} and d_{wl} on figure 8.

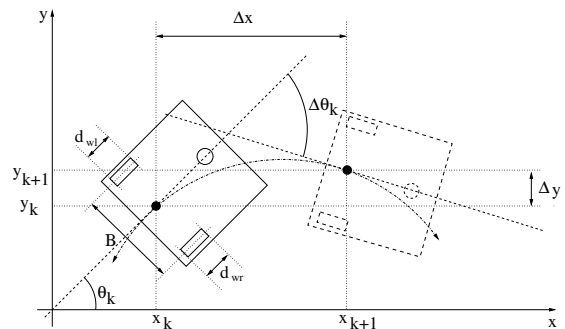


Fig. 8. Differential model in discrete time.

The state equation is the same as for the Ackerman steered vehicle, i.e. (1). However, the control input is the covered distance δd_k and the change in rotation of the robot $\delta\theta_k$.

$$\bar{u}_k = \begin{bmatrix} \delta d_k \\ \delta\theta_k \end{bmatrix} \quad (4)$$

4.2 Sensors

GPS The Global Positioning System (GPS) is a system that uses satellites to determine a position on earth. The GPS system uses well known satellite positions and precise measurements of the distance from GPS receiver to the satellites, to calculate a position measurement (p_x, p_y) on the surface of the earth.

$$\begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_x \\ e_y \end{bmatrix}$$

Here x and y are the true position whereas e_x and e_y are uncertainty in the GPS position.

Odometry This form of sensor is based on encoders attached to the wheels or motor. The measurement is simply the distance (m_l and m_r) the wheels has travel during a sampling period, i.e.

$$\begin{bmatrix} m_r \\ m_l \end{bmatrix} = \begin{bmatrix} \delta d_k + \frac{B}{2} \delta \theta_k \\ \delta d_k - \frac{B}{2} \delta \theta_k \end{bmatrix} + \begin{bmatrix} \varepsilon_r \\ \varepsilon_l \end{bmatrix}$$

By measuring how far the wheels have turned, it is possible to calculate how far the vehicle has moved and how the heading has changed. If this calculation is done often enough, the small changes in distance and direction can be summed up to give a representation of position and heading relative to the starting point. The result of this method is highly dependent of knowing the vehicle proportions precisely, since any error will be added to the position, which will then drift.

4.3 Total system models

In bot case the robots can be modelled in discrete time by a state space description

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, v_k) \\ z_k &= g(x_k, e_k) \end{aligned}$$

where the the process noise, $v_k \in \mathbf{N}(0, Q)$, and the measurement noise $e_k \in \mathbf{N}(0, R)$, are assumed to be sequences of zero mean white noise.

5. SIMULATIONS AND EXPERIMENTS

5.1 HAKO

The simulation experiments are performed in order to demonstrate the usability of the Kalmtool platform.

The covariance matrices, Q and R, has ben determined such that the overall system including the Kalman filter gives a reasonable performance.

To check if the filter converges as expected, the HAKO tractor is tested with a straight test run. The tractor is driven 10 m with the filter off and then 50 m with the filter on. The result is very satisfactory. Figure 9 shows the position. The filter position, Figure 9, converges very smoothly after about 12 s and 1.5 m of driving, and shows good results as to the capabilities of the sensor fusion algorithm, when running on the HAKO tractor.

The algorithm has been compared to some benchmark test, such as the so-called fish tail maneuver (see Figure 10), where the tractor shifts its track and direction by turning, reversing and turning (much like a three-point turn in a car).

As can be seen from Figure 10 the estimate of the position is almost spot on from the beginning, no matter what the starting angle of the filter is. The maximum easting error is 7 cm which is hardly visible on the figure. This is due to the choice of large starting value for the error covariance matrix Q_k and the low noise on the GPS measurement.

Simulation of a turn-test in Figure 11 shows that the algorithm work very well with the HAKO tractor.

The row skip maneuver is quite relevant for an agricultural vehicle as the HAKO tractor. Simulation of a row skip maneuver also shows that the algorithm work very well with the HAKO tractor.

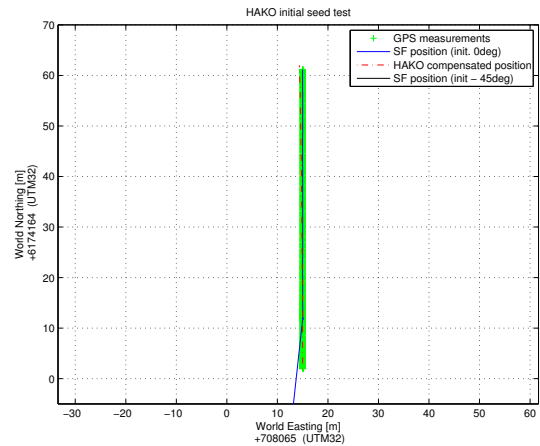


Fig. 9. Simulation of HAKO running straight run to check for convergence of the filter - Position plot.

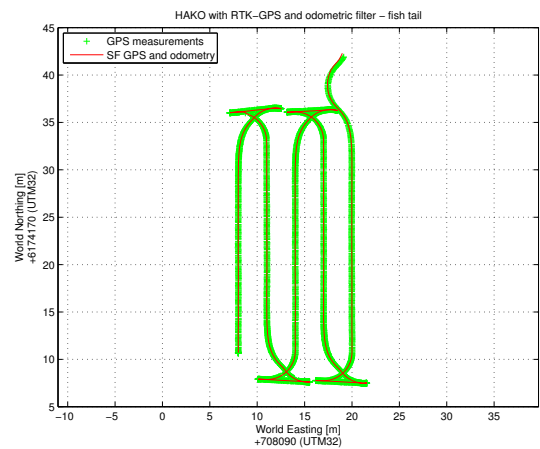


Fig. 10. Simulation of HAKO running fishtail maneuver - Position plot. The result is visually identical to the one found in Reske-Nielsen et al. (2006).

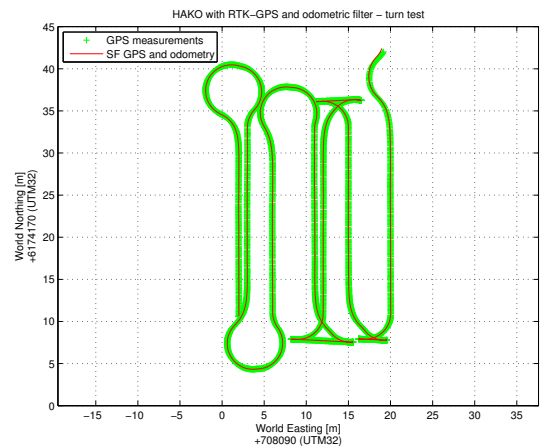


Fig. 11. Simulation of HAKO running turn test maneuver - Position plot.

5.2 MMR

As with the HAKO, the MMR is tested in the simulator in order to understand the capabilities of the filter and make it more likely to succeed in real life tests. However, only real life test is shown here.

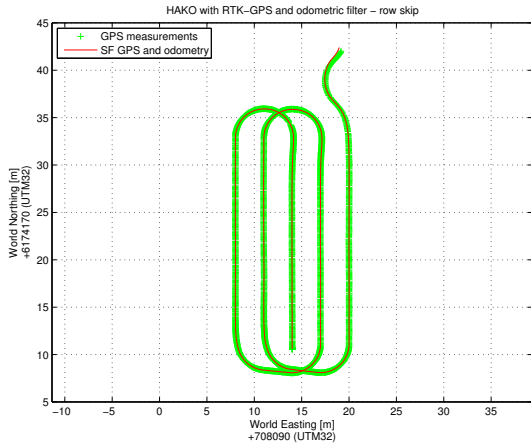


Fig. 12. Simulation of HAKO running row skip maneuver - Position plot.

Parking lot The test run is carried out on a tarmac with different gradients. This proved to be rather interesting, as the directional stability of the robot makes it challenging to make test runs exceeding 50 m.

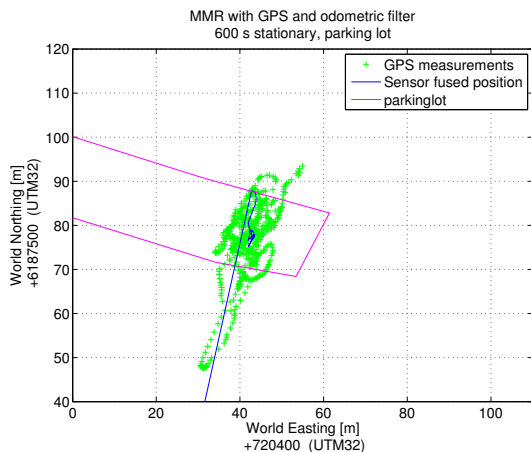


Fig. 13. Real life test of MMR stationary in parking lot - Position plot. The figure shows the drift of the GPS.

First an experiment is conducted where the MMR is kept on a stationary position. The result of the stationary test indicates that there is a substantial drift on the GPS when using it in the parking lot, see Figure 13. The variance is 8.5 m in the northing direction and the 4.3 m in the easting direction. In the stationary situation the position converges nicely and shows a variance of only 0.5 m in the northing direction and 0.2 m in the easting direction. The test of the GPS shows that the variance for the GPS measurement is set fairly low, but since the noise is reduced when the vehicle is driving; the noise is not changed for the later tests.

The direction drifts like the internal odometry. This is not corrected much by the GPS measurement, since the vehicle is not moving, see Figure 14. The drift is 0.06 deg/min which is very satisfactory, considering the odometry is based on a calibrated system using both wheel encodes and gyro. This is what could be expected of a calibrated system, but the low confidence in the GPS makes it interesting to see how it will perform when the vehicle is moving.

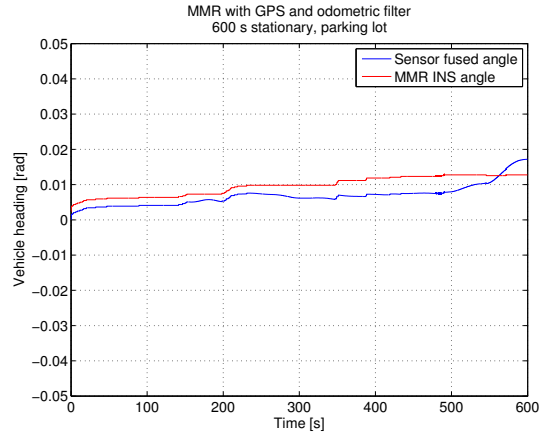


Fig. 14. Real life test of MMR stationary in parking lot - Heading plot. The vehicle heading is drifting as expected, which is also picked up by the Kalman filter.

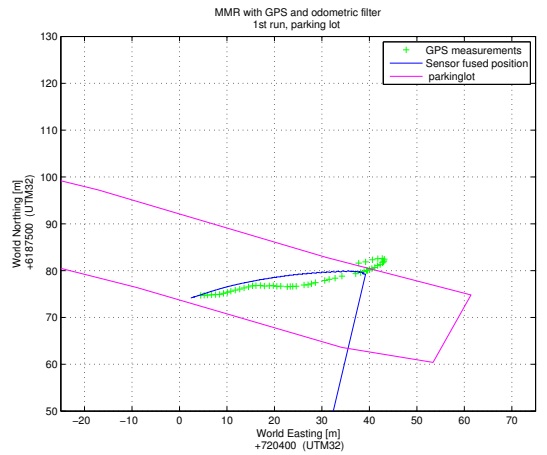


Fig. 15. Real life test of MMR driving in parking lot - Position plot.

Next a new experiment is conducted where the MMR is commanded to run on a straight line. As can be seen from the test result in Figure 15 the filter position converges fine and it tends to follow the GPS measurement as it move forward.

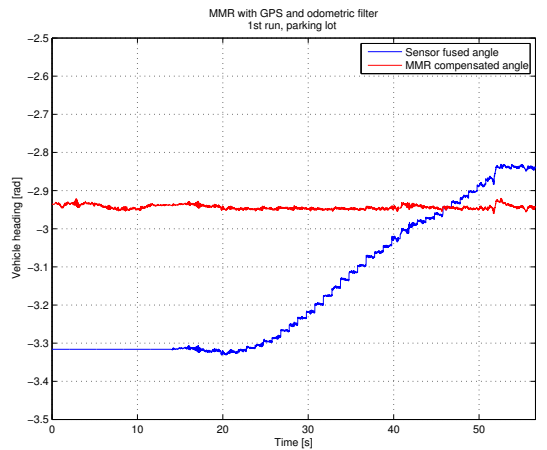


Fig. 16. Real life test of MMR driving in parking lot - Heading plot.

Figure 16 shows some convergence even though the drive is rather short, but that is also partly explained by the initialization value. The filter was initialized with a good initial heading, not more than 15 deg off the true heading. This does account for some of the good performance. When the GPS measurement does not drift too fast, the filter converges to the correct value.

Dyrehaven The runs carried out in Dyrehaven (a wood land area close the university), tests the sensor fusion solution when driving longer stretches and hopefully with better GPS coverage. Figure 17 and Figure 18 show that the filter converges like the simulation predicted. Using the modeled variances for the filter does make the filter converge, but not as fast or stable as the HAKO tractor.

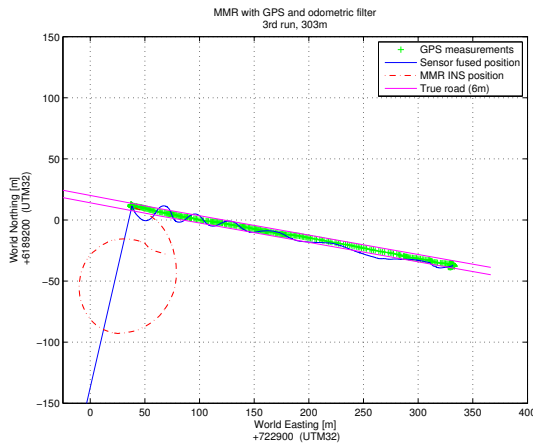


Fig. 17. MMR real life test in Dyrehaven - Position plot. MMR INS position is the internal odometry and gyro based position estimate.

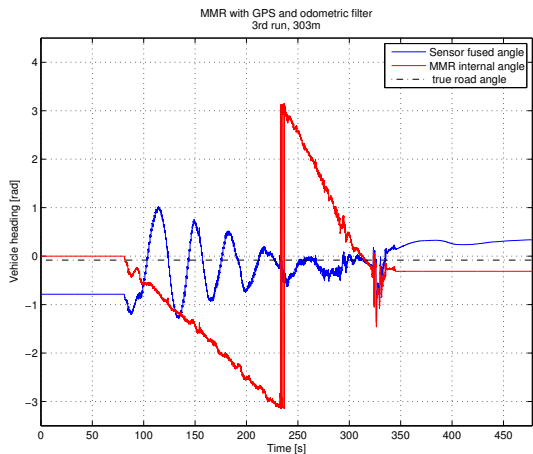


Fig. 18. MMR real life test in Dyrehaven - Heading plot. Final variance is $\sigma = 0.1 \text{ rad}$

The stationary test indicates that the drift on the GPS persists when using it in Dyrehaven. The variance is 2.5 m in the northing direction and 1.9 m in the easting direction, significantly lower than for the parking lot test. Positioning the robot on the road solely using the GPS cannot be done, with precision recorded. The test in Dyrehaven showed that the position could drift off the road when driving. Local guidance from other sensors is therefore needed. The Lat/Lon to UTM conversion algorithm shows the promised precision, taking the precision

of the extraction of the road information from an on-line map into account.

The estimate is not optimal as it is oscillating as the simulations have shown. The variance of the stationary angle taken on the last half of the run is $\sigma_{\text{heading}} = 0.1 \text{ rad}$ and the largest deviation from the road is 5 m with a bias to the right of the GPS measurements and the road.

6. CONCLUSION

In this paper we have applied a toolbox (Kalmtool) on models of two different mobile robot platforms. Kalmtool is a set of MATLAB procedures for state estimation for nonlinear systems. It contains functions for extended Kalman filtering as well as for the two new filters, the DD1 filter and the DD2 filter. It also contains functions for the Unscented (standard and scaled) Kalman filter as well as three versions of particle filters.

One mobile robot is an agricultural vehicle (HAKO tractor) designed for experiments in the fields. The other robot is a Medium Mobile Robot (MMR), which is designed for indoor as well as outdoor experimental purposes.

The result of this work is a series of successful applications of the navigation platform. It has also proven to be a simple, transparent and yet a powerful platform for navigation experiments.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support from Associate Professor Hans Werner Greipentrog and Ph.d. Micheal Nørre-remark for the use of the HAKO tractor at the Department of Agricultural Science at The Royal Veterinary and Agricultural University.

REFERENCES

- Kai O. Arras. The cas robot navigation toolbox, quick guide. Technical report, CAS, KTH, January 2004.
- Magnus Nørregaard, Niels Kjølstad Poulsen, and Ole Ravn. Kalmtool for use with matlab. In *13th IFAC Symposium on System Identification, SYSID03, Rotterdam*, pages 1490–1495, Rotterdam, 2003. IFAC.
- A. Reske-Nielsen, A. Mejnertsen, N. Andersen, O. Ravn, M. Nørre-remark, and H. W. Griepentrog. Multilayer controller for outdoor vehicle. In *Automation Technology for Off-Road Equipment ATOE 2006*, Bonn, Germany, sep 2006.
- Thomas Hanefeld Sejerø, Niels Kjølstad Poulsen, and Ole Ravn. A simulation platform for localization and mapping. In *IFAC conference on SYSID in Newcastle*, 2005.
- Thomas Hanefeld Sejerø, Niels Kjølstad Poulsen, and Ole Ravn. A new evaluation platform for navigation systems. In *16th IFAC World Congress, Prague, Czech Republic*, pages Tu–A18–TO/2, ID: 03891, 2005.
- Rudolph van der Merwe. Quick-start guide for rebel toolkit. Technical report, Oregon Health and Science University, February 2004.