2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Orchard navigation using derivative free Kalman filtering

Søren Hansen, Enis Bayramoglu, Jens Christian Andersen,
Ole Ravn, Nils Andersen and Niels Kjølstad Poulsen

Abstract— **This paper describes the use of derivative free filters for mobile robot localization and navigation in an orchard. The localization algorithm fuses odometry and gyro measurements with line features representing the surrounding fruit trees of the orchard. The line features are created on basis of 2D laser scanner data by a least square algorithm. The three derivative free filters are compared to an EKF based localization method on a typical run covering four rows in the orchard. The Matlab® toolbox Kalmtool is used for easy switching between different filter implementations without the need for changing the base structure of the system.**

Index Terms— **State estimation, Sensor fusion, Robot navigation, Autonomous mobile robots, Localization.**

I. Introduction

Commercial orchards is a very promising area for autonomous robots. A number of routine tasks could be handled by autonomous robots, e.g. spraying with fungicides and insecticides. Precision navigation of tractors using Real Time Kinematics (RTK) GPS has been studied by many researchers e.g. [1], [2] and [3]. Several commercial systems exist, among others Intellisteer by New Holland and Greenstar by John Deere. Also visual guidance and crop row following are well established [4], [5], [6] and [7] and supported by a commercial product 'Eye drive' by Agrocom.

The orchard environment is a relatively organized environment, which is ideal for autonomous robots like the automated Hako tractor (fig. 1). Trees are systematically planted, which makes laser navigation based on the surroundings interesting, since the dense vegetation of the rows of trees make a surface suitable for the laser scanner. Odometry, gyro and features extracted from the laser scanner can be fused to perform the localization. An existing solution using an Extended Kalman filter (see [8]) for localization and navigation has been developed for the HAKO tractor. This solution has the drawback that changes to the nonlinear equations describing the system and its sensor models, imply that new derivatives should be found which can be inconvenient. Therefore a derivative free solution is of interest.

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. The filter implementations outlined in this paper is done using the tool, Kalmtool [9] and [10].

Kalmtool is a collection of MATLAB implementations for estimation and simulation 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.

Fig. 1. The HAKO tractor.

Several toolboxes has been proposed for state estimation over the years, each with a different approach and focus. ReBEL (Recursive Bayesian Estimation Library) [11] is a MATLAB toolkit of functions and scripts, designed to facilitate sequential Bayesian inference (estimation) in general state space models. The Nonlinear Estimation Framework (NEF) [12] is a collection of software routines that facilitates filtering, prediction and smoothing of discrete time systems. In [13] the Bayesian Decision-Making library (BDM) is described. This is a C++ implementation of various Bayesian filters with focus on particle filters.

The CAS Robot Navigation Toolbox [14] is a tool for doing off-line off-board localization and SLAM for mobile robots. The design of the CAS toolbox decouples robot model, sensor models, features and algorithms used giving the user the possibility to modify the toolbox. The OpenSLAM project [15] initiative, is a user driven website, which gathers algorithms and toolboxes. The amount of solutions shows that the field is maturing, and there is still new methods being developed.

S. Hansen, E. Bayramoglu, J. C. Andersen, O. Ravn and N. Andersen is with DTU Electrical Engineering, Technical University of Denmark, Elektrovej B. 326, DK-2800 Lyngby, Denmark {sh,eba,jca,or,naa}@elektro.dtu.dk

N. K. Poulsen is with DTU Informatics, Technical University of Denmark, Richard Petersens Plads B. 305, DK-2800 Lyngby, Denmark nkp@imm.dtu.dk

The paper describes how three different derivative free filters can be used for the task of making the HAKO tractor drive autonomously through the orchard. The localization solution uses the tree rows as measurements to correct the pose estimated by the filters. The paper is divided into the following parts. Firstly, the estimation algorithms are briefly listed. Secondly, the application of mobile robot localization in an agricultural setting is discussed and the finally the three chosen filters are benchmarked with real life data against each other and a localization implementation using an Extended Kalman filter.

II. Estimation algorithms

Consider a system in which the evolution of the state sequence $\{x_k \in \mathbb{R}^n,\ k \in \mathbb{N}\}$ is given by

$$x_{k+1} = f_k(x_k, u_k, v_k) \tag{1}$$

where f_k is a possible nonlinear function of the state, x_k, the input (control) signal, u_k and the process noise, v_k. The process noise is assumed to be a sequence $\{v_k \in \mathbb{R}^n\ k \in \mathbb{N}\}$ of independent and identically distributed (i.i.d.) stochastic vectors.

The objective is to estimate x_k from measurements

$$y_k = g_k(x_k, e_k) \qquad \in \mathbb{R}^m \tag{2}$$

where also g_k is a possible nonlinear function of the state and the measurement noise, e_k. The measurement noise is assumed to be a sequence, $\{e_k \in \mathbb{R}^m\ k \in \mathbb{N}\}$, of i.i.d. stochastic vectors. More specific we seek an estimate of x_k based on all available measurements (and known inputs) $Y_{0:k} = \{(y_i, u_i),\ i = 0, ..., k\}$.

The solution to this problem is embedded in the conditional degree of belief in the state, x_k given the data, $Y_{0:k}$. The problem is then (recursively) to determine the probability distribution function (pdf). $p(x_k|y_{0:k})$. If the initial distribution, $p(x_0)$, is known then the solution can in principle be determined through the recursions:

$$p(x_k|Y_{0:k-1}) = \int_{\Omega_x} p(x_k|x_{k-1})p(x_{k-1}|Y_{0:k-1})dx_{k-1} \tag{3}$$

and

$$p(x_k|Y_{0:k}) = \frac{p(y_k|x_k)}{p(y_k|Y_{0:k-1})}p(x_k|Y_{0:k-1}) \tag{4}$$

These two recursions are related to the dynamic ((3)) and the inference ((4)) step, respectively and can only in special cases be solved analytically. In the linear Gaussian case the pdf. can be parameterized in terms of mean and variance and the recursions results in the well known Kalman filter. The Kalman filter is given by the prediction or the time updates

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k} + Bu_k \tag{5}$$

$$P_{k+1|k} = AP_{k|k}A^T + R_1 \tag{6}$$

and the inference recursion

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1}) \tag{7}$$

$$P_{k|k} = P_{k|k-1} - K_kCP_{k|k-1} \tag{8}$$

The various filters differs in the way they handle the propagation of the distributions through the two nonlinearities, f and g, and how the inference is carried out. The next four filters are all based on the Projection Theorem.

In this case, the prediction in (3) results in (5) and can also be found as an application of calculus for linear operations on Gaussian vectors. The inference recursion in (7) emerge from (4) or as an application of the Projection Theorem.

The Extended Kalman filter is as its name indicate based on an extension of the application of the Kalman filter to the nonlinear case. The Extended Kalman filter (EKF) is based on a standard Taylor expansion of the nonlinear functions and can be regarded as a local approximation. In general the approximation is best for small deviations from the point of linearization.

The divided difference filter exists in a first order version (DD1) and in a second order version (DD2) and is based on Stirlings interpolation formula (see [17]).

The Unscented Kalman filter is based on the (uncented) transformation of a stochastic variable, x, through a nonlinear function, $F(x)$ (see [18]). The standard UKF is based on the approximation mentioned above and the Projection Theorem. In the scaled version of UKF the weight is chosen in a slightly different manner (see [19] or [20] for details).

The linear regression Kalman filter (LRKF) is as the name indicated based on a linear regression (see [21]).

Particle filters comes in several versions and implementations (see e.g. [22] or [23]). In the most basic version (Exp. PF) implemented in the platform the nonlinearities are dealt with by propagating a swarm of particle through the nonlinearities.

III. Autonomous orchard navigation

Navigation through the orchard is done by estimating the pose of the robot based on the commanded velocity and steering angle. This prediction of the pose is corrected using the difference between where a known feature is measured to be and where it should be according to a map. This gives an innovation that is used in the different filters.

The HAKO tractor - see fig. 1, used for the experiments, is described in more detail below.

A. HAKO tractor

The robot used for the experiments in this paper is based on the HAKO Hakotrac 3000 tractor. The tractor has been converted to drive by wire and is hence capable of driving autonomously and it has the following sensors attached in the setup used for orchard navigation.

- **RTK-GPS**
  A high precision RTK-GPS receiver logs the position in UTM coordinates. This is used as the ground truth in comparison with the localization algorithm.
- **Odometry**
  A quadrature encoder is attached to the tractors drive shaft. This measures the revolutions of the shaft and thereby a measure of the length driven by the tractor is found. In addition the angle of the front wheels are

measured. By using these measurements the tractors position can be found by odometry.

- **Gyro**
  A fiber optic gyro measuring the rate of the tractors heading is also attached.
- **Laser scanner**
  The laser scanner attached is a SICK LMS-200 and it has a 180° degree field of view, with a 0.5° resolution. The scanner is placed at the front of tractor 0.5 m over ground.

The data recorded from the tractor is recorded while the tractor is doing online estimation. Since both control signals and sensor readings are saved it is possible to do offline data treatment also. This feature is used for debugging and tuning the estimation algorithm. The data used in this paper is recorded in this way.

B. Test environment

In order to relate measurements from the 2D laser scanner to actual tree rows in the orchard, a-priori knowledge about the tree rows position is needed. This information is saved in a map from which the robot can look up tree row positions when needed in the localization procedure. A simple map which contains starting and ending points of rows is used. The map is formed in the UTM coordinate system. From the map information a straight line representation ($Ax + By + C = 0$) of the orchard rows can be found easy. The fruit tree rows are mapped as well as two boundary hedges, one in the southern region and one to the east. The southern hedge serves an important purpose since it is directed perpendicular to the rest of the rows.

C. Localization

The localization algorithm is based on the state estimate $\mathbf{x}_k$ and odometry measurement $\mathbf{u}_k$ given by

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \qquad \mathbf{u}_k = \begin{bmatrix} L_k \\ \theta_{SA,k} \end{bmatrix} \tag{9}$$

where x_k, y_k and θ_k is the robot pose in the global coordinate system. The control signals L_k and $\theta_{SA,k}$ is the distance driven in one sample and the tractors steering angle. The state- and measurement transition is done by a pair of non-linear stochastic difference equations like equation 1 and equation 2 on page 2.

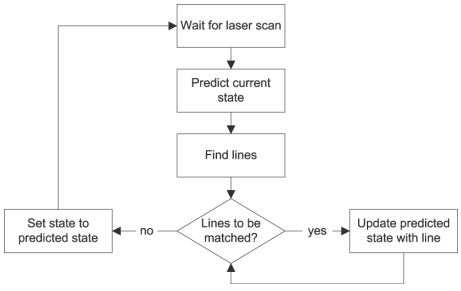

Fig. 2. The execution loop of the estimator.

Figure 2 shows the overall idea of the localization algorithm. The different parts a described in greater detail below. The algorithm starts when a laser scan is received from the SICK scanner. Then the current pose of the robot is estimated using the vehicle model. From this the line features of the scan is found. If any of these can be matched to corresponding features in the map the pose estimate is updated using this information.

D. Vehicle model

The model used to derive the Ackermann odometry is called tricycle drive. This is a way of simplifying the model of a standard four wheel drive geometry as seen on fig. 3. The two front wheels are collected to one which is turning around the common turning point. Odometry is the method of calculating the position of a vehicle from heading and speed changes. On the HAKO tractor, the speed is measured using an encoder on the gearbox. The heading is measured by a gyro and can also be calculated from the steering angle, which is measured by an absolute encoder, returning the steering angle.

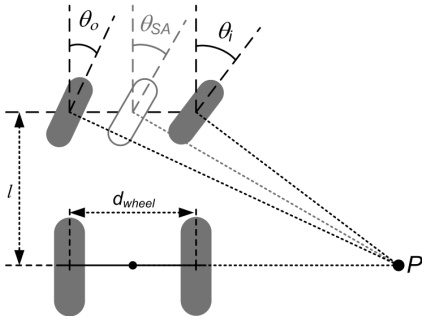

Fig. 3. Figure showing the change in heading and position of a Ackermann steered vehicle.

On the robot the steering angle θ_{SA} is given. The change in heading angle is derived from the Ackermann geometry. The equation is:

$$\Delta\theta = \frac{L_k \tan(\theta_{SA})}{l} \tag{10}$$

where L_k is the distance the robot has driven in one sample and l is the distance between the front and rear axle of the HAKO tractor.

The robots new position and angle can now be found as:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \begin{bmatrix} L_k \cos\left(\theta_k + \frac{\Delta\theta}{2}\right) \\ L_k \sin\left(\theta_k + \frac{\Delta\theta}{2}\right) \\ \Delta\theta \end{bmatrix} + \mathbf{G}\,\mathbf{w}_k \tag{11}$$

where $\mathbf{G}$ is a linearized function to relate the noise from the control signal measurements to the states.

E. Process noise model

The process noise is the noise originating from the measurement of the control signals - in this case - the odometry.

The process noise is the covariance of the control signal $\mathbf{u}_k$ and is denoted $\mathbf{Q}_k$.

$$\mathbf{Q}_k = \begin{bmatrix} \sigma_L^2 & 0 \\ 0 & \sigma_{\theta,SA}^2 \end{bmatrix} \tag{12}$$

For the error covariance to be dependent of the travelled distance, the variances must be proportional to the travelled distance.

The variance for the linear displacement is given by [24]:

$$\sigma_L^2 = \epsilon_L^2 |L| \tag{13}$$

where ϵ_L is the standard deviation error from one meter of travel.

The variance of the heading θ is found using equation 14 below. The equation is based on equation 10, which determines the change in direction.

$$\sigma_{\theta,SA}^2 = \left(\frac{\tan\left(\epsilon_A + \epsilon_{SA}|\theta_{SA}|\right)}{l} \right)^2 |L| \tag{14}$$

ϵ_A is the angle error standard deviation caused by driving one meter straight ahead and ϵ_{SA} is the contribution from the steering angle while turning.

F. Feature extraction

In this section the feature extraction will be presented. The SICK laser scanner returns 361 different range measurements spread out in an $180°$ fan originated in the center of the scanner. The principle is illustrated in fig. 4

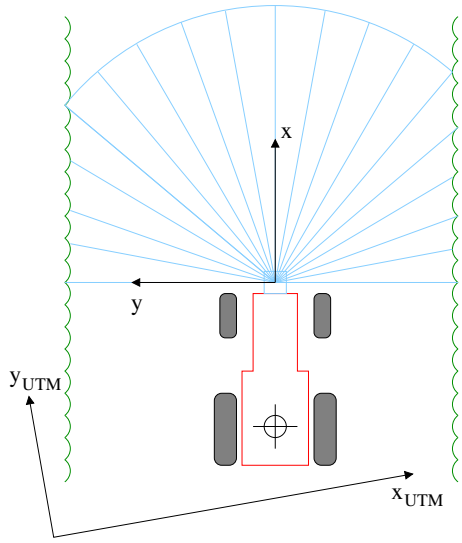

Fig. 4. The HAKO tractor driving between rows in the orchard. The laser scanner returns distances from the origo of the $\{x, y\}$ coordinate system to the tree rows. The maximal distance measured is 8.1 m.

By using knowledge about the tractors pose the range scans are transformed into global UTM coordinates. These laser measurements are pre-processed by calculating the numeric distance to the predicted line, based on the tractor position and the map. If the distance is below a pre-defined threshold, the measurement is marked and used.

The distance d_l between the laser scan range measurement point-representation and predicted line is calculated using equation 15.

$$d_l = \frac{Ax_{UTM} + By_{UTM} + C}{\sqrt{A^2 + B^2}} \tag{15}$$

where the equation involving A, B, C is the normal equation of the a priori line and x_{UTM}, y_{UTM} is the range measurement point-representation.

All the measurements which are inside the interval on either side of the predicted line are now used, to find an estimate of the measured line using linear regression. When the line is found further checks are performed to see if the line estimate is sufficiently accurate to be used by the filter.

G. Line features

A laser scan usually contains several line features, and all of them are passed on for further processing. The description below concerns directly the lines extracted with a priori knowledge. To simplify the Kalman filter the features are represented in polar coordinates.

Feature vector

The features used are the rows that makes up the orchard. If the distance from the center of the robot to the row is denoted by d_m and the bearing of the row in UTM coordinates is denoted θ_m the feature vector is given by:

$$f(z_k) = f_k^1, f_k^2, \ldots = \left\{ \begin{bmatrix} d_{m,k}^1 \\ \theta_{m,k}^1 \end{bmatrix}, \begin{bmatrix} d_{m,k}^2 \\ \theta_{m,k}^2 \end{bmatrix}, \ldots \right\} \tag{16}$$

Fig. 5 shows the definition of the distance and angle measured.

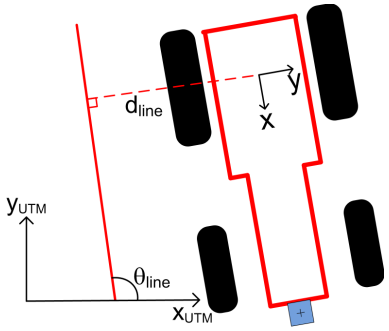

Fig. 5. Measured line shown in robot and UTM coordinate system.

Since the rows found are represented as $A_{m,i}x_{UTM} + B_{m,i}y_{UTM} + C_{m,i} = 0$ initially the distance and angle of each feature is found as:

$$\begin{bmatrix} d_m \\ \theta_m \end{bmatrix} = \begin{bmatrix} \dfrac{A_m x_{UTM} + B_m y_{UTM} + C_m}{\sqrt{A_m^2 + B_m^2}} \\ \arctan\left(\dfrac{-A_m}{B_m}\right) - \theta \end{bmatrix} \tag{17}$$

The θ_m is subtracted with the state estimate of the angle θ to relate the line to the current state. Since the same is done to the measurement vector, this does not result in a difference in the update vector.

IV. Results

The results presented in this section are based on recorded data from the HAKO tractor, during an autonomous drive in the orchard. The log is spanning four rows (see figure 6), which makes is possible to show the position update when re-finding the rows at the northern and southern positions of the run.

Three derivative-free filters has been tested for the localization task. The RTK-GPS measurements are used as ground truth for the position estimates, which all are kept in a global reference frame for easy comparison. The tested filters are

- Second order divided difference filter (dd2).
- Linear regression filter (linreg).
- Unscented Kalman filter (ukf).

The three filters are benchmarked against a localization implementation using EKF.

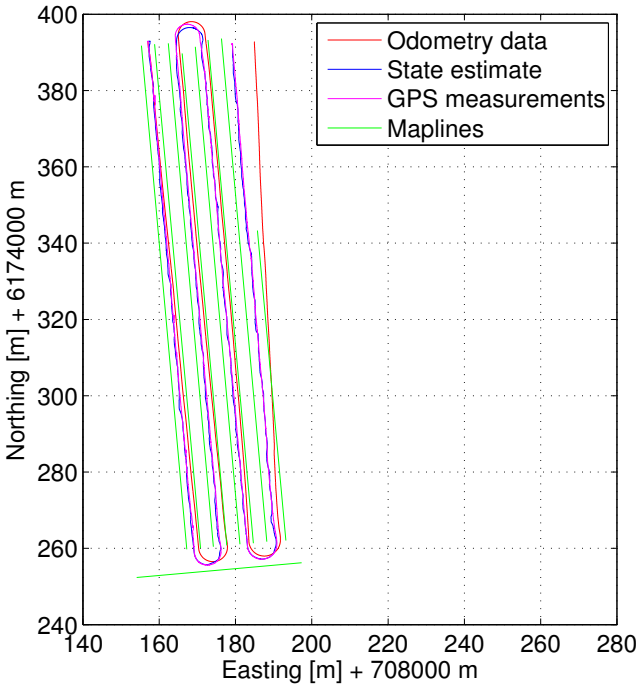

Fig. 6. Plot of the full 4-row run through the orchard. The state is estimated by the dd2 filter.

The localization algorithm using all four filters are able to complete the run through the orchard in a satisfactory way. They are able to keep the robot in track and close to the ground truth. In figure 6 the full run of the orchard is shown for the dd2 filter. The robot starts in the upper left corner and ends in the upper right corner of the run. In the end it is seen that the odometry estimate has drifted due to wheel slippage and therefore ends about 5 meters from true robot position. However the state estimate is only about 20 cm from the correct end position.

Figure 7 shows the time development of the standard deviations of the states, estimated by the three filters under similar noise initializations. They show remarkable similarity. The deviation on the estimate of y, σ_y, is raising except

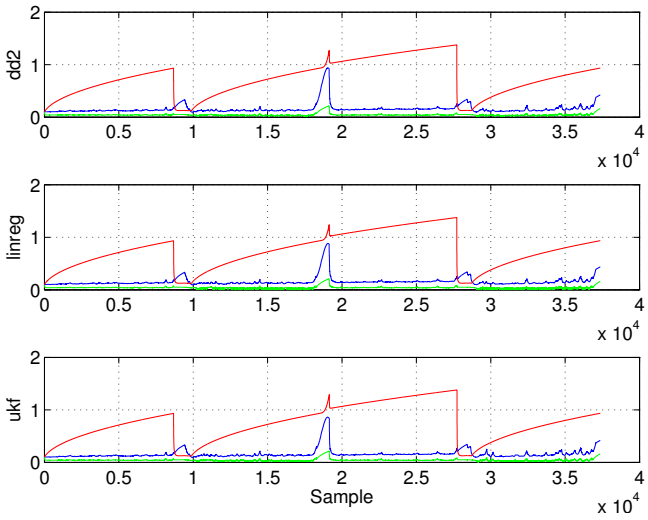

Fig. 7. Square root of covariances of the three estimated states for the derivative free filters.
Legend: | σ_x | Blue | σ_y | Red | σ_θ | Green |

in the lower corners around sample 0.9×10^4 and sample 2.8×10^4 where the robot is in the most southern part of the orchard. At these two points a tree row perpendicular to the others ensure that measurements in the y-direction (north to south direction) is possible and hence the belief in the pose estimate is increased.

Filter type	Avr. CPU time	Sum of GT. distance
ekf	71.14 s	147.07
dd2	74.78 s	154.01
linreg	74.60 s	151.70
ukf	75.96 s	163.10

TABLE I
AVERAGE COMPUTATIONAL TIME AND SUM OF DISTANCE DEVIATION FROM GROUND TRUTH.

In table I the average computational time for the four filters is listed. This is the time it takes each filter to iterate through 2500 samples corresponding to a 50 meters drive with two rows visible by the laser scanner. The time is an average over 50 experiments. The filters use almost the same amount of time with the ukf being marginally slower. All three derivative free filters are around 4 s slower in computing the pose estimate than the EKF solution.

In the last column of table I the sum of the distance deviation between the state estimate and the ground truth for the complete orchard run is shown. The EKF performs better again in this measure, however the difference between EKF and the derivative free solutions is within an acceptable boundary. The Unscented Kalman filter has again the highest deviation among the derivative free solutions. In figure 8 the position estimate of the ukf is compared with the linear regression filter. Both filters distort the turn compared to the GPS. However, the linear regression filter estimates a

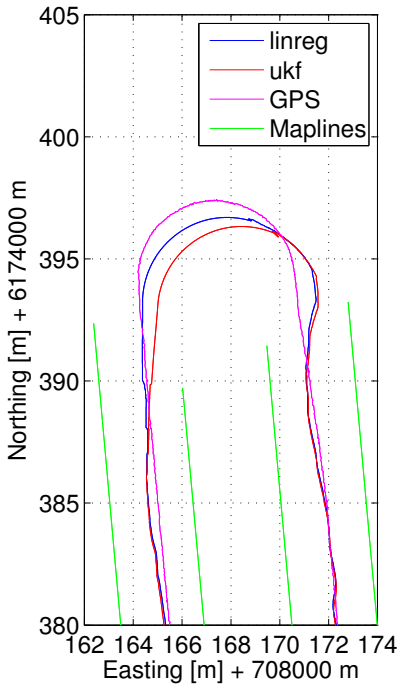

Fig. 8. Comparison between Unscented Kalman filter and linear regression filter at the northern turn of the orchard.

trajectory closer to the GPS.

V. Conclusion

A localization solution using three derivative free filters has been implemented, enabling the HAKO tractor to drive autonomously through the orchard using laser scanner features to correct the pose estimate. These three filters has been benchmarked against an Extended Kalman based solution. All three filters perform equally both with regard to precision and computational time and the derivative free filters matches the existing solution using an Extended Kalman filter in performance, but is more flexible towards changes in the system and measurement model descriptions.

The switching between the different estimator solutions without having to rewrite the model is made easy by the use of the Kalmtool toolbox and the visualization features makes the debugging less tedious.

The toolbox is available for download at: *http://server.elektro.dtu.dk/www/or/kalmtool/*

VI. Acknowledgements

The faculty of Life Sciences in Copenhagen University is great-fully acknowledged for their cooperation and help with the data collection and tests done in connection to this work. Professor Hans-Werner Griepentrog, University of Hohenheim, Stuttgart, Germany is also great-fully acknowledged for his involvement in this work and the work with HAKO tractor.

References

[1] W. Larsen, G. Nielsen, and D. Tyler, "Precision navigation with gps," *Computers and Electronics in Agriculture*, vol. 11, no. 1, pp. 85–95, 1994.

[2] M. Ehrl, H. Auernhammer, W. Stempfhuber, M. Demmel, and M. Kainz, "Autotrac - accuracy of a rtk dgps based autonomous vehicle guidance system under field conditions," *Proceedings of the International Conference on Automation Technology for Off-road Equipment, ATOE 2004*, pp. 274–282, 2004.

[3] M. Nørremark, H. Griepentrog, J. Nielsen, and H. Søgaard, "The development and assessment of the accuracy of an autonomous gps-based system for intra-row mechanical weed control in row crops," *Biosystems Engineering*, vol. 101, no. 4, pp. 396–410, 2008.

[4] J. Reid and S. Searcy, "Vision-based guidance of an agriculture tractor," *IEEE Control Systems Magazine*, vol. 7, no. 2, pp. 39–43, 1987.

[5] J. Billingsley and M. Schoenfisch, "Vision-guidance of agricultural vehicles," *Autonomous Robots*, vol. 2, no. 1, pp. 65–76, 1995.

[6] N. Tillett, T. Hague, and S. Miles, "Inter-row vision guidance for mechanical weed control in sugar beet," *Computers and Electronics in Agriculture*, vol. 33, no. 3, pp. 163 – 177, 2002.

[7] M. Kise, Q. Zhang, and F. Rovira Más, "A stereovision-based crop row detection method for tractor-automated guidance," *Biosystems Engineering*, vol. 90, no. 4, pp. 357–367, 2005.

[8] L. V. Mogensen, S. Hansen, J. C. Andersen, O. Ravn, N. A. Andersen, M. Blanke, and N. K. Poulsen, "Kalmtool used for laser scanner aided navigation," in *15th IFAC Symposium on System Identification (SYSID)*, jul 2009.

[9] L. V. Mogensen, S. Hansen, O. Ravn, and N. K. Poulsen, "Comparing mobile robot localisation algorithms using kalmtool," in *15th IFAC Symposium on System Identification (SYSID)*, jul 2009.

[10] E. Bayramoglu, S. Hansen, O. Ravn, and N. K. Poulsen, "Derivative free filtering using kalmtool," in *Proceedings of the 13th International Conference on Information Fusion*, July 2010.

[11] R. van der Merwe, "Quick-start guide for rebel toolkit," Oregon Health and Science University, Tech. Rep., February 2004.

[12] O. Straka, M. Flídr, J. Duník, M. Šimandl, and E. Blasch, "Nonlinear estimation framework in target tracking," in *Proceedings of the 13th International Conference on Information Fusion*, July 2010.

[13] V. Šmídl, "Software analysis unifying particle filtering and marginalized particle filtering," in *Proceedings of the 13th International Conference on Information Fusion*, July 2010.

[14] K. O. Arras, "The cas robot navigation toolbox, quick guide," CAS, KTH, Tech. Rep., January 2004.

[15] C. Stachniss, U. Frese, and G. Grisetti, "Open slam - give your algorithm to the community," Web site - www.openslam.org, October 2008, authors assosiated with University of Freiburg.

[16] M. Nørgaard, N. K. Poulsen, and O. Ravn, "Kalmtool for use with matlab," in *13th IFAC Symposium on System Identification, SYSID03, Rotterdam*. Rotterdam: IFAC, 2003, pp. 1490–1495.

[17] ——, "New development in state estimation for nonlinear systems," *Automatica*, vol. 36, pp. 1627–1638, 2000.

[18] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceeding of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[19] S. Julier, "The scaled unscented transformation," *Proceedings of the American Control Conference*, pp. 4555–4559, May 2002.

[20] E. Wan and R. van der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000, pp. 153–158.

[21] H. B. Tine Lefevbre and J. D. Schutter, "Kalman filters for non-linear systems: a comparison of performance," *International Journal of Control*, vol. 77, no. 7, pp. 639–653, 2004.

[22] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, February 2002.

[23] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, "The unscented particle filter," Cambridge University Engineering Department, Tech. Rep., August 2000.

[24] L. Kleeman, "Advanced sonar and odometry error modeling for simultaneos localisation and map building," *Proceedings of the 2003 IEEE International Conference on Intelligent Robots and System, Las Vegas*, 2003.