IFAC

Implementation and Comparison of Attitude Estimation Methods for Agricultural Robotics

Trygve Utstumo [*,**] **Jan Tommy Gravdahl** [*]

[*] *Department of Engineering Cybernetics, Norwegian University of Science and Technology, NO-7491 Trondheim, Norway.*
jan.tommy.gravdahl@itk.ntnu.no, trygve.utstumo@itk.ntnu.no
[**] *Adigo AS, Berghagan 3, NO-1405 Langhus, Norway.*

Abstract: The field of precision agriculture increasingly utilize and develop robotics for various applications, many of which are dependent on high accuracy localization and attitude estimation. Special attention has been put towards full attitude estimation by low-cost sensors, in relation to the development of an autonomous field robot. Quaternions have been chosen due to its continuous nature, and with respect to applications in the pipeline with on other platforms.

The performance and complexity of two approaches to attitude estimation has been investigated: One Multiplicative Extended Kalman Filter (MEKF) and one non-linear observer. Both were implemented on an ARM Cortex M3 microcontroller with sensors for a Attitude Heading Reference System (AHRS), and benchmarked towards a relative high grade commercial AHRS device.

The relative computational burden of the MEKF have been underlined, by execution times more than 10 times those of the non-linear estimator. The implementation complexity is also significantly lower for the non-linear observer, which facilitate test and verification through more transparent software.

Keywords: Instrumentation and Sensing; Robotics and Mechatronics for Agricultural Automation; Precision Agriculture

1. INTRODUCTION

Modern agriculture is increasingly utilizing advanced technology to automate and better manage its production processes. The use of autonomous systems for weed control is a research field with growing interest, and several autonomous systems have been demonstrated, where some are presented in the review by Slaughter et. al. Slaughter et al. (2008).

Adigo is developing a mobile robot, illustrated in Figure 1, for research on precision agriculture[1]. Building on previous experience with autonomous robots, the attitude estimation is given special attention as part of the localization and navigation for the robot. Attitude heading reference systems (AHRS) are widely utilized in other applications of autonomous ground vehicles, and provide important input to the localization sensor fusion.

The work presented can facilitate customized and better integrated solutions with attitude estimation and enable the use of low cost sensors.

1.1 Multiplicative Extended Kalman Filter

The survey by Crassidis, Markley and Cheng Crassidis et al. (2007) provides a good background and review of

Fig. 1. A robot developed for autonomous N_2O measurements on cereal fields. The robots autonomy is currently under development where efficient attitude estimation is a focus.

various attitude filters, observers and smoothers. For a singularity free representation of attitude we have considered quaternion estimators.

The Extended Kalman filter (EKF) has become the workhorse of attitude estimation, largely through the research effort and numerous applications in space exploration. There are numerous variations on how to implement an EKF for quaternion estimation, especially in the update step where both quaternion addition and multipli-

---

[1] Consortium research program "Multisensory Precision Agriculture - Improving yields and reducing environmental impact" sponsored by the Norwegian Research Council [207829].

cation can be utilized. Multiplication should however be preferred Shuster (1993).

Implementing multiple vector measurements directly in the MEKF is not trivial, and is accompanied with a complex set of tuning parameters Markley et al. (2003). It is however quite straightforward to implement a quaternion measurement in the MEKF. Using the QUEST Shuster and Oh (1981), or an equivalent algorithm, for preprocessing measurement vectors greatly simplify the filter interface. The QUEST algorithm provides support for an arbitrary number of measurement vectors, and the QUEST covariance matrix can directly feed the Kalman $\mathbf{R}$ matrix Shuster (1990),Crassidis et al. (2007).

1.2 Non-linear observer: Explicit Complementary Filter

The EKF does have a number of drawbacks: Implementation is not straight-forward, the numerous parameters require tuning, it is computationally expensive, and it is usually difficult to prove its convergence Martin and Salaun (2010).

Many of these issues can be addressed by nonlinear observers, and especially the stability properties can be proven, using Lyapunov-based methods. A significant step was taken with the *Explicit Complementary Filter* proposed by Hamel and Mahony (2006) and refined in Mahony et al. (2008). It makes use of the vector measurements directly in body frame, and also includes gyro bias estimates. The filter provides near global stability. The concept has later been extended to include time-varying reference vectors by Hua (2010) and Grip et al. (2011). For systems with low accelerations the filter performs as well as these later designs (Hua, 2010), and is thus suitable for wheeled robotic applications with slow dynamics.

2. MODELLING

We operate with three coordinate frames, where our reference frame is North-East-Down (NED), the robot body frame is defined as forward-right-down (BODY) and the instrument frame depending on how the sensor is mounted in the robot (I). The frames are indicated by subscripts n, b and i respectively.

2.1 Unit Quaternions

Rotations and attitude are represented by unit quaternions, where the scalar is defined as the first element in

$$\mathbf{q} = \eta + i\epsilon_2 + j\epsilon_2 + k\epsilon_3 = [\eta, \boldsymbol{\epsilon}]^T \qquad (1)$$

Quaternion multiplication is expressed by:

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} \eta_p \\ \boldsymbol{\epsilon_p} \end{bmatrix} \otimes \begin{bmatrix} \eta_q \\ \boldsymbol{\epsilon_q} \end{bmatrix} = \begin{bmatrix} \eta_p \, \eta_q - \boldsymbol{\epsilon_p^T} \, \boldsymbol{\epsilon_q} \\ \eta_p \, \boldsymbol{\epsilon_q} + \eta_q \, \boldsymbol{\epsilon_p} + \boldsymbol{\epsilon_p} \times \boldsymbol{\epsilon_q} \end{bmatrix}. \qquad (2)$$

Consecutive rotations by quaternions are done by postmultiplication, in contrast to rotation matrices. Thus, the rotation from NED to I, can be composed by rotations from NED to BODY to I as $\mathbf{q}_n^i = \mathbf{q}_n^b \otimes \mathbf{q}_b^i$

2.2 Angular velocity

For a rotation from NED to BODY, the kinematic differential equation is given by Egeland and Gravdahl (2002)

$$\dot{\mathbf{q}} = \tfrac{1}{2} \left[0, \boldsymbol{\omega}_n \right]^T \otimes \mathbf{q} = \tfrac{1}{2} \mathbf{q} \otimes \left[0, \boldsymbol{\omega}_b \right]^T \qquad (3)$$

where $\boldsymbol{\omega}_n$ and $\boldsymbol{\omega}_b$ are rotational velocities.

2.3 Measurements

The AHRS receives measurement vectors from a MEMS accelerometer, $\mathbf{a}_b$, and magnetometer, $\mathbf{m}_b$, in BODY frame. They are normalized and compared with their reference vectors in NED frame, $\mathbf{a}_n$ and $\mathbf{m}_n$. The MEMS gyro is modelled with a bias in body frame as $\boldsymbol{\omega}_b = \boldsymbol{\omega}_{actual} + \mathbf{b}$

3. MULTIPLICATIVE EXTENDED KALMAN FILTER

Representing the full quaternion in the filter would lead to a singularity in the co-variance matrix $\mathbf{P}$, which results in numerical errors and possibly negative eigenvalues in $\mathbf{P}$, Shuster (1993).

An intuitive solution is to leave out the scalar element, η, of the quaternion. Since the quaternion is of unit length, it can be reconstructed by

$$\mathbf{q}(\boldsymbol{\epsilon}) = \begin{bmatrix} \sqrt{1 - \|\boldsymbol{\epsilon}\|^2} \\ \boldsymbol{\epsilon} \end{bmatrix} \qquad (4)$$

With this modification singularities arise at multiples of π. By only representing the rotation error in the filter, $\delta\boldsymbol{\epsilon}$, the singularities are less likely to occur. Error representations with better margins are described by Markley et al. (2003), which should be considered for a robust implementation. The resulting state vector, $\mathbf{x} = [\delta\boldsymbol{\epsilon} \,;\, \mathbf{b}]$ maintains the error of each update, and the gyro bias estimate.

Update	$\bar{\mathbf{q}}(k) \to \hat{\mathbf{q}}(k)$
Error est.	$\delta\mathbf{q}(k) = \begin{bmatrix} \delta\eta \\ \delta\boldsymbol{\epsilon} \end{bmatrix} = \hat{\mathbf{q}}^{-1}(k) \otimes \mathbf{q}_{ref}(k)$
Innovation	$\Delta\mathbf{x} = \mathbf{K}\,\delta\boldsymbol{\epsilon}$
State update	$\hat{\mathbf{q}}(k) = \mathbf{q}\left(x_{[1:3]}(k) \right) \otimes \bar{\mathbf{q}}(k)$
Reset	$\mathbf{x}_{[1:3]} = \mathbf{0}$

Table 1. Details of the MEKF State update, $\bar{\mathbf{q}}(k)$ is the prediction and $\hat{\mathbf{q}}(k)$ is the posterior.

The essence of the MEKF is how the state update is performed by quaternion multiplication. The error is calculated relative to the QUEST position estimate, and is represented by its vectorial part $\boldsymbol{\epsilon}$. This allows the three dimensional error representation to construct the Kalman matrices, $\mathbf{P, K}$ and $\mathbf{H}$ without singularities. The details of the MEKF update in discrete time is shown in Table 1.

The full singularity free unit quaternion represent the attitude estimate and is used in the nonlinear state propagation, through the kinematic equation (3), discretized by Euler's method Crassidis et al. (2007); Markley et al. (2003); Lefferts et al. (1982); Shuster (2009).

4. THE EXPLICIT COMPLEMENTARY FILTER

The Explicit Complementary Filter described by Mahony et al. (2008), for magnetometer and accelerometer input can be presented as:

$$\boldsymbol{\sigma} = k_a \left(\mathbf{a}_b \times \hat{\mathbf{a}}_b \right) + k_m \left(\mathbf{m}_b \times \hat{\mathbf{m}}_b \right) \qquad (5a)$$

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2} \hat{\mathbf{q}} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{gyro} - \hat{\mathbf{b}} + k_p \boldsymbol{\sigma} \end{bmatrix} \qquad (5b)$$

$$\dot{\hat{\mathbf{b}}} = -k_I \boldsymbol{\sigma} \qquad (5c)$$

Where $\boldsymbol{\sigma}$ is the filter correction term, and $k_{[p,I,a,m]}$ are the gains for correction, bias integration and weights on accelerometer and magnetometer measurements. The reference vectors $\hat{\mathbf{a}}_b$ and $\hat{\mathbf{m}}_b$ are found by rotating the reference in NED frame by the transposed rotation estimate, $\mathbf{R}(\mathbf{q})^T$.

$$\hat{\mathbf{a}}_b = \mathbf{R}(\hat{\mathbf{q}})^T \mathbf{a}_i / |\mathbf{a}_i|, \quad \hat{\mathbf{m}}_b = \mathbf{R}(\hat{\mathbf{q}})^T \mathbf{m}_i / |\mathbf{m}_i| \qquad (6)$$

The magnetometer can have large influx of noise, especially in vehicles with electrical motors. This problem is well known and different solutions are proposed in literature.

To minimize the impact of this it is possible to reduce the weighting of the magnetometer in periods with high noise on the magnetometer,(Mahony et al., 2008). With more permanent noise on the magnetometer one can limit its effect to only the yaw rotation, (Martin and Salaun, 2010). This can be done by aligning the magnetometer cross product with the measured accelerometer vector, changing equation (5a) to:

$$\sigma_m = k_a \left(\mathbf{a}_b \times \hat{\mathbf{a}}_b \right) + k_m \left(\left(\mathbf{m}_b \times \hat{\mathbf{m}}_b \right)^T \mathbf{a}_b \right) \mathbf{a}_b \qquad (7)$$

Local exponential stability can be shown with this modification, but it complicates the analysis for region of attraction, (Martin and Salaun, 2010). Hua (2010) confirms the insulation of magnetic perturbations from roll and pitch in simulations, and do comparisons with the update in equation (5a). A price to pay for this modification, is increased error amplitude from accelerations.

5. HARDWARE

The implementation of the filter algorithms investigated has been done on the AHRS CHR-6dm by CH Robotics, and benchmarked towards the Microstrain 3DM-GX3-25 ARHS.

The two sensors have been aligned and mounted to an aluminum bar, with double-sided tape, to minimize magnetic disturbances.

The CHR-6dm AHRS was chosen because of its Open Source firmware, potent ARM Cortex-M3 processor and its low cost. The individual sensors are surface mounted to the PCB, as seen in Figure 2. The accelerometers and gyros are mounted in agreement with the BODY-frame, whereas the magnetometer is constructed with the z-axis pointing up, which result in the following rotation from the magnetometer instrument frame:

$$\mathbf{m}_b = \mathbf{R}_b^i \mathbf{m}_i = [\mathbf{m_y} \quad \mathbf{m_x} \quad -\mathbf{m_z}]_i^T \qquad (8)$$

The gyro and accelerometers are analog devices, sampled by an AD-converter at 400Hz. The magnetometer is connected over the I2C bus, and reports it's measurements at 87Hz.
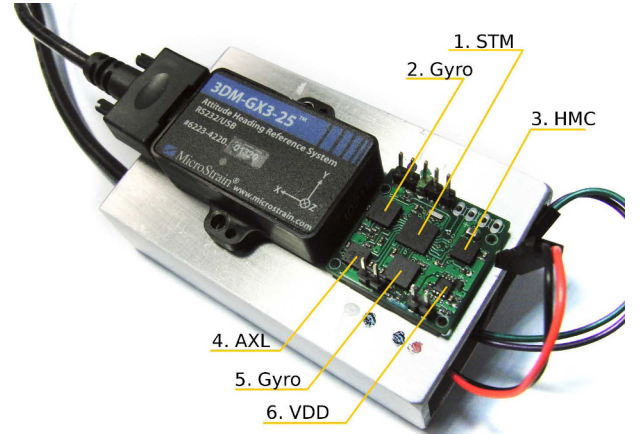

Fig. 2. The components of the CHR-6dm are: **1.** Microcontroller, STM32F103T8 **2.** Gyro Pitch-Roll, LPR510AHL **3.** Magnetometer, HMC5843 **4.** Accelerometer, ADXL335 **5.** Gyro Yaw, LY510AHL **6.** 3.3V Voltage regulator

The gyro bias, $\mathbf{b}$, was measured on several devices, and in various temperatures. The bias values varied in tests in the range of 0.004 to 0.120 rad/s.

The magnetometer measurements are corrected both by a scaling factor and a constant bias, calculated from an initial calibration routine.

6. FIRMWARE DEVELOPMENT

The algorithms for attitude estimation have been developed by first prototyping the algorithms in Matlab with recorded data, then implemented in C both on the PC and then on the microcontroller. The filter parameters were tuned in the Matlab filter prototypes using recorded sensor data.

The QUEST algorithm was implemented on the basis of Shuster (2006) and the flow chart in Takahashi et al. (2009), for two input vectors. Further details on this can be found in Utstumo (2011).

The MEKF was implemented with QUEST as a preprocessor as described in Mahony et al. (2008). The detailed steps of the algorithm are shown in Table 2. The covariance matrix, $\mathbf{P}$, is treated by its block-diagonal elements, $\mathbf{P}_a$, $\mathbf{P}_b$, and the correlation blocks $\mathbf{P}_c$. This further enables us to exploit the simple form of $\mathbf{H}$ to simplify the covariance propagation and reduce the number of matrix operations, as described by Markley et al. (2003).

The covariance matrix returned from the QUEST-algorithm is presented directly to the Kalman filter as the measurement covariance matrix, $\mathbf{R}$. The $\mathbf{P}$ matrix is set initially large on the error estimate, $\mathbf{P}_a(0) = \text{diag}(100)$, to quickly converge to the correct attitude, while the initial bias estimate dynamics are limited by setting, $\mathbf{P}_b(0) = \text{diag}(0.1)$.

The time step, Δt, is calculated between each step by using the internal timer `TIM3` on the microcontroller.

The Explicit Complementary filter is expressed in three equations (5), and those three equations are implemented

Step	Equations
Update step	(87 Hz)
Quest	$[\mathbf{q}_q, \mathbf{R}_q](k) = \text{QUEST}(\mathbf{a_n}, \mathbf{m_n}, \mathbf{a_b}, \mathbf{m_b}, \sigma_\mathbf{a}, \sigma_\mathbf{m})$
Kalman gain	$\mathbf{K}(k) = \begin{bmatrix} \bar{\mathbf{P}}_a \\ \bar{\mathbf{P}}_c \end{bmatrix}^T [\bar{\mathbf{P}}_a + \mathbf{R}_q]^{-1}$
P update	$\hat{\mathbf{P}}(k) = \bar{\mathbf{P}}(k) - \mathbf{K}(k)[\bar{\mathbf{P}}_a \ \bar{\mathbf{P}}_c]$
Error estimate	$\delta\mathbf{q}(k) = \bar{\mathbf{q}}'(k) \otimes \mathbf{q}_q(k)$
Innovation	$\hat{\mathbf{x}}(k) = \bar{\mathbf{x}}(k) + \mathbf{K}(k)\,\delta\boldsymbol{\epsilon}_b$
State update	$\hat{\mathbf{q}}(k) = \bar{\mathbf{q}}(k) \otimes \begin{bmatrix} \sqrt{1 - |\hat{\mathbf{x}}_{[1:3]}(k))|} \\ \hat{\mathbf{x}}_{[1:3]}(k) \end{bmatrix}$
Normalize	$\hat{\mathbf{q}}(k) = \hat{\mathbf{q}}(k) \,/\, |\hat{\mathbf{q}}(k)|$
Reset	$\hat{\mathbf{x}}_{[1:3]}(k) = [0 \ \ 0 \ \ 0]^T$
Propagate	(400 Hz)
Propagate state	$\bar{\mathbf{q}}(k+1) = \hat{\mathbf{q}}(k) + \frac{\Delta t}{2}\left(\hat{\mathbf{q}}(k) \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_b - \hat{\mathbf{x}}_{[4:6]}(k) \end{bmatrix} \right)$
Linearization	$\mathbf{F} = \begin{bmatrix} -[\boldsymbol{\omega}_b \times] & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \ \mathbf{G} = \begin{bmatrix} -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$
Propagate P	$\bar{\mathbf{P}}(k+1) = \bar{\mathbf{P}}(k) + \Delta t \left(\mathbf{FP} + \mathbf{PF}' + \mathbf{GQG}' \right)$

Table 2. The implementation of the MEKF

Step	Equations
Update	(87Hz) If new magnetometer data
Rotate reference	$\hat{\mathbf{m}}_b = \mathbf{R}(\hat{\mathbf{q}})^T \mathbf{m}_n$
	$\hat{\mathbf{a}}_b = \mathbf{R}(\hat{\mathbf{q}})^T \mathbf{a}_n$
Correction	$\sigma_m = k_a(\mathbf{a}_b \times \hat{\mathbf{a}}_b) + k_m \left((\mathbf{m}_b \times \hat{\mathbf{m}}_b)^T \mathbf{a}_b \right) \mathbf{a}_b$
Update	$\delta\hat{\mathbf{q}} = \frac{1}{2}\hat{\mathbf{q}} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} - \hat{\mathbf{b}} + k_p + \sigma_m \end{bmatrix}$
	$\hat{\mathbf{q}}(+) = \hat{\mathbf{q}}(-) + \Delta t \, \delta\hat{\mathbf{q}}$
	$\hat{\mathbf{b}}(+) = \hat{\mathbf{b}}(-) + \Delta t \, (-k_I \sigma_m)$
Normalize	$\hat{\mathbf{q}}(+) = \hat{\mathbf{q}}(+)/|\hat{\mathbf{q}}(+)|$
Propagate	(400Hz) If no new magnetometer data
Propagate	$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\hat{\mathbf{q}} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} - \hat{\mathbf{b}} \end{bmatrix}$
	$\hat{\mathbf{q}}(+) = \hat{\mathbf{q}}(-) + \Delta t \, \dot{\hat{\mathbf{q}}}$

Table 3. The Mahony implementation

directly, with some surrounding logic to handle asynchronous updates, and correction of numerical drift on the quaternion. With the magnetic noise in mind, the implementation has been adapted to use the update (7) from Martin and Salaun (2010). The detailed steps of the algorithm are shown in Table 3.

A significant part of the filter implementation is the supporting libraries to handle matrix, vector and quaternion operations in 32 bit floating point precision. We have chosen to write the methods specific for each matrix dimension, eliminating the overhead accompanied with generic functions supporting arbitrary length vectors.

7. EVALUATION

The hardware implementation has been tested with the QUEST algorithm, MEKF and The Explicit Complementary Filter. And the run-time of each algoritm has been recorded. The timer `TIM4` was set up specifically to time the algorithms, with a clock resolution of 1 μs. The results are shown in Table 4. Note that this is only timing the algorithm run time, excluding the time spent fetching the measurement from the sensors.

Algorithm	Propagate [μs]	Update [μs]
MEKF	1 337	5 580
QUEST, (part of MEKF)		571
Explicit Complimentary Filter	92	376

Table 4. Execution time of the algorithms

To evaluate the accuracy and estimation performance of the sensor, it has been manually aligned and mounted together with the Microstrain 3DM-GX3, see Figure 3.

7.1 Runtime

Our QUEST implementation runs at 571 μs. Directly comparable results have been published earlier by Takahashi et al. (2009), where two processors are compared at 24MHz. Execution time for the ARM Cortex M3 at 72MHz in 32 bit precision is reported to 2444 μs. The large discrepancy in execution time may be due to the custom 3x3 matrix math library, differences in timing, including, or not, the time taken to read and decimate sensors, or communicate data.

The execution time of the Multiplicative Extended Kalman Filter is slow in comparison, and we cannot uphold a constant output rate of 400 Hz through the update.

The Mahony non-linear observer, on the other hand is more than 10 times faster, which leave runtime for auxiliary tasks such as communication etc.

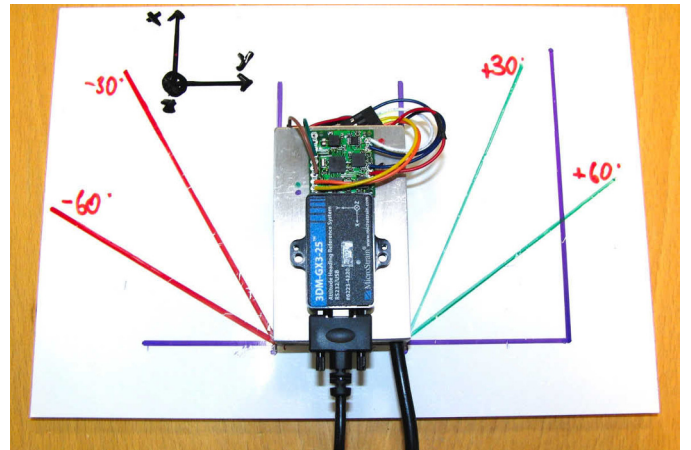### 7.2 Benchmarking towards the 3DM-GX3

Fig. 3. This setup is used to test the step-response and accuracy of the yaw estimate. The filter is benchmarked towards the known movement, as well as the reference Microstrain 3DM-GX3 sensor.

The relative high-grade MEMS sensor by Microstrain was used as a benchmark in this project. To be able to compare the MEKF and Mahony filters on the same data-sets, the AHRS was set up to output raw sensor measurements. The algorithms have then been run in Matlab on the logged data. This also facilitates the process of tuning the filters.

To test the accuracy of the algorithms, a level plate was attached by double-sided tape to a table, Figure 3. The plate was assured level by using a hand level tool, and directed north as reported by the Microstrain sensor.
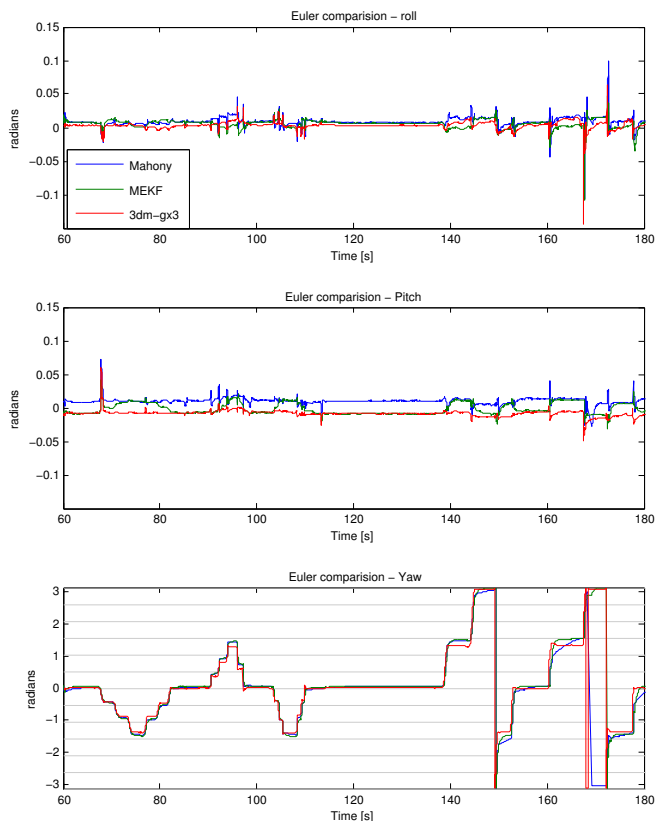
Fig. 4. Accuracy test for Yaw-angle, the lines in the Yaw plot are drawn at $n\pi/6$.

Three test sequences were performed: One for yaw accuracy and one for roll and pitch, where the sensor was moved in steps to test transient response and absolute accuracy. The last test was in free hand motion with smoother motions.

7.3 Yaw accuracy

In Figure 4 the filters output is shown. There are several interesting features in these plots. Please note that the y-axis range on Roll and Pitch is $\pm 8.6°$.

The pitch estimate shows how a small positive bias on the accelerometer y-axis directly affects the Explicit Complementary filter, with the modified update (7). The magnetometer readings are discarded from the pitch and roll updates, and the filter then solely rely on the accelerometer.

The MEKF on the other hand, uses the QUEST algorithm with near equal weights on the two measurements. As the two vector measurements conflict, QUEST alternates between trusting the magnetometer, or the accelerometer the most.

This is a desirable trait to the QUEST algorithm, and the pure Mahony update would rather have output a weighted average between the two.

Further, notice that the transients caused by the abrupt stop and go motion affect the two sensor nearly the same. This is an effect of the accelerometer measurement swinging out when rotating the sensor in a stop-and-go fashion.
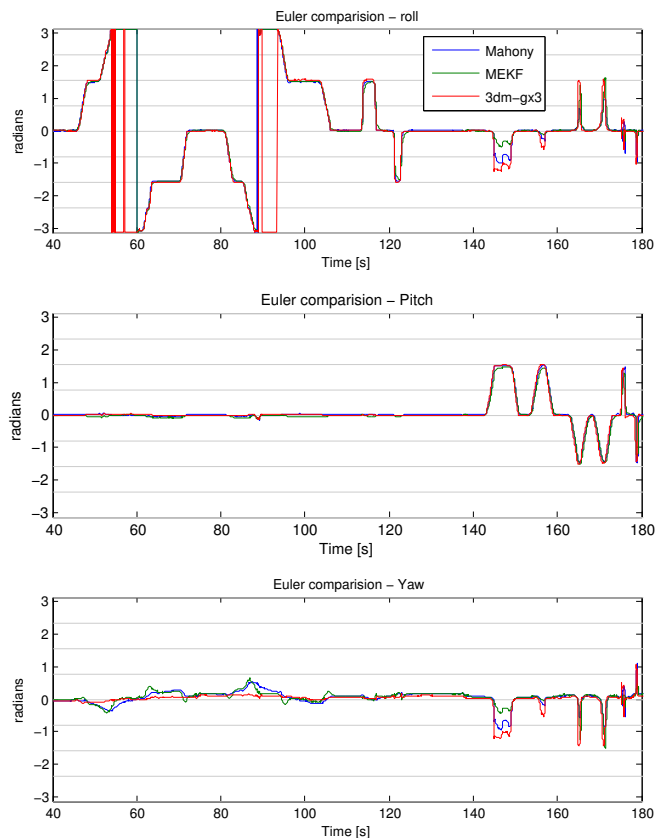

Fig. 5. Accuracy test for Roll and Pitch angle.

In the first part of the Yaw plot, the three lines follow closely, and the Microstrain slightly underestimate the rotation which was performed in three steps to $\pm\pi/2$. In the latter part of the graph, the Microstrain display better performance under fast dynamics, where the two filters on the AHRS data first underestimate the rotation, but slowly swing in to near correct levels.

7.4 Roll and pitch accuracy

The deviation in Figure 5 is mostly due to a combination of non-perfect alignment of the two sensors, and differences in how the magnetic field is measured.

The filters also show excellent accuracy in estimating the roll and pitch angles when the gravitation vector is aligned with a measurement vector. At 170s the Euler conversion experience near-gimbal-lock conditions, and the Roll and Yaw measurements does not represent the physical motion.

7.5 Hand held

The dataset shown in Figure 6 was recorded by freely holding the sensor-block, and moving it around. In contrast to the Yaw Accuracy test, the Microstrain now show an higher amplitude in yaw rotation. Otherwise the filters display very similar dynamics, where the MEKF is the most conservative one.

At approximately 52 seconds, the Microstrain Euler conversion experience a gimbal lock, and spin both the Roll and Pitch angle 360°, a mathematical artifact in the conversion to Euler angles.
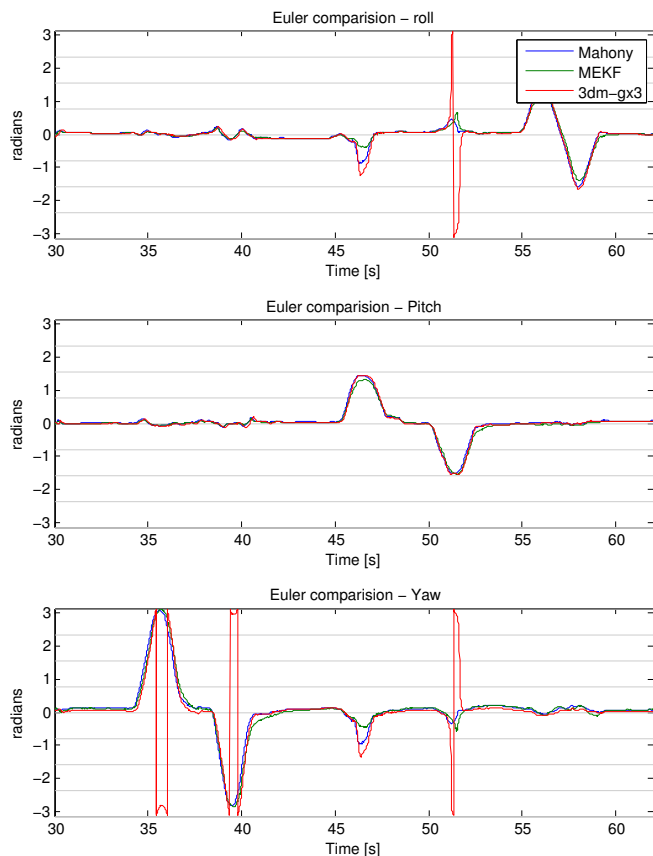
Fig. 6. The filters under controlled hand motion.

8. RESULTS

We have demonstrated the implementation of the QUEST algorithm, the much implemented Multiplicative EKF and a non-linear observer, the Explicit Complimentary Filter.

The algorithms have been implemented on the CHR-6dm, where most of the firmware has been re-engineered to better accompany the filters. The implemented algorithms have been analysed by run-time on the microprocessor. The QUEST algorithm runs at 571 μs, which is four times faster than a previously published implementation Takahashi et al. (2009), on nearly the same processor.

The Mahony non-linear estimator demonstrate its computational advantage, by running more than ten times faster than the MEKF.

By comparing the algorithms with a relative high-grade sensor, the usability and accuracy of the filters have been positively indicated.

9. CONCLUSION

The implementation, and run-time results of the MEKF and the Mahony non-linear observer illustrate clearly the relative computational cost of Kalman filters to non-linear observer designs. Whereas relatively cheap micro-controllers are fully capable of running a MEKF filter, the chosen non-linear observer run ten times faster.

The Extended Kalman filter design provides few obvious advantages, besides being the industry standard for decades. Non-linear observers may be more demanding in design, but provide attractive stability properties and the implementation in code is significantly more lightweight and transparent. This in turn leave less room for software bugs and facilitate test and verification.

For our agricultural robotic attitude estimation, the explicit complementary filter is preferred over a Kalman based design. And we focus on applying observers which also incorporate heading estimates from the forward motion measured by GPS to improve absolute accuracy.

REFERENCES

Crassidis, J., Markley, F., and Cheng, Y. (2007). Survey of nonlinear attitude estimation methods. *Journal of Guidance Control and Dynamics*, 30(1), 12.

Egeland, O. and Gravdahl, J. (2002). *Modeling and Simulation for Automatic Control.*

Grip, H., Fossen, T., Johansen, T., and Saberi, A. (2011). Attitude estimation based on time-varying reference vectors with biased gyro and vector measurements.

Hamel, T. and Mahony, R. (2006). Attitude estimation on so [3] based on direct inertial measurements. In *Robotics and Automation, 2006. ICRA 2006. Proc. 2006 IEEE Int. Conference*, 2170–2175. IEEE.

Hua, M. (2010). Attitude estimation for accelerated vehicles using gps/ins measurements. *Control Engineering Practice*, 18(7), 723–732.

Lefferts, E., Markley, F., and Shuster, M. (1982). Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 5(5), 417–429.

Mahony, R., Hamel, T., and Pflimlin, J. (2008). Nonlinear complementary filters on the special orthogonal group. *Automatic Control, IEEE Transactions on*, 53(5), 1203–1218.

Markley, F. et al. (2003). Attitude error representations for Kalman filtering. *J. of Guidance Control and Dynamics*, 26(2), 311–317.

Martin, P. and Salaun, E. (2010). Design and implementation of a low-cost observer-based attitude and heading reference system. *Control Engineering Practice*, 18(7), 712–722.

Shuster, M. (1990). Kalman filtering of spacecraft attitude and the QUEST model. *J. of the Astronautical Sciences*, 38, 377–393.

Shuster, M. (1993). The Quaternion in Kalman Filtering. *Advances in the Astronautical Sciences*, 85, 25–37.

Shuster, M. (2006). The quest for better attitudes. *Journal of the Astronautical Sciences*, 54(3/4), 657.

Shuster, M. (2009). Filter QUEST or REQUEST. *Journal of Guidance, Control, and Dynamics*, 32, 643–645.

Shuster, M. and Oh, S. (1981). Three-axis attitude determination from vector observations. *J. of Guidance and C.*, 4(1), 70–77.

Slaughter, D., Giles, D., and Downey, D. (2008). Autonomous robotic weed control systems: a review. *Computers and Electronics in Agriculture*, 61(1), 63–78.

Takahashi, S.N., de Souza, A., and Tosin, M. (2009). Execution time of QUEST algorithm in embedded processors. In *Symposium on Aerospace Eng. and Applications*.

Utstumo, T. (2011). *Attitude Estimation in Agricultural Robotics*. Master's thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology.