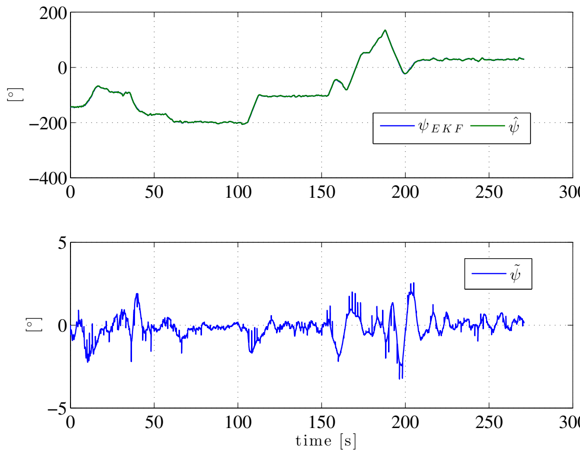

Fig. 5: The upper plots show the yaw angle estimates of the onboard EKF, ψ_{EKF}, and the nonlinear observer, $\hat{\psi}$. The lower plot shows the difference between the estimates, $\tilde{\psi} = \psi_{EKF} - \hat{\psi}$.

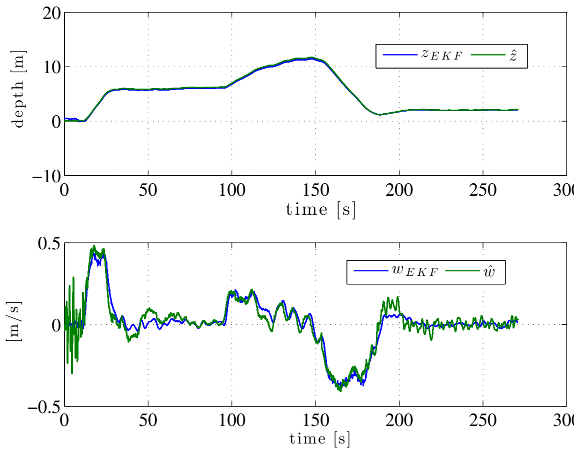

Fig. 7: Estimates of depth and heave velocity.

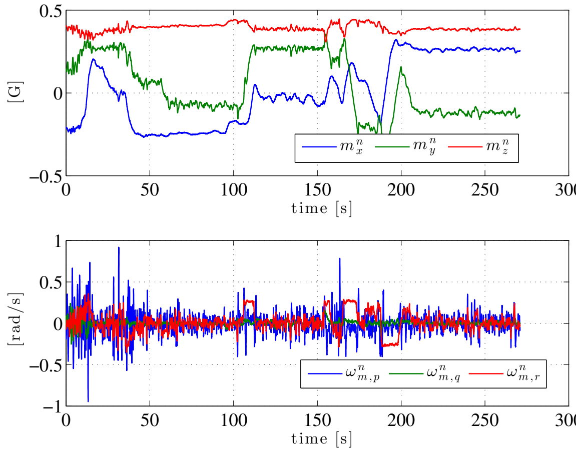

Fig. 6: The upper plots show the measured magnetic field strength, m^n. The lower plot shows the measured angular velocities, ω^n_m.

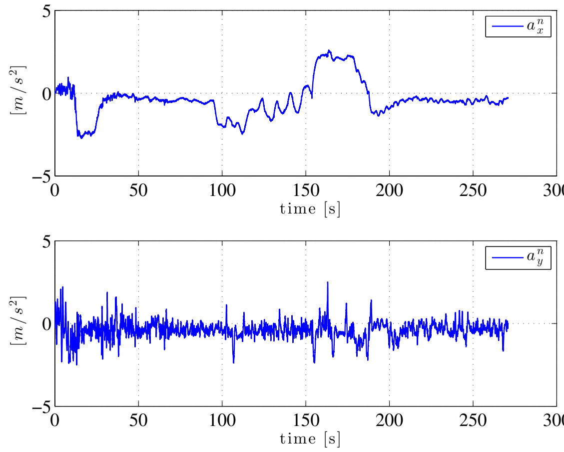

Fig. 8: The acceleration components of the horizontal plane in NED.

E. Assumptions Discussion

Figure 8 shows the Kalman filtered estimates of the acceleration in the x and y direction in the NED frame. It is evident that *Assumption 3* is not strictly valid throughout the experiment, since the horizontal accelerations deviate from zero. Even though the Seacon-1 vehicle's behavior does not strictly fulfill *Assumption 3*, the nonlinear observer still displays excellent performance in estimating attitude and heave position.

VI. Conclusion

In this paper the experimental results from testing a nonlinear observer on an AUV data set were presented. The nonlinear observer was modified by only utilizing a pressure sensor instead of GNSS position and velocity. The measured pressure was effectively treated as a measure of depth and the observer was therefore modified to only estimate the heave motion along with the attitude. The experimental testing showed excellent performance of the nonlinear observer, with performance comparable with the onboard Kalman filter. The nonlinear observer was global exponentially stable and it is much less computational demanding than the Kalman filter.

Acknowledgment

This work was funded by The Research Council of Norway through the FRINATEK program (Project no. 221666) together with the center of Autonomous Marine Operations and Systems (AMOS) at the Norwegian University of Science and Technology.

References

[1] P. Batista, C. Silvestre, and P. Oliveira. A Sensor-based Long Baseline Position and Velocity Navigation Filter for Underwater Vehicles. *Proceedings of the 8th IFAC Symposium on Nonlinear Control Systems*, pages 202–307, 2010.

[2] R. G. Brown. Integrated Navigation Systems and Kalman Filtering: A Perspective. *Navigation: Journal of the Institute of Navigation*, 19:355–362, Winter 1972-1973.

[3] J. A. Farrell. *Aided Navigation*. McGraw-Hill, 2008.

[4] T. I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, 2011.

[5] H. F. Grip, T. I. Fossen, T. A. Johansen, and A. Saberi. A Nonlinear Observer for Integration of GNSS and IMU Measurements with Gyro Bias Estimation. *American Control Conference (ACC)*, pages 221–228, 2012a.

[6] H. F. Grip, A. Saberi, and T. A. Johansen. Observers for cascaded nonlinear and linear systems. *In Proceedings of the IEEE Conference on Decision and Control*, pages 3331–3337, 2011.

[7] H. F. Grip, A. Saberi, and T. A. Johansen. Observers for Interconnected nonlinear and linear systems. *Automatica*, 48:1339–1346, 2012.

[8] M. D. Hua. Attitude Estimation for Accelerated Vehicles Using GPS/INS. *Control Engineering Practice*, 18:723–732, 2010.

[9] J. Jouffroy and J. Opderbecke. Underwater Vehicle Trajectory Estimation Using Contracting PDE-Based Observers. *In Proceedings of the 2004 American Control Conference*, 5:4108–4113, 2004.

[10] J. Kinsey and L. Whitcomb. Preliminary Field Experience with the DVLNAV Integrated Navigation System for Manned and Unmanned Submersibles. *In Proceedings of the 1st IFAC workshop on guidance and control of underwater vehicles*, pages 83–88, 2003.

[11] M. Krstic, I. Kanellakopoulos, and P. Kokotiovic. *Nonlinear And Adaptive Control Design*. John Wiley and Sons, Inc., 1995.

[12] M. Larsen. *Autonomous Navigation of Underwater Vehicles*. PhD thesis, Department of Automation, Technical University of Denmark, 2001.

[13] J. Leonard, A. Bennett, C. Smith, and H. Feder. Autonomous underwater vehicle navigation. *Technical Report Technical Memorandum*, 1, 1998.

[14] B. Vik and T. I. Fossen. A Nonlinear Observer for GPS and INS Integration. *In Proceedings of the IEEE conference on decision and control*, pages 2956–2961, 2001.

Appendix

A. Pseudo Code for Nonlinear Observer

The Matlab code for a single iteration of the depth-aided nonlinear observer is given below. Prior to entering the observer loop the variables needed for operation are initialized, including the needed conversion from pressure measurements to depth. The code does not contain storage of variables needed for plotting, but only the basic mechanics of the nonlinear observer needed for implementation.

The variable `n_depth` handles the frequency difference from measurement updates and `j_depth` is an index counter for the depth measurements. The overall observer loop iterates using `i` and `i` therefore also functions as the index variable for the IMU measurements.

The naming of variables have been attempted to be self-explanatory, e.g. z_{hat} is represented by `z_hat`, $\hat{a}^n$ by `a_hat_n`. The gains K_{zz}, K_{wz} and $K_{\zeta z}$ are represented by `K_zz`, `K_wz` and `K_zeta`, respectively and have been chosen as $K_{zz} = 0.5$, $K_{wz} = 0.5$ and $K_{\zeta z} = 0.14$. The attitude estimation gains are $k_I = 1.0$, $\sigma = 1.0$ and K_p as a diagonal matrix with $K_{p,11} = 0.5$, $K_{p,22} = 2.0$, and $K_{p,33} = 0.5$.

The function `skew_sym(A)` represents the skew-symmetric operator $\mathbb{P}_a(A)$, the linear function vex(A) is represented by `vex(A)` and the skew-symmetric matrix $S(x)$ is represented by the function `space_iso(x)`.The functions are not included in the code below, but are straightforward to implement in Matlab.

```
if n_depth == 5; % Update depth measurements.
    z = z_meas(j_depth);
    j_depth = j_depth+1;
    n_depth = 0;
end

% Corrector for heave motion observer
z_hat = z_bar + K_zz*(z-z_bar);
w_hat = w_bar + K_wz*(z-z_bar);
zeta_hat = zeta_bar + [0 0 1]'*K_zeta*(z-z_bar);
a_hat_n = B*R_hat*a_b(:,i) + B*zeta_hat;

% Normalizing the vectors for the injection term J
a_hat_n_norm = a_hat_n/norm(a_b(:,i));
m_b_norm= m_b(:,i)/norm(m_b(:,i));
a_b_norm = a_b(:,i)/norm(a_b(:,i));

A_b = [m_b_norm space_iso(m_b_norm*a_b_norm) ...
 space_iso(m_b_norm)*space_iso(m_b_norm)*a_b_norm];

A_hat_n = [m_n_norm space_iso(m_n_norm)*...
 a_hat_n_norm space_iso(m_n_norm)*...
 space_iso(m_n_norm)*a_hat_n_norm];

J_hat = A_hat_n*A_b'-R_hat*(A_b*A_b');

% Saturation of values below and above -1,1
R_hat_s = max(min(R_hat,1),-1);

% Attitude estimator from eq. (4)
R_dot_hat = R_hat*space_iso(omega_b_imu(:,i)...
 -b_hat_b_gyro)+rho*K_p*J_hat;
b_dot_hat_b_gyro = projection(b_hat_b_gyro,...
 -k_I*vex(skew_sym(R_hat_rho'*K_p*J_hat)),M_b);

% Predictor
z_bar = z_hat + dt*w_hat;
w_bar = w_hat + dt*a_hat_n(3)+dt*g;
zeta_bar = zeta_hat - dt*rho*K_p*J_hat*a_b(:,i);

% Euler integration updates of attitude estimator
R_hat = R_hat + dt*R_dot_hat;
b_hat_b_gyro = b_hat_b_gyro + dt*b_dot_hat_b_gyro;

% Update depth counter
n_depth = n_depth+1;
```