USBL Pose Estimation using Multiple Responders

Leif Christensen* and Martin Fritsche* and Jan Albiez* and Frank Kirchner*
*Robotics Innovation Center
DFKI Bremen
Germany, 28359 Bremen
Email: leif.christensen@dfki.de

Abstract— **Ultra Short Baseline Systems (USBLs) to track a submerged target normally consist of the transceiver unit mounted to a vessel and usually only one responder per underwater vehicle to track. Since the system measures vertical and horizontal angles of the returning signal in conjunction with the range, the system is only capable of providing a 3D position and not the entire pose (position and orientation) of the submerged vehicle or its dynamic state, which sometimes is needed. In this paper we propose the use of multiple USBL responders mounted onto a sub sea crawler to improve the accuracy and robustness of the tracked vehicle's position and velocity and, in the case of restricted or non-available orientation sensor data from the vehicle itself (e.g. no expensive inertial navigation system, no real-time communication with the vehicle), to make an external sub sea orientation estimation possible at all.**

In the paper, the decisions that lead to the chosen probabilistic filter to accomplish the state estimation task mentioned above and the concrete filter adaption is explained in detail with special focus on state composition, the corresponding measurement model with three responders and outlier rejection using the Mahalanobis Distance. The tracking system application consisting of the Bayesian filter and the data acquisition interfaces and the actual data flow is presented as well as the design of the related simulation software. Finally, the performance of the filter with varying parameters and its robustness against outliers and signal loss is shown.

I. Introduction

With the world-wide rising demand for maritime resources like offshore oil and gas, more and more strategies are employed to use robotic systems for exploration, setup and maintenance of sub sea production facilities. These robotic systems are either remotely operated (ROV) or full autonomous (AUV). The ROVs are primary used for intervention tasks and the AUV domain is bathymetry. In both cases the knowledge of the current position of the robot is of uttermost importance.

Compared to most land-based robotic systems, the localisation of an underwater robotic vehicle is a major problem due to the lack of proprioceptive sensors with adequate accuracy (e.g the accuracy of laser scanner sensors commonly in use in terrestrial mobile robotics). Furthermore, whereas positioning on the surface has found its reference technology in Global Navigation Satellite Systems (GNSS) like GPS-Navstar, supplying inexpensive solutions for hydrograhpers, marine surveyors and maritime work in general, solutions to the problem of sub sea positioning are still complicated and expensive in comparison, mostly due to the fact that higher frequency radio signals don't penetrate water [1]. Therefore, usually several different sensor systems are mounted on an underwater vehicle and sensor fusion algorithms are then used to obtain a more robust estimation of the vehicles position [2]. Unfortunately, there is no single proprioceptive sensor that allows for accurate and robust underwater localisation, although there are efforts to combine all needed different sensors in a single unit like the Global Acoustic Positioning System (GAPS) [1] [3].

Current AUVs use a variety of different sensors to estimate their position. The euclidean orientation with respect to the magnetic and gravitational field of earth is measured by an *Inertial Measurement Unit (IMU)* (e.g. integrated with USBL data in [4]) combined with an electronic compass, a *Doppler Velocity Log (DVL)* [5] is used to measure the speed over ground, depth sensor and echo sounder give information about the position in the water column. For an overview see [6]. Most of the sensors named above are not only quite expensive, but also add significant amounts to power consumption and construction space, so the demand arises to reduce the amount of sensors needed for underwater localisation.

Fig. 1. Scaled multipurpose sub sea working unit of the project ISUP (founded by the German Ministry for Economics and Technology (BMWi), grant No. 03SX229A), courtesy of Aker WIRTH GmbH [7]

In a case where the robot operates in a predefined area, several additional possibilities for acoustic localisation exist. *Long Base Line (LBL)* localisation relies on a network of acoustic beacons with known location, emitting an acoustic signal in a predefined interval. These signals are captured by an array of hydrophones on the robot. The time-delay and direction is used to calculate the baselines for each transponder, the robot is at the intersection of all baselines. LBL systems can be very accurate, but the transponder network has to be setup and calibrated, a timely and therefore costly effort [8]. *Ultra Short Baseline Systems (USBLs)* to track a submerged target normally consist of the transceiver unit mounted to a vessel and usually only one responder per underwater vehicle to track. Since the system measures vertical and horizontal angles of the returning signal in conjunction with the range, the system is only capable of providing a 3D position and not the entire pose (position and orientation) of the submerged vehicle or its dynamic state.

In this paper we propose the use of multiple USBL responders mounted onto a multipurpose sub sea crawler to improve the accuracy and robustness of the tracked vehicles' position and velocity and, in the case of restricted or non-available orientation sensor data from the vehicle itself (e.g. no expensive inertial navigation system, no real-time communication with the vehicle), to make an external sub sea orientation estimation possible at all.

First we will show the *System configuration* (II) and then discuss the *Filter Design* (III) and outlier rejection (III-D) in detail. In the section *Implementation, Tests and Results* (IV) we will show the concrete implementation and the filter performance with different parameters before we state our *Conclusion* (V).

II. System configuration

The impulse to create a simple, yet robust underwater tracking system emerged within the project ISUP (Integrated Systems for Underwater Production of Hydrocarbons [9]).

Fig. 2. One of the three responder units mounted onto the DWU, courtesy of Aker WIRTH GmbH

Goal of the ISUP project is the development of innovative and future-proof components and subsystems for construction and operation of underwater production sites, with one component being a multi functional underwater working unit based on a crawler design, that is able to assist teleoperated and semi-autonomous at the sea bottom in construction, operation, extension and decommissioning [7]. Aside from a big manipulator arm to grip swinging loads that have to be installed at the sea bottom, the working unit consists, among other components, of a bulldozer plate and a multi purpose coupling for different tools like a vibrational drill or different kind of mills. For testing purposes, a smaller version of the targeted system (which will weight about 32 tons), the *demonstration working unit* (DWU) was build, depicted in fig. 1.

TABLE I

Tritech MicronNav specification [10]

System	
Frequency Band	20 - 28 kHz
Tracking Range	typical 500 m horiz., 150 vert.
Position Update Rate	0.5 - 10.0 s
Range Accuracy	+/- 0.2 m typical
Bearing Accuracy	+/- 3°
Targets Tracked	1 - 15
Surface Station Power	110 - 220 V AC or 9 - 30 V DC
MicronNav underwater unit	
Beam width	omni-directional
Power Consumption	3.5 W (280 mW standby)
Transmitter Source Level	169 dB re 1 uPa @ 1M
Depth Rating	750 m

For navigational purposes at first tests in the harbour of Hamburg, the DWU was equipped with a *MicronNav* USBL positioning system from *Tritech International Ltd*.

Under the assumption, that every acoustic navigational system would suffer from reflection or even extinction problems due to the complex structured steel surface of the vehicle, three underwater responder units were distributed on the DWU.

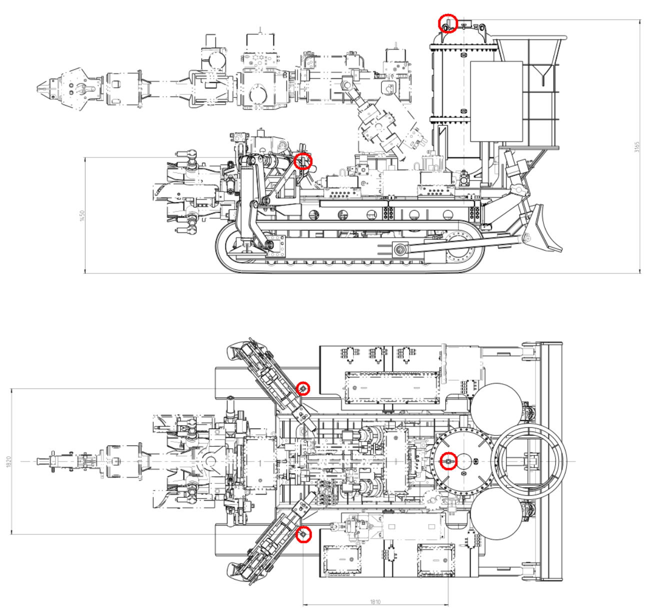


Fig. 3. Mounting positions of the three USBL responders

In addition to add redundancy for failure safety purposes in positioning, the use of three responders should allow not only for a position- but also an orientation estimation. Compared with systems that make use of multiple acoustic units like Short Baseline Systems (SBL) or Long Baseline Systems (LBL), the baseline between the responders onto the DWU with approx. 1.8 m is still very short (see fig. 3), but it can be extended in the larger working unit in the future. For the demonstrator we expected only rough orientation estimates given the sensor noise (fig. 4), so an additional tilt-compensated compass module was installed on the DWU.

III. Filter Design

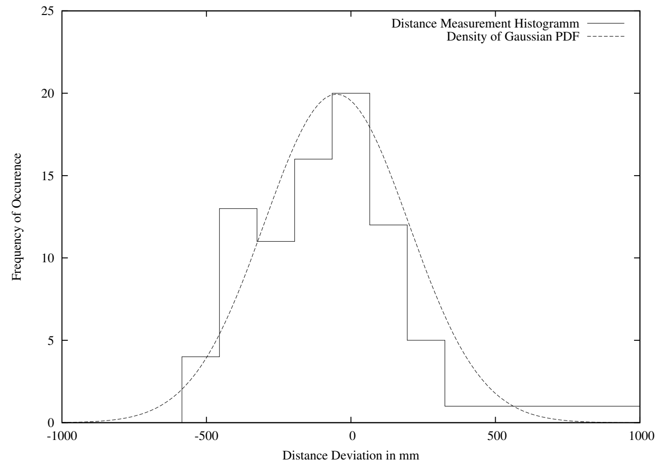
All sensors are error-prone and this is inherent to all classes of sensors, therefore we have to infer the (not observable) system state of our underwater robot from noisy and/or partial sensor data using probabilistic methods. In general, *Bayes filter* algorithms calculate the belief distribution of a state x at time t using two basic steps: *prediction* and *measurement update*. The following pseudo algorithm taken from [11] shows these steps:

Algorithm 1 Bayes filter algorithm

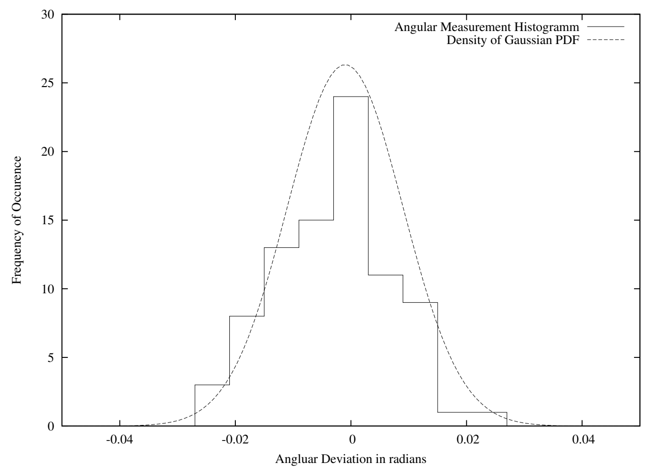
1: **for all** x_t **do**
2: $\quad \overline{bel}(x_t) = \int p(x_t|u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1}$
3: $\quad bel(x_t) = \lambda p(z_t|x_t) \overline{bel}(x_t)$
4: **end for**
5: **return** $bel(x_t)$

In the *prediction* step in line 2, the filter uses the old state x_{t-1} in conjunction with a control u_t to estimate the new state x_t. That the filter needs only the last state and not the whole chain of previous states is common to all Bayes filter types, and is called the *Markov Assumption*. The *measurement update* step in line 3 takes the prediction $\overline{bel}(x_t)$ and multiplies it with the probability p that the measurement z_t is observed, given the state x_t and normalized using λ. These two basic steps, *prediction* and *measurement update* are common to all filtering algorithms based on the Bayes Filter. The main difference is, how they represent the conditional probability distribution of the current belief. The group of *Gaussian filters* represents the current belief as a multivariate normal distribution represented only by its respective mean μ (the "center" of the normal distribution) and covariance $\sum$ (the "width" of the Gaussian bell), whereas a non-parametric filter like the *particle filter* represents the belief distribution by a set of random samples (the particles) drawn from it. Inherently, the representation of the belief only with mean and covariance in the case of Gaussian filters restrict these group of filters to unimodal normal distributions. In comparison, the particle filter is not restricted in the way how the belief distribution is shaped, but the quality of the representation is strongly related to the amount of samples drawn from the distribution. In case of complex multivariate distributions (e.g. a high-dimensional state vector), the particle filter can get highly computational expensive [11].

Measurement data logged in first tests with our USBL system showed, that all the noise in our sensor can be classified as Gaussian noise (see fig. 4) and that the noise distribution is comparable to similar findings in [12].



(a) Distance Noise



(b) Angular Noise

Fig. 4. USBL noise distribution (static position data set)

According to this, the actual belief in the state of the tracked vehicle is a multivariate distribution that can be represented with only their respective mean and covariance. Given these prerequisites, a Gaussian filter like the Kalman Filter or a variation of it can play out its strength concerning computational costs and space compared to non-parametric filters like the particle filter, so we decided to use a Gaussian filter. The still open question, what kind of Gaussian filter to choose, is treated in III-B.

A. State composition

Everything that the filter should estimate later has to be put into the state. This is independent of the choice of the concrete filtering algorithm, whether we use a Kalman Filter (KF), an Extended Kalman Filter (EKF), an Unscented Kalman Filter (UKF) or whatever flavor of Gaussian or, more general, Bayesian filtering (for example a particle filter) is allowed or needed by the system.

A full 6 degree of freedom state x_t consists of the vehicles location (Cartesian coordinates x, y, z) and orientation (roll θ, pitch ϕ, yaw ψ) relative to a global coordinate frame, the *static state* η, as well as the respective translational velocities (surge u, sway v and heave w) and the respective rotational velocities (see fig. 5), the *dynamic state* ν.

$$\eta = \begin{bmatrix} x & y & z & \theta & \phi & \psi \end{bmatrix}^T \tag{1}$$

$$\nu = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T \tag{2}$$

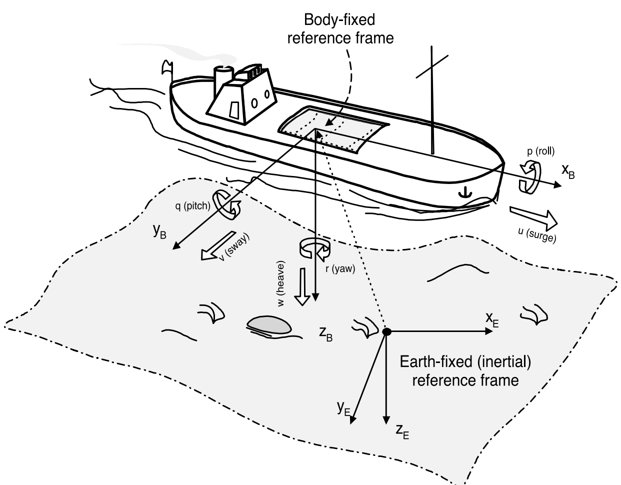


Fig. 5. Maritime coordinate system [13]

Since our underwater vehicle is a sub sea crawler, which is confined to a planar harbour environment, our static state η_{cwl} can be reduced to x, y and yaw ψ:

$$\eta_{cwl} = \begin{bmatrix} x & y & \psi \end{bmatrix}^T \tag{3}$$

Our crawler can adopt every possible constellation of the variables in the static state, but not every transition from one constellation to another in time is possible, x, y and ψ are not totally independent on a small time scale. These holonomic constrictions lead to a non-holonomic system and, since our crawler sits solid on the sea floor and is only marginally affected by external forces like the current, it suffices to take the forward velocity u into our dynamic state ν_{cwl} which leads to our state x_t

$$x_t = \begin{bmatrix} \eta_{cwl}^T & \nu_{cwl}^T \end{bmatrix}^T = \begin{bmatrix} x & y & \psi & u \end{bmatrix}^T \tag{4}$$

The USBL System in use puts out raw position data in Cartesian coordinates, but since the system physically measures vertical and horizontal angles (bearing) and the range to the responder (see fig. 6), the error and noise of the sensor are mixed in the x- and y-coordinate, whereas they can be appropriately allocated in the respective polar coordinates after conversion. This is reflected in our state x at time t which consists of the angle to the vehicle α and the range d, instead of Cartesian coordinates x and y.

$$x_t = \begin{bmatrix} \alpha & d & \psi & u \end{bmatrix}^T \tag{5}$$

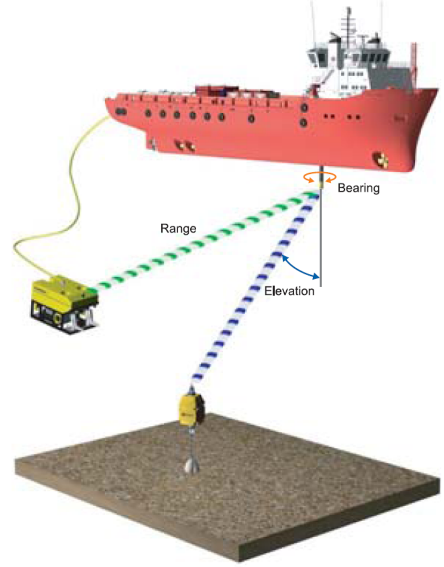


Fig. 6. Typical USBL setup [14]

B. Prediction Step

One of the best studied implementation of a Gaussian filter, which is the class we decided for (see III), is the *Kalman Filter* [11]. Using a Kalman Filter, the new state x_t can be computed from the control u_t and the last state x_{t-1} in the following way:

$$x_t = Ax_{t-1} + Bu_t + \epsilon_t \tag{6}$$

Since we just want to observe the vehicle and don't utilize the control commands, we can leave out u_t:

$$x_t = Ax_{t-1} + \epsilon_t \tag{7}$$

The consequences of a control command will, however, not be ignored, but will result in a different state and therefore be incorporated in the next measurement. Our state contains everything we need to predict the new state: the heading ψ and velocity u form a vector that just has to be added to the vector composed of α and d. If we would convert our state back to Cartesian coordinates and express both vectors with their respective x- and y- values

$$x_t = \begin{bmatrix} \alpha_t & d_t & \psi_t & u_t \end{bmatrix}^T \rightarrow x'_t = \begin{bmatrix} x & y & \Delta x & \Delta y \end{bmatrix}^T \tag{8}$$

, we could construct Matrix A, which just adds componentwise the velocity vector values to the position, which would lead to the following equation:

$$\begin{pmatrix} x' \\ y' \\ \Delta x' \\ \Delta y' \end{pmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ \Delta x \\ \Delta y \end{pmatrix} + \epsilon_t \tag{9}$$

But as stated above, we composed the state with direction, distance, heading and speed for good reason, which is that our sensor has different noise in direction and distance which would be mixed in the respective x- and y- values. So since we want to stick with our polar-coordinate based state, the prediction of the next state x_t given the one before looks like this:

$$\begin{pmatrix} \alpha' \\ d' \\ \psi' \\ u' \end{pmatrix} = \begin{pmatrix} atan2(d\sin\alpha + u\sin\psi, d\cos\alpha + u\cos\psi) \\ \sqrt{(d\cos\alpha + u\cos\psi)^2 + (d\sin\alpha + u\sin\psi)^2} \\ \psi \\ u \end{pmatrix} \quad (10)$$

This non-linear system update can no longer be represented with the Transition-Matrix A from (6), so we have to use an Extended Kalman Filter (EKF) instead of the Kalman Filter, since the EKF can deal up to a certain point with non-linearities. Instead of using a linear function with matrices A and B

$$x_t = Ax_{t-1} + Bu_t + \epsilon_t \quad (11)$$

it suffices for the EFK to use a differentiable function f:

$$x_t = f(x_{t-1}, u_t) + \epsilon_t \quad (12)$$

Unfortunately, we can't apply f directly to the covariance any more, so we have to compute the matrix of first partial derivatives (the Jacobian) for our EKF:

$$J(f(x_{t-1})) = \begin{bmatrix} \frac{\partial\alpha'}{\partial\alpha} & \frac{\partial\alpha'}{\partial d} & \frac{\partial\alpha'}{\partial\psi} & \frac{\partial\alpha'}{\partial u} \\ \frac{\partial d'}{\partial\alpha} & \frac{\partial d'}{\partial d} & \frac{\partial d'}{\partial\psi} & \frac{\partial d'}{\partial u} \\ \frac{\partial\psi'}{\partial\alpha} & \frac{\partial\psi'}{\partial d} & \frac{\partial\psi'}{\partial\psi} & \frac{\partial\psi'}{\partial u} \\ \frac{\partial u'}{\partial\alpha} & \frac{\partial u'}{\partial d} & \frac{\partial u'}{\partial\psi} & \frac{\partial u'}{\partial u} \end{bmatrix} \quad (13)$$

Exemplaric for the partial derivative $\frac{\partial\alpha'}{\partial\alpha}$:

$$\begin{aligned} \frac{\partial\alpha'}{\partial\alpha} &= \frac{d\sin\alpha(u\sin\psi + d\sin\alpha)}{(u\sin\psi + d\sin\alpha)^2 + (u\cos\psi + d\cos\alpha)^2} \\ &+ \frac{d\cos\alpha(u\cos\psi + d\cos\alpha)}{(u\sin\psi + d\sin\alpha)^2 + (u\cos\psi + d\cos\alpha)^2} \end{aligned}$$

The construction of the Jacobian J_f has to be done only once, later it is just evaluated with the current predicted state, which results in an linearisation of the non-linear function f around the current estimate.

C. Update Step

According to line 3 of the Bayes Filter algorithm (Algorithm 1), in the update step we have to compute the probability $p(z_t|x_t)$ that a measurement z_t occurs, given the state x_t. For a basic Kalman Filter, we would have to find a Matrix C_t that computes z_t from x_t and add the measurement noise δ_t:

$$z_t = C_t x_t + \delta_t \quad (14)$$

In the configuration with one USBL transponder mounted to the vehicle center, this would be easy: we would measure the position of the vehicle using the USBL system as polar coordinates with direction α and distance d, the heading ψ from the tilt-compensated compass and the speed u from the crawler's odometry directly or by computing it from the last two known positions, so that

$$z_t = \begin{bmatrix} \alpha_z & d_z & \psi_z & u_z \end{bmatrix}^T \quad (15)$$

which is exactly what we have as components in the state, so the Matrix C_t would be the identity matrix:

$$\begin{pmatrix} \alpha_z \\ d_z \\ \psi_z \\ u_z \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \alpha_t \\ d_t \\ \psi_t \\ u_t \end{pmatrix} + \delta_t \quad (16)$$

However, the extension of the tracking system from one to, in our case, three responders and the inherent conversion from Cartesian (sensor) to polar coordinates (state) leads to a measurement model that is no longer linear, so by utilizing an Extended Kalman Filter, instead of (14) with Matrix C, we have a non-linear function h:

$$z_t = h(x_t) + \delta_t \quad (17)$$

with $z_t = \begin{bmatrix} x_1 & y_1 & x_2 & y_2 & x_3 & y_3 & \psi & u \end{bmatrix}^T$ and x_n, y_n the Cartesian coordinates of transponder n. z_t can be computed using the current direction α, distance d, heading ψ and known center offsets with Δ_{dn} denoting the distance offset of transponder n and $\Delta_{\angle n}$ its angular offset:

$$h(x_t) = h \begin{pmatrix} x_1 \\ y_1 \\ \vdots \\ x_3 \\ y_3 \\ \psi \\ u \end{pmatrix} = \begin{pmatrix} d\cos\alpha + \Delta_{d1}\cos(\psi + \Delta_{\angle 1}) \\ d\sin\alpha + \Delta_{d1}\sin(\psi + \Delta_{\angle 1}) \\ \vdots \\ d\cos\alpha + \Delta_{d3}\cos(\psi + \Delta_{\angle 3}) \\ d\sin\alpha + \Delta_{d3}\sin(\psi + \Delta_{\angle 3}) \\ \psi \\ u \end{pmatrix} \quad (18)$$

For h we again have to construct our Jacobian Matrix J_h of the first partial derivatives, to be able to linearise g at the point of the current estimate, with J_h having the size of $z_t\times$ size of x_t, in our case 8×4:

$$J_h = \begin{bmatrix} -d\sin\alpha & \cos\alpha & -\Delta_{d1}\sin(\Delta_{\angle 1} + \psi) & 0 \\ d\cos\alpha & \sin\alpha & \Delta_{d1}\cos(\Delta_{\angle 1} + \psi) & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -d\sin\alpha & \cos\alpha & -\Delta_{d3}\sin(\Delta_{\angle 3} + \psi) & 0 \\ d\cos\alpha & \sin\alpha & \Delta_{d3}\cos(\Delta_{\angle 3} + \psi) & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

D. Outlier Rejection

Gaussian filters represent the confidence in the measurements of the system at time t in the covariance matrix $\sum_t$. Usually, this reflects the noise inherent to the sensors in the situation at hand. Nonetheless from time to time, there will be measurements, that are completely out of the usual range, with "usual" being represented by $\sum_t$. These outliers have to be dealt with, because otherwise the typical Gaussian filter will react very sensitive to these misleading measurements.

To identify these outliers, we have to use a distance measure to determine the similarity between the actual measurement z_t and the current state x_t. Since the belief in our current state at time t is modelled as a multivariate distribution represented only by the respective mean μ_t and covariance $\sum_t$, we have to use a statistical distance measure, that can handle such multivariate distributions and, in contrast to an euclidean distance, takes into account the correlation of the data set. The *Mahalanobis Distance* [15] gives us just that. It is defined as

$$d_M(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})} \tag{20}$$

with $\vec{x}, \vec{y}$ being random vectors of the same distribution and S^{-1} the inverse covariance matrix. Given a measurement z_t and a distribution of the current belief in the vehicle's state represented with the mean vector μ_t and covariance $\sum_t$, this leads in our case to

$$d_M(z_t) = \sqrt{(z_t - \mu_t) \sum\nolimits^{-1} (z_t - \mu_t)} \tag{21}$$

In [16], a fixed threshold for time- and spatial-domain validation in an EKF design is used, but it is also stated that there is a great potential for the applying a time-varying threshold to enhance performance and robustness of an EKF. Since the Mahalanobis distance always depends on the actual covariance of our state, the profit of a time-varying threshold is inherent to our outlier rejection.

In the configuration with only one USBL transponder, our measurement vector z_t had the same dimension as μ_t, so we could use equation (21) directly. In the case of using three transponders, our measurement vector z_t has eight components instead of four, so we calculated the Mahalanobis distance for each transponder by reducing z_t accordingly. Since our state (5) represents the vehicle's position at the center of the crawler (in polar coordinates α, d), we re-project the Cartesian coordinates x_n, y_n of the transponder n onto the virtual vehicle center x_c, y_c (where the center of the vehicle would be, given the current measurement) using the known angular- $(\Delta_\angle)$ and distance-offset (Δ_d) of this transponder together with the current heading θ and do a Cartesian to polar coordinate conversion:

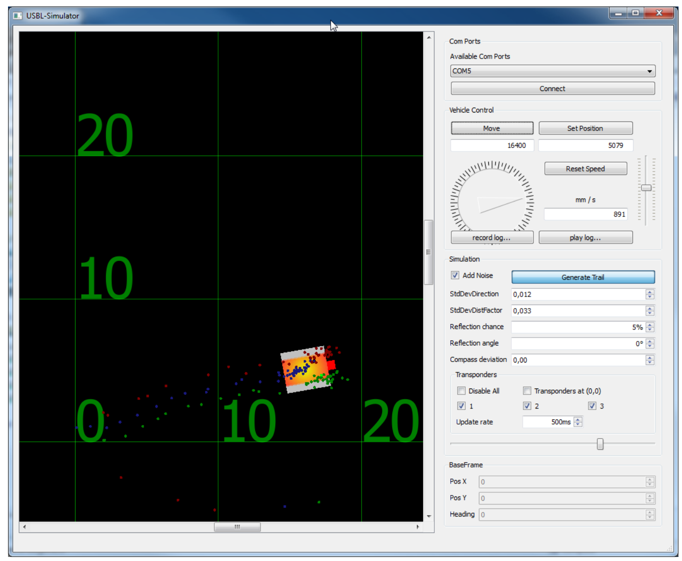
$$\begin{aligned}
x_c &= x_n - \Delta_d \times \cos(\theta + \Delta_\angle) && (22)\\
y_c &= y_n - \Delta_d \times \sin(\theta + \Delta_\angle) && (23)\\
\alpha &= \arctan \frac{y_c}{x_c} && (24)\\
d &= \sqrt{x_c^2 + y_c^2} && (25)
\end{aligned}$$

α and d can now be evaluated against the current state using (21). Please note, that the conversion is exemplary for the interval $[-\pi, \pi]$ with $x > 0$, in our implementation we used an $atan2(y, x)$ function, but this function is always implementation- / language-specific.

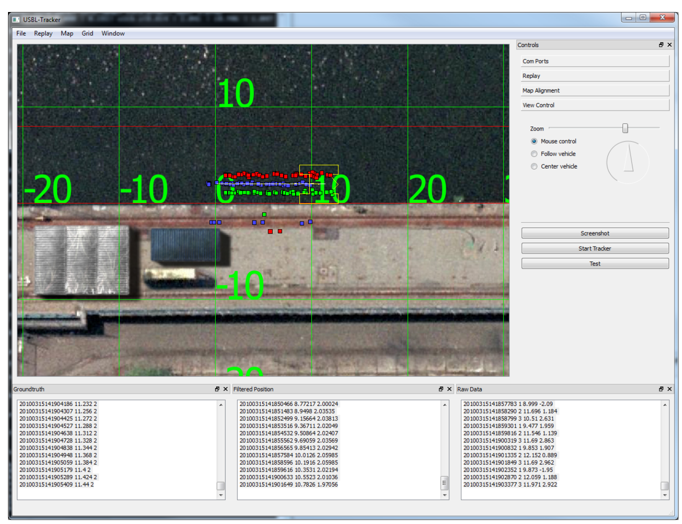
Using the Mahalanobis Distance and reducing our measurement vector z_t as shown above, we can distinguish the quality of a measurement for each transponder in a measurement set and deal with it individually, for example get rid of sound reflections on a steel sheet pile wall in the harbour.

IV. Implementation, Tests and Results

The tracking system application consisting of the Extended Kalman Filter (EKF), the data acquisition interfaces and GUI were implemented in *C++ / Qt* utilizing the Bayesian filtering library *BFL* [17] with *boost* as underlying external matrix and random number generator library. In addition, a vehicle simulation application to generate test data for the layout process of the filter with comparable additional sensor noise was also implemented.



(a) Simulator



(b) Tracking Software

Fig. 7. Developed Software for Simulation and Tracking

A. Dataflow

The functions of the DWU are controlled by a programmable logic controller (PLC) that gets it's instructions via the industry standard OPC (OLE for Process Control). Because all sensor information from the DWU, except for camera images, is also transmitted via OPC we needed an OPC connection for our tracking software. This is done by the OPC Connector that converts OPC data item updates to our in house developed communication framework ADRF and vice versa. Using this connector, our tracking software has access to the DWU's compass and track speed information. The data generated by the USBL is transmitted over a RS232 connection directly to the tracker. Instead of the real setup with DWU and USBL a simulation may be used for testing and evaluation purposes. This simulator connects to the tracker via the OPC connector and a loopback RS232 device completely simulating the existance of the real devices.

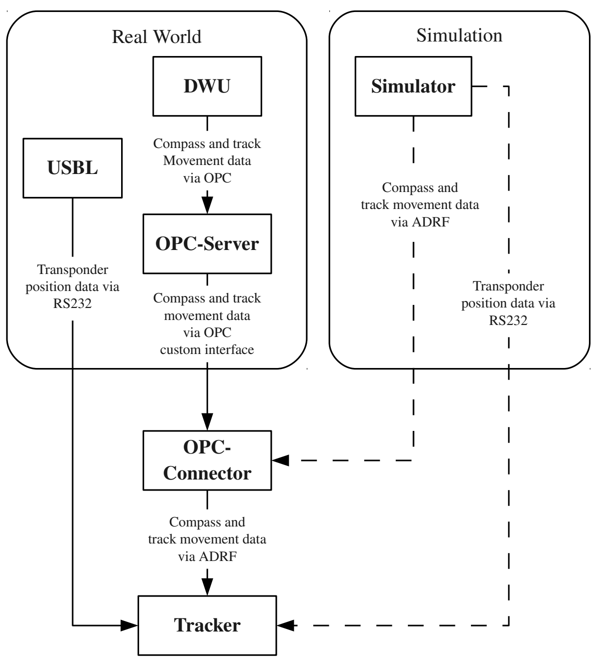


Fig. 8. Sensor dataflow

B. Vehicle Simulator

The purpose of the vehicle simulator is to provide the tracking software with all information it would get when connected to the real USBL and the DWU control system. The simulator allows the user to control the movement of a virtual vehicle with three USBL transponders according to the setup on the DWU. Everytime a virtual USBL ping is triggered, the simulator calculates the 2D position of one virtual USBL transponder according to vehicle position and heading and adds random noise dependent on the distance to the virtual USBL transducer and noise settings. The simulation of mirrored measurements caused by sound reflections is also possible. The calculated position of the transponder is tranformed into the data format used by the real device and written to a serial port. Information that would be provided from the DWU control system is sent

TABLE II

FILTER SETUP

System Model Covariance	
$\sigma\alpha$	0.08725 rad $\sim 5^\circ$
σd	$0.005m$
$\sigma\psi$	0.1745 rad $\sim 10^\circ$
σu	$0.01m/s$
Measurement Model Covariance	
$\sigma\alpha_1 \cdots \alpha_3$	0.00998 rad $\sim 0.57^\circ$
$\sigma d_1 \cdots d_3$	$315mm$
$\sigma\psi$	0.01745 rad $\sim 1^\circ$
σu	$0.1m/s$ (odom.), $0.2m/s$ (comp.)
Priori Covariance	
$\sigma\alpha_{pr}$	0.012 rad
σd_{pr}	$0.5m$
$\sigma\psi_{pr}$	0.01 rad
σu_{pr}	$0.1m$

to the tracking software via the OPC connector. To simulate non responding transponders, the user can disable single or all transponders via checkboxes. Furthermore it is possible to record and play back the movement of the simulated vehicle to compare different filter and noise settings.

Using the configuration shown in table II, the filter showed the results depicted in fig. 9 and fig. 10 for the static case and a straight run (default noise, no reflections). The μ of the *priori* distribution was set to the respective starting point. Fig. 11 shows a more complex run with stronger measurement noise ($2 \times \sigma$) and temporarily failure of individual transponders. In that case the standard Kalman Filter with only one transponder would depends on dead reckoning until the absent transponder signals could be received again.

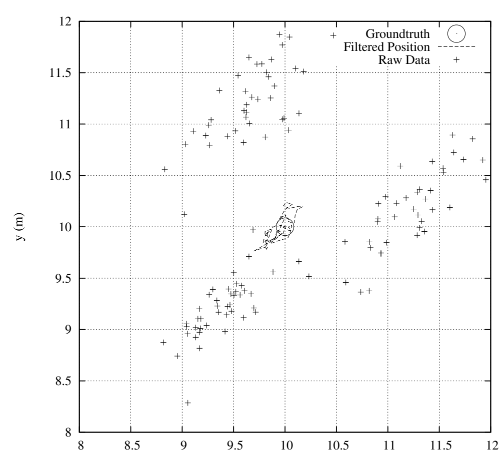


Fig. 9. Filter Performance Static

The performance of our outlier rejection using the Mahalanobis Distance is depicted in fig. 12 and 13. Here a 5 % chance of a reflection along the x axis was added to resemble the reflection noise encountered at tests in the harbour of

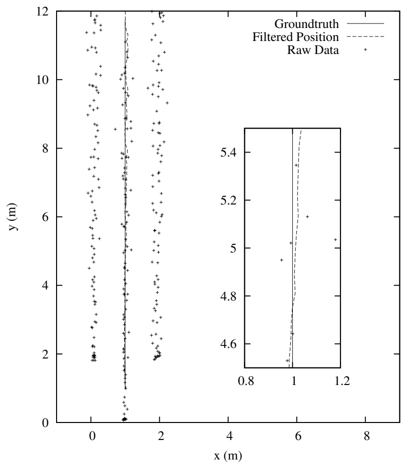


Fig. 10. Filter Performance Straigth Run

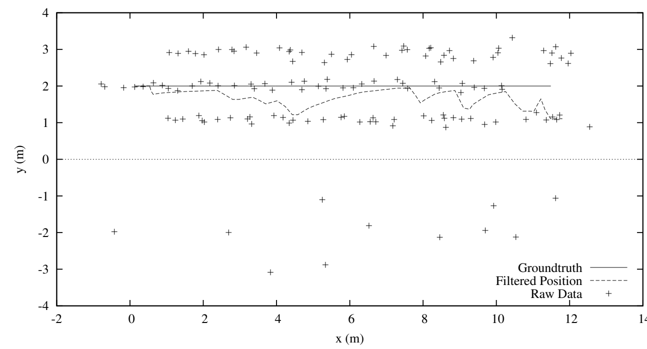


Fig. 12. Filter Performance without Mahalanobis Distance Rejection

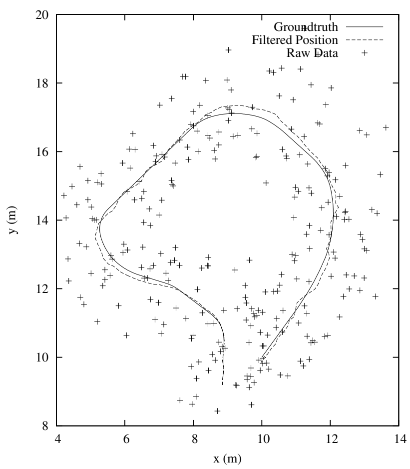


Fig. 11. Filter Performance Random Run

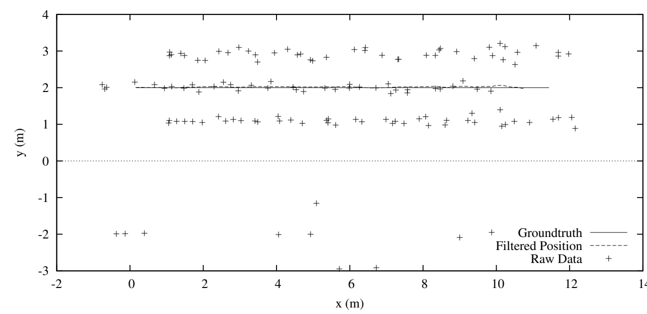


Fig. 13. Filter Performance with Mahalanobis Distance Rejection

Hamburg. Fig. 12 shows the strong filter reaction to those kind of noise and the slow convergence to the true position afterwards, since the filter increases the σ significantly before settling again. Fig. 13 depicts the same run with applied outlier rejection based on the Mahalanobis Distance.

V. Conclusion

In this paper we proposed the use of multiple USBL responders mounted onto a multi-purpose sub sea crawler to improve the accuracy and robustness of the tracked vehicles' position and velocity and, in the case of restricted or non-available orientation sensor data from the vehicle itself (e.g. no expensive inertial navigation system, no real-time communication with the vehicle), to make an external sub sea orientation estimation possible in the first place. Future work would include the postulation of a more detailed system model fitted to the 1:1 scaled sub sea crawler as well as extensive on-site testing. Nonetheless, the designed multi-responder USBL tracking system allows to externally estimate orientation and dynamic state of the tracked vehicle, while in addition enhancing the accuracy and robustness of the position estimation.

Acknowledgment

The authors would like to thank all colleagues at the DFKI RIC Bremen for their support and feedback. The work presented here was done as a subcontract to Aker WIRTH GmbH within the ISUP project, funded by the German Ministry for Economics and Technology (BMWi), grant No. 03SX229A. The authors would like to thank Torsten Kleinen, Greg Kessler and Herman-Josef von Wirth at Aker WIRTH GmbH.

References

[1] F. Napolitano, F. Cretollier, and H. Pelletier, "GAPS, combined USBL + INS + GPS tracking system for fast deployable high accuracy multiple target positioning," in *Oceans 2005 - Europe*, vol. 2, 2005, pp. 1415 – 1420 Vol. 2.

[2] G. Antonelli, *Underwater Robots: Motion and Force Control of Vehicle-Manipulator Systems*. Berlin: Springer, 2006.

[3] M. Audric, "GAPS, a new concept for USBL [Global Acoustic Positioning System for Ultra Short Base Line positioning]," in *OCEANS '04. MTTS/IEEE TECHNO-OCEAN '04*, vol. 2, 2004, pp. 786 – 788 Vol.2.

[4] M. Morgado, P. Oliveira, C. Silvestre, and J. F. Vasconcelos, "USBL/INS Tightly-Coupled Integration Technique for Underwater Vehicles," in *Information Fusion, 2006 9th International Conference on*, 2006, pp. 1–8.

[5] P. Rigby, O. Pizarro, and S. B. Williams, "Towards Geo-Referenced AUV Navigation Through Fusion of USBL and DVL Measurements," in *OCEANS 2006*, 2006, pp. 1–6.

[6] J. J. Leonard, A. A. Bennett, C. M. Smith, and H. J. S. Feder, "Autonomous Underwater Vehicle Navigation," Cambridge, pp. 1–17, 1998.

[7] T. Kleinen and D. Koenig, "Aufbruch in die Tiefsee," *Schiff & Hafen*, no. 1, pp. 74–77, 2009.

[8] E. Olson, J. J. Leonard, and S. Teller, "Robust Range-Only Beacon Localization," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 4, pp. 949–958, 2006. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4089076

[9] W. Schmitz, "Gewinnung von Erdöl & Erdgas aus dem Meeresboden," *Schiff & Hafen*, no. 9, p. 274, 2008.

[10] Tritech, "Micron-nav datasheet," http://www.tritech.co.uk/products/datasheets/micron-nav.pdf, Aberdeen, 2010.

[11] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[12] S. M. Smith and D. Kronen, "Experimental Results of an Inexpensive Short Baseline Acoustic Positioning System for AUV Navigation," in *Proc. of the IEEE/MTS OCEANS Conference and Exhibition*, Halifax, Nova Scotia, Canada, 1997, pp. 714–720.

[13] F. El-Hawary, *The Ocean Engineering Handbook*, F. El-Hawary, Ed. CRC Press, 2001.

[14] R. Balloch, D. Brown, and A. Covey, "Optimising USBL," pp. 8–13, 2007.

[15] P. C. Mahalanobis, "On the generalised distance in statistics," in *Proceedings National Institute of Science, India*, vol. 2, no. 1, April 1936, pp. 49–55. [Online]. Available: http://ir.isical.ac.in/dspace/handle/1/1268

[16] A. Alcocer, P. Oliveira, and A. Pascoal, "Study and implementation of an EKF GIB-based underwater positioning system," *Control Engineering Practice*, vol. 15, no. 6, pp. 689–701, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/B6V2H-4K7FJ9H-1/2/dd2445de9d3b7fae673e71ba2690ae75

[17] K. Gadeyne, "{BFL}: {B}ayesian {F}iltering {L}ibrary," http://www.orocos.org/bfl, 2001.