Wet Whale: Hey, where am I??

1st Hellwig Jonathan	2nd Adamanov Asan	3rd Dierfeld Sven	4rd Dirikgil Tolga
Wet Whale	*Wet Whale*	*Wet Whale*	*Wet Whale*
UH	*TUHH*	*TUHH*	*TUHH*
7381194	11861429	21595967	21158403
jonathan.hellwig@studium.uni-hamburg.de	asan.adamanov@tuhh.de	sven.dierfeld@tuhh.de	tolga.dirikgil@tuhh.de

***Abstract*—This paper proposes three localization approaches for an underwater ROV living in a rectangular tank with apriltags attached to a wall and the floor. The approaches are based on a Least-Square approach and Kalman filters using the range, IMU, and pressure data. After the three localization algorithms are proposed, advantages and drawbacks of each of them are discussed and then all three approaches are implemented. Furthermore, a PID-controller for each translation axis and for the yaw rotation is implemented.**

***Index Terms*—underwater ROV, localization, Least Square, (linear) Kalman Filter, Extended Kalman Filter, PID-Controller.**

I. Introduction

The possible benefits of Remotely Operated Underwater Vehicles (ROV) seem to be sheer endless. The working conditions of those robots will be especially conditions that are too dangerous for humans while cutting the costs for expensive human-carrying submarines. Search and Rescue, Military, Recreation and Discovery, Aquaculture, Marine Biology, Oil, Gas, Offshore Energy, Shipping, Submerged Infrastructure are all areas that can benefit greatly from well working ROVs. These underwater applications come with their challenges though. Localization underwater turns out to be more complicated than for example, for autonomous cars or drones. Common GPS does not work underwater as the radio frequency signals can not propagate. Furthermore, the ROVs are often pushed around by external forces like currents or waves. Therefore, more sophisticated algorithms are needed to estimate its relative or inertial position in the environment. To find applicable solutions for those problems the least square method, the least square method fused with a linear Kalman filter and an extended Kalman filter will be compared in this report as localization methods.

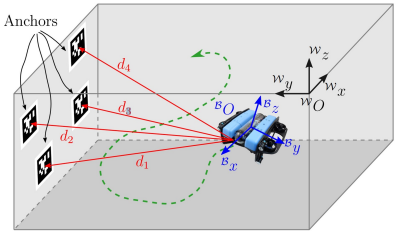


Fig. 1: Experimental design.

II. Methods

A. Experimental design

To test the proposed approaches for the localization the BlueROV2 by Bluerobotics is used. It is equipped with a front and a downwards camera, a pressure sensor, an IMU sensor, and lights. The goal is to enable the ROV to localize itself in a 3.25 x 1.67 x 1.4 m tank and automatically dive to given locations in the tank. The distances to four apriltags attached to a wall are to be used to estimate the position of the ROV (see Figure 1). The pose is assumed to be given.

B. Localization

1) Least squares: Our state vector $\mathbf{s} \in \mathbb{R}^6$ is given by

$$\mathbf{s} = \begin{bmatrix} x & y & z & v_x & v_y & v_z \end{bmatrix}^T \tag{1}$$

In the following let

$$\mathbf{s_p} = \begin{bmatrix} x & y & z \end{bmatrix}^T \tag{2}$$

denote the first three components and

$$\mathbf{s_v} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T \tag{3}$$

the last three components of $\mathbf{s}$. Given range measurements $d_1, d_2, d_3, d_4 \in \mathbb{R}$ between the tag and the camera of the ROV , the tag positions defined by $\mathbf{r_1}, \mathbf{r_2}, \mathbf{r_3}, \mathbf{r_4} \in \mathbb{R^3}$ and a depth measurement $m_{\text{Depth}} \in \mathbb{R}$ obtained by a pressure sensorwe can formulate the nonlinear least squares problem [1]

$$\min_{\mathbf{s_p} \in \mathbf{T}} \sum_{k=1}^{4} |||\mathbf{s_p} - \mathbf{r_k}||_\mathbf{2} - \mathbf{d_k}|^\mathbf{2} \tag{4}$$

$$\text{s.t. } z = m_{\text{Depth}} \tag{5}$$

where $\mathbf{T} \subset \mathbb{R^3}$ is given by the tank size and $||\cdot||_2$ is the standard euclidean norm in $\mathbb{R}^3$. Using a first order Talyor approximation we obtain

$$||\mathbf{s_p} - \mathbf{r_k}||_\mathbf{2} \approx ||\hat{\mathbf{s}}_\mathbf{k}||_\mathbf{2} + \frac{\hat{\mathbf{s}}_\mathbf{k}}{||\hat{\mathbf{s}}_\mathbf{k}||_\mathbf{2}} \cdot (\mathbf{s_p} - \mathbf{r_k} - \hat{\mathbf{s}}_\mathbf{k}) \tag{6}$$

where $\hat{\mathbf{s}}_\mathbf{k} \in \mathbf{T}$ is chosen to be the estimate of the previous iteration. Using this linearization we can reformulate the least squares problem to

$$\min_{\mathbf{s_p}\in\mathbf{T}} \sum_{k=1}^{4} |\frac{\hat{\mathbf{s}}_\mathbf{k}}{||\hat{\mathbf{s}}_\mathbf{k}||_\mathbf{2}} \cdot \mathbf{s_p} + ||\hat{\mathbf{s}}_\mathbf{k}||_\mathbf{2} - \mathbf{d_k} - \frac{\hat{\mathbf{s}}_\mathbf{k}}{||\hat{\mathbf{s}}_\mathbf{k}||_\mathbf{2}} \cdot (\mathbf{r_k}| + \hat{\mathbf{s}}_\mathbf{k})|^\mathbf{2} \tag{7}$$

. This problem is a linear least squares problem with box constrains and can be solved using a standard optimization toolbox. The center $\hat{\mathbf{s}}_\mathbf{k}$ is chosen to be the previously estimated position. Furthermore we average the last two measurements of the ranges to reduce the impact of outliers.

2) Least squares and Kalman filter: Even though the least-squares method gives a good approximation of the localization while the tags are being seen, a problem occurs if the robot is not able to detect any tags. If there is not any range measurement for a certain tag, the related equation is dropped from the least-squares method, which results in an under-determined position estimate $\mathbf{s_p}$ which jumps since it is not able to predict the exact ROVs position. Moreover, the range measurements which come from the camera are slow, due to the low working frequency of the camera sensor. This makes controller approaches that are dependent on the derivative of the specific state measurement ineffective and results in a slow or overshoot prone step response.

These problems can be easily solved with Gaussian filters as the Kalman filter or with non-parametric filters [3]. Besides, these filters allow us to take other sensors into account that are installed on the robot e.g IMU, Depth sensor, and fuse them to improve localization performance. We decided to extend our least-squares method by integrating a linear Kalman filter on top. This is done by using s_p from equation 7 as input for the Kalman filter. The Kalman filter is being represented in moments form. The moment is representation consists of the mean (first moment) and the covariance (second moment) of the Gaussian. Furthermore, as it can be seen from the family name of Kalman filter, it does have a strong belief in a Gaussian such that initial belief, motion model, and measurements have to be Gaussian.

The implementation of the Kalman filter consists of two-steps: prediction and update. First, based on the current step, the prediction of the next step has to be done. When a measurement of any sensor is received, an update step is performed to correct the current prediction. In the following let $\mathbf{F}$ denote the state-transition matrix, $\mathbf{H}$ the observation matrix, $\mathbf{Q}$ the model covariance matrix, $\mathbf{R}$ the measurement covariance matrix, $\mathbf{K}$ the optimal Kalman gain, $\mathbf{S}$ innovation covariance matrix , $\mathbf{s}'$ the predicted state and $\mathbf{P}$ the updated covarinace matrix of the state. Using these terms, we can formulate the following equations:
Prediction step

$$\mathbf{s}' = \mathbf{F}\mathbf{s} \tag{8}$$

$$\mathbf{P}' = \mathbf{F}\mathbf{P}\mathbf{F}^\mathbf{T} + \mathbf{Q} \tag{9}$$

Measurement update

$$\mathbf{y} = \mathbf{z} - \mathbf{H}\mathbf{s}' \tag{10}$$

$$\mathbf{S} = \mathbf{H}\mathbf{P}'\mathbf{H}^\mathbf{T} + \mathbf{R} \tag{11}$$

$$\mathbf{K} = \mathbf{P}'\mathbf{H}^\mathbf{T}\mathbf{P}\mathbf{S}^{-\mathbf{1}} \tag{12}$$

$$\mathbf{s} = \mathbf{s}' + \mathbf{K}\mathbf{y} \tag{13}$$

$$\mathbf{P} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}' \tag{14}$$

Since on the current stage of an assignment, the consideration of angular motion is not required. We use a simple constant velocity model where we define the state transition matrix $\mathbf{F}$ by

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

where Δt denotes the time passed between each prediction step. The transformation for a LSE based measurement update .

$$\mathbf{H_{LSE}} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{16}$$

The transformation for an IMU-based measurement update

$$\mathbf{H_{IMU}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

We decided to fuse measurements from two different sensors: the range measurement $d_1, d_2, d_3, d_4 \in \mathbb{R}$ based on the camera which is converted into x, y, z through least square method and the velocity measurements v_x, v_y, v_z from the IMU. Both sensors have different update time steps, if we have had a limited computation source on our hardware, this would have caused a big problem in real-time applications. However, since our code is run on the off-board computer it is possible to run the prediction step continuously and the updating step only whenever the measurement is gotten from one of the sensors to improve our localization estimate.

3) Extended Kalman filter: Another option, which is even more straight forward and less computationally expensive, is to feed the range measurements directly into an Extended Kalman filter, instead of first calculating a position estimate using the LSE. For that, we only needed to transform the range measurements $\mathbf{r}$ into x, y, z states in the measurement update part. Measurement update for EKF becomes

$$\mathbf{y} = \mathbf{z} - \mathbf{h}(\mathbf{s}') \tag{18}$$

$$\mathbf{S} = \mathbf{J_h}\mathbf{P}'\mathbf{J_h^T} + \mathbf{R} \tag{19}$$

$$\mathbf{K} = \mathbf{P}'\mathbf{J_h^T}\mathbf{P}\mathbf{S}^{-\mathbf{1}} \tag{20}$$

$$\mathbf{s} = \mathbf{s}' + \mathbf{K}\mathbf{y} \tag{21}$$

$$\mathbf{P} = (\mathbf{I} - \mathbf{K}\mathbf{J_h})\mathbf{P}' \tag{22}$$

The positions of the four tags in the tank are given by $\mathbf{r_1}, \mathbf{r_2}, \mathbf{r_3}, \mathbf{r_4} \in \mathbb{R}^3$. We define two state to measurement functions $h : \mathbb{R}^6 \longrightarrow \mathbb{R}^4$ and $H : \mathbb{R}^6 \longrightarrow \mathbb{R}$ by

$$h(\mathbf{s}) = (||\mathbf{s_p} - \mathbf{r_k}||_\mathbf{2})_{\mathbf{k=1}}^{\mathbf{4}} \tag{23}$$

The function h maps the state $\mathbf{s}$ to the range measurements $d_1, d_2, d_3, d_4 \in \mathbb{R}$. Clearly, the function h is nonlinear. Therefore, we have to have to make use of an Extended Kalman filter using a first-order Taylor approximation in order to linearize the nonlinear function. The Jacobian $\mathbf{J_h}$ of h is given by

$$\mathbf{J_h}(\mathbf{s}) = \begin{bmatrix} \frac{x-\mathbf{r_{1x}}}{||\mathbf{s_p}-\mathbf{r_1}||_\mathbf{2}} & \frac{y-r_{1y}}{||\mathbf{s_p}-\mathbf{r_1}||_\mathbf{2}} & \frac{z-r_{1z}}{||\mathbf{s_p}-\mathbf{r_1}||_\mathbf{2}} & 0 & 0 & 0 \\ \frac{x-r_{2x}}{||\mathbf{s_p}-\mathbf{r_2}||_\mathbf{2}} & \frac{y-r_{2y}}{||\mathbf{s_p}-\mathbf{r_2}||_\mathbf{2}} & \frac{z-r_{2z}}{||\mathbf{s_p}-\mathbf{r_2}||_\mathbf{2}} & 0 & 0 & 0 \\ \frac{x-r_{3x}}{||\mathbf{s_p}-\mathbf{r_3}||_\mathbf{2}} & \frac{y-r_{3y}}{||\mathbf{s_p}-\mathbf{r_3}||_\mathbf{2}} & \frac{z-r_{3z}}{||\mathbf{s_p}-\mathbf{r_3}||_\mathbf{2}} & 0 & 0 & 0 \\ \frac{x-r_{4x}}{||\mathbf{s_p}-\mathbf{r_4}||_\mathbf{2}} & \frac{y-r_{4y}}{||\mathbf{s_p}-\mathbf{r_4}||_\mathbf{2}} & \frac{z-r_{4z}}{||\mathbf{s_p}-\mathbf{r_4}||_\mathbf{2}} & 0 & 0 & 0 \end{bmatrix} \tag{24}$$

The transformation for a depth based measurement update.

$$\mathbf{H_{Depth}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{25}$$

The transformation for a IMU based measurement update.

$$\mathbf{H_{IMU}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{26}$$

The covariance matrices $\mathbf{R}$ for the model $\mathbf{Q}$ are chosen to diagonal matrices.

After testing our algorithm in a simulation. We figured out that we have to apply a modification to how an EKF processes a range measurement vector $h(\mathbf{s}')$. Constructing a static $h(\mathbf{s}') \in \mathbb{R}^\mathbf{4}$ vector leads to an undesirable results of EKF. Hence we build a dynamical $h(\mathbf{s}')$ vector that changes its size based on the arrived amount of measurements. The dimension vectors and matrices in EKF change themselves accordingly to the arrived amount of range measurements. An alternative version of updating range based measurement according to [3], is to loop over-range measurement vector and update each time a state estimation according to a single arrived measurement. The idea behind this version is, instead of processing as a whole measurement vector directly, each time improve your localization performance.

C. Controller

We decide to use four separate PID controllers to control displacement in the x,y, z-direction, and the yaw rotation. This allows us to control the ROV without knowing the internal dynamics but makes it necessary that the axis of the ROVs relative coordinate system overlap with the axis of the inertial coordinate system. For that reason, and also to make sure the localization works optimally. It is important to ensure that the camera is facing the tags on the wall. To tune the PID controllers, we used the Ziegler-Nichols method, which is known to yield an aggressive controller with some overshoot. [2] To avoid collisions with the walls, we restricted set points to a save zone.

D. Dependencies of camera position and Baselink position

The localization estimates the position of the camera in the inertial coordinate system of the tank. For further usage, it is more practical to use the base of the ROV as location. After estimating the position of the camera in the inertial coordinates of the tank using the Kalman filters, we can do a coordinate transformation of the base link-vector, given in the relative coordinate system of the ROV, into the inertial system using the orientation of the ROV from the IMU. Then we can add these two vectors to calculate the position of the base.

III. Results

In Figure 2 is shown, how the localization algorithm which is based on the extended Kalman filter estimates the BlueROV2 to be out of the tank. Because we are using the ranges and the depth as information for the location of the ROV there are two possible solutions for the y-position mirrored at the plane the tags are in. This problem is solved by simply mirroring the calculated position at said plane if y position is greater than the actual length of the tank in the y-direction.

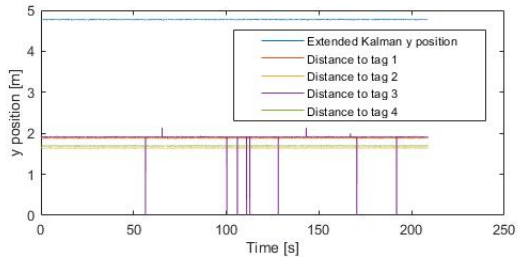


Fig. 2: Kalman filter estimating the ROV to be out of the tank.

Figures, 3, 4 and 5 show the result from comparing the least-squares approach, the Kalman filter in combination with the least-squares approach, and the extended Kalman filter as localization methods. All positioning of the BlueROV2 is done by hand with the help of a broom.
The first thing to be recognized is the least-squares approach showing much more noise than the other methods. Furthermore, Figure 3 shows some discontinuous course of curve around 10 seconds resulting from turning the ROV away from the tags so that there are no tags seen at all while trying to keep its x position. The position results are estimated by the least-squares approach, and the Kalman filter in combination with the least-squares approach diverging quickly from its actual position and only the extended Kalman filter estimating the x position in a reasonable manner.
Once the tags are seen again, the extended Kalman filter and least-squares approach estimate similar positions, and the Kalman filter in combination with the least-squares approach is getting to this similar estimation with a slight oscillating lag.

Figure 4 shows the same discontinuity around 10 seconds as Figure 3 resulting from turning the ROV away from the tags, as well. The important thing to be recognized here is

the large difference between the y-position estimated by the least-squares approach, the Kalman filter in combination with the least-squares approach, and the Extended Kalman filter.
To verify which of the three methods estimates the most plausible position. The distance from the position of the tags to the middle of the BlueROV2 in the y-direction is measured. With this distance being $l_{tags\ to\ ROV} = 1.8\ m$ and with the distance in the y-direction from the tags to the origin of the coordinate system we used is $l_{tags\ to\ origin} = 3.35\ m$ the y coordinate of the ROV can be calculated using

$$y_{ROV} = l_{tags\ to\ origin} - l_{tags\ to\ ROV} = 1.55\ m \qquad (27)$$

which verifies the extended Kalman filter showing the most plausible position. The other two methods estimating an obviously wrong y-position might result from a mistake in our least-squares approach algorithm. As for estimating the position in the z-direction Figure 5 shows very similar estimations by all of the three methods. As the depth is set as a constraint for the LSE and the EKF also uses an update step for the depth the results in the z-direction are highly similar.

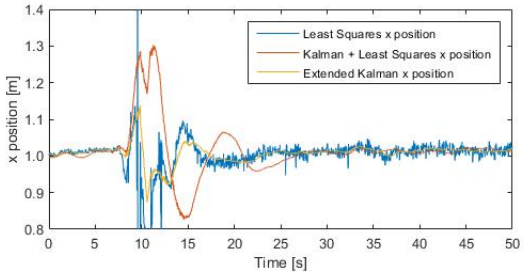


Fig. 3: Comparison of different localization methods in x direction.

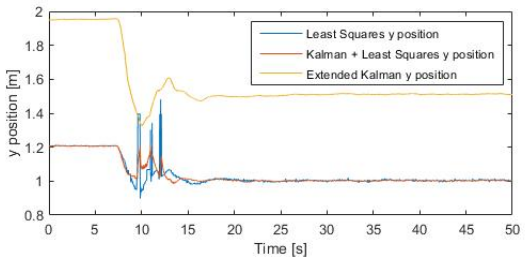


Fig. 4: Comparison of different localization methods in y direction.

Figures 6a, 6b, 6c and 6d show the tuning of the PID parameters for motion control of the ROV for the motions thrust, lateral thrust, vertical thrust and yaw with step inputs. For the position estimate we use the EKF. These figures show overshoots and steady state errors being small enough for using the BlueROV2 in the tank environment with our safe zone parameters. In table I the values for the proportional gain k_p, the derivative gain k_d and the integral gain k_i for each PID-Controller can be seen.

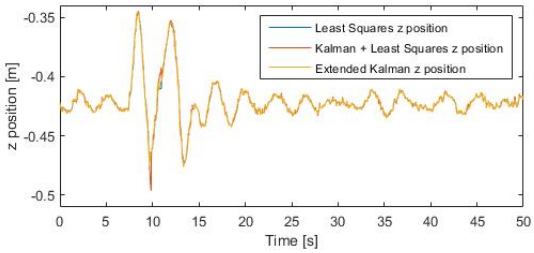


Fig. 5: Comparison of different localization methods in z direction.

	vertical thrust	lateral thrust	thrust	yaw
k_p	4	3	2,6	1,4
k_d	0,006	0,03	0,03	0,008
k_i	0,018	0	0	0,022

TABLE I: Parameters for the PID-Controllers from the experiment.

IV. Conclusion

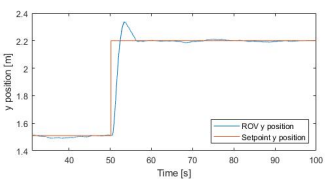
The major disadvantage of the least-squares approach is its lack of robustness. Because of this and the obvious error in our least-squares approach algorithm to be seen in Fig. 4 the localization method used in our final project will be the extended Kalman Filter. To improve this filter and being able to predict the yaw angle as well, we will change the tag detection algorithm to work with the bottom camera for detecting the bottom tags. Furthermore, implement green a model that takes the orientation of the ROV into account a constant turn rate and acceleration (CTRA) model .

References

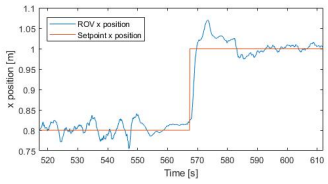
[1] B. Jonathan, and C. Taylor, "Localization in Sensor Networks," pp. 277–310, 2005

[2] Ziegler, John G., and Nathaniel B. "Optimum settings for automatic controllers. Nichols," 1942

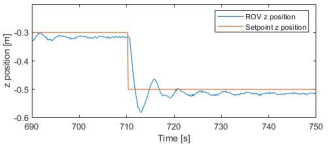
[3] S. Thrun, W. Burgard, D. Fox, and R. Arkin, "Probabilistic Robotics," MIT Press, 2005



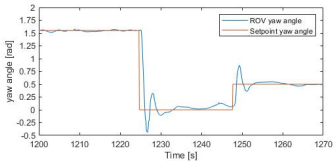
(a) y-direction



(b) x-direction



(c) z-direction



(d) yaw-angle

Fig. 6: Step response of the ROV for all directions.