Online Kinematic and Dynamic Parameter Estimation for Autonomous Surface and Underwater Vehicles

Anwar Quraishi, Alcherio Martinoli

***Abstract*— One of the main challenges in underwater robot localization is the scarcity of external positioning references. Therefore, accurate inertial localization in between external position updates is crucial for applications such as underwater environmental sampling. In this paper, we present a framework for estimating kinematic and dynamic model parameters used for inertial navigation. Accurate values of these parameters result in better trajectory estimation. Our approach can run online as well as offline, with either choice providing different advantages. Further, our framework can correct errors in the past trajectory at each estimation step. By doing so, we are able to provide improved geo-references for past as well as future spatial measurements made by the robots. This has an impact on adaptive sampling methods, which use geo-tagged measurements for building local spatial distributions and choose future sampling points. We present results from field experiments and demonstrate improvement in trajectory estimation accuracy. We also experimentally show that with optimal parameter estimates, robots can tolerate longer intervals in external positioning updates for a specified acceptable level of estimation error.**

I. Introduction

Application of mobile robots to sensing tasks in any environment brings several advantages [1]. Not only does the use of robots scale better than static sensing nodes in terms of spatial coverage [1], but robots can also specifically target regions of higher interest for gathering data [2], [3], [1]. We have developed the Vertex Autonomous Underwater Vehicle (AUV) [4] in our laboratory, a miniature, easy to deploy platform equipped with an environmental sensing payload. However, localization in underwater environments is a major challenge, especially with miniature AUVs that lack high-accuracy sensors such as a fiber-optic gyroscope or a Doppler velocity log. Satellite-based positioning is unavailable when the robots are submerged. Visually distinct features are sparse in natural water bodies, and turbidity may hinder visibility. Employing acoustic signals is usually the method of choice for external position referencing in the underwater domain, but, the drawback is that the time between successive acoustic updates is often several seconds. Therefore, accurate inertial localization is crucial for underwater robots.

Although inertial sensors of high accuracy may be sufficient to perform inertial localization, sensing modules compatible with miniature AUVs (e.g., MEMS based devices) do not offer the required accuracy. Fusing these measurements with a dynamic model of the robot enhances the accuracy of the position estimates [5]. However, this requires knowledge of a wide range of parameters that make up such a model. Measuring or estimating some of the parameters (such as hydrodynamic constants) can be difficult, time-consuming or even require special equipment. Further, some parameters such as mass and actuator properties may differ from one robot to another. They may also change, for example when new equipment or sensors are added on the robot.

In this paper, we present a method for automatically estimating kinematic and dynamic parameters online using only on-board sensors. In doing so, our objectives are three-fold. Firstly, we seek to improve the accuracy of inertial localization in between successive external position updates. Secondly, we aim to correct errors in the past trajectory by recomputing it using the newly obtained parameter estimates. This is important for adaptive sampling methods such as [6], which rely on accurately geo-referenced past measurements to decide future sampling locations. Finally, we aim to calibrate the parameters for several robots. Identical robots may have slightly different parameters owing to different on-board equipment and differing wear and tear over time.

A number of methods exist in the state-of-the-art for identifying model parameters of robots. Kelly et. al. in [7] estimate geometric and camera parameters using camera images and IMU measurements. They add the parameters to be estimated as state variables to an Unscented Kalman Filter (UKF) framework. A similar approach is presented in [8]. An approach based on error-minimization for sensor calibration is presented in [9], which is applied offline to recorded datasets. Authors in [10] take an information theoretic approach to asses the usefulness of robot operation data before using them for re-calibration. The latter two perform parameter identification offline. All of the above use camera images, which are used to deduce relative pose between successive measurements.

Approaches such as [11], [12], [13] attempt to estimate dynamic or inertia parameters, usually to be fed into a model-predictive control module. In [11], the authors rely on an unspecified absolute pose sensor to estimate parameters using an Extended Kalman Filter (EKF). In [13], a motion capture system is used to observe absolute robot poses. Burri et. al. in [14] use a Maximum Likelihood approach for parameter identification, which is performed offline. They use a camera, IMU and an external tracking system.

Online methods such as [7], [11] often estimate parameters as additional state variables in an EKF framework.

Distributed Intelligent Systems and Algorithms Laboratory (DISAL), School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland. This work was partially funded by the Swiss National Science Foundation under grant CRSII2_160726/1. disal.epfl.ch/research/AUVDistributedSensing.

A drawback of such approaches is that the parameters are rarely directly observed, especially if the measurements are sparse. In order to cope with this partial observability, ad hoc workarounds such as inclusion of intermediate state variables are necessary.

In this work, we used GNSS or an acoustic range and bearing system for obtaining absolute position. Suitable tracking systems for accurate pose measurements at a high frequency for outdoor aquatic environments are unavailable. Indoor or controlled environments such as pools are too small to perform adequate motion. Both GNSS and acoustic systems are less accurate and have low update rates (100 ms for GNSS, a few seconds for acoustics) compared to the systems used in the aforementioned works. Note that they also only measure position, not full pose.

Our work takes an approach similar to other error-minimization based methods. However, in order to deal with sparse position measurements, a timestamped history of all robot control and measurement updates is maintained. We use the sum of Kalman update errors (product of Kalman gain and innovation) over a window of time as the error metric to be minimized. When new external position information is available, error minimization is performed over a specified window of the measurement history that includes several position measurements.

This approach offers several advantages. To begin with, the error metric accounts for uncertainty in measurements and state estimates. This is crucial for acoustic positioning devices, since the uncertainty in their measurements is directional and dependent on relative positions of acoustic beacons and robots, as explained later. In addition, this formulation makes it easy to integrate our method into any kind of state estimation framework with minimal modification to it. In this paper, we plugged the parameter estimation framework to an existing EKF framework. Finally, our approach allows for correction of possible errors in past trajectory by tracing a desired length of the history and recomputing past estimates using updated parameters.

The proposed method can run both online or offline, each providing different benefits. Online execution uses onboard information to estimate parameters on-the-fly, but the computational budget is limited. They are suited for parameters that can change quickly such as drift in yaw angle. Offline execution, on the other hand, is not constrained by computational cost, and can perform estimation using a larger amount of data. They are well suited for parameters that are likely to remain constant or change rarely, such as physical properties of the robot.

In the rest of this paper, we describe our method in detail, and evaluate it using real world experiments as well as simulations in which real-world data are plugged in.

II. Relevant System Components

A. Robots

Two different robot platforms are used – the Vertex AUVs [4] and Autonomous Surface Vehicles (ASVs), shown in Fig. 1. The AUVs and ASVs share the same sensing and computation hardware. The ASVs serve as mobile acoustic beacons, or are used as surrogates for AUVs for experimental verification of new methods, using GNSS as ground truth.

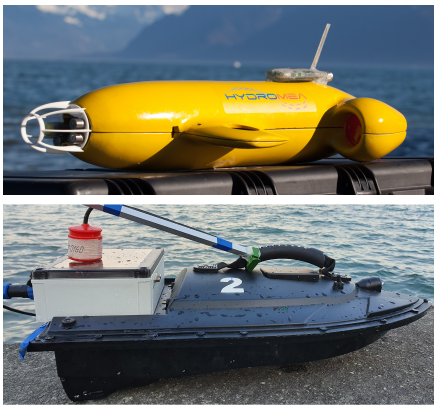


Fig. 1: Above: The Vertex AUV with its sensor suite in the front. The AUV is about 70 cm long, weighs 7 kg. Below: The ASV with an acoustic transceiver. The ASV is about 50 cm long and weighs around 2 kg. The acoustic transceiver is suspended in water from the rear of the robot.

The AUVs weigh 7 kg, are 0.7 m in length and are actuated by five thrusters. Two thrusters provide forward propulsion and three are used for attitude control. The ASVs weigh 2.5 kg and are 0.6 m long. They are actuated by two propellers, which provide differential-drive-like maneuvering.

B. Sensing

The robots are equipped with an Xsens MTi-1 MEMS IMU for inertial measurements. Along with acceleration and angular velocity, it also provides a fused attitude estimate. The accuracy along the roll and pitch axes is $0.5°$. A magnetometer provides referenced yaw angle with an accuracy of $2.0°$, but owing to magnetic disturbances induced by motor current, it is not used. Therefore, we need to track drift in yaw angle in addition to knowing initial heading.

A uBlox GNSS receiver provides absolute position with an accuracy of around 1 m on the surface. On the AUV, a pressure-based depth sensor provides an accurate measurement of depth. A DiveNET Commander acoustic positioning system [15] provides range and bearing measurements. It uses a base station equipped with an Ultra Short Baseline (USBL) receiver array for measuring range and bearing to a target. It transmits these measurements back to the target, which is connected to the robot. Accuracy characteristics will be shown later in the paper.

C. Coordinate systems

In our calculations, we use two different coordinate systems – a ground-fixed, local level frame with an NED (North-East-Down) axes convention, and a robot-fixed moving frame whose axes definition is shown in Fig. 2. We denote the frame of reference of a vector by a subscript ‘g’ and ‘b’ for ground-frame and robot-body-frame respectively. Rotations are represented using the quaternion notation.

D. Dynamics model

In between acoustic or GNSS updates, a dynamics model and IMU measurements are fused using an EKF framework

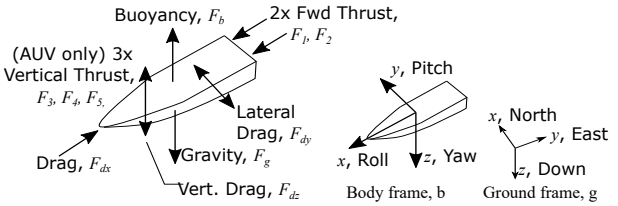


Fig. 2: Left: Forces on the robots (AUV or ASV). Right: Body frame and ground frame (NED) coordinate systems.

for inertial localization. We use a simplified dynamics model that uses motor commands to compute thrust, and takes into account mass, inertia and hydrodynamic drag forces. Fig. 2 shows the forces acting on the robots.

1) Linear dynamics: First, motor commands, $[u_1, u_2, ...u_i]$ are sent to the motors in terms of the desired rotational speed of each propeller. We compute the thrust produced by each propeller i as

$$F_i = k u_i^2, \tag{1}$$

where k is the thrust constant of the propellers, and $i \in [1, 2]$ for the ASV, $i \in [1, 5]$ for the AUV. All propellers on a particular robot are identical. We calculate the forces due to viscous drag on the robot body in the robot frame $\vec{F}_{\text{drag}}$ as

$$\vec{F}_{\text{drag}} = \begin{bmatrix} -\text{sign}(v_{rx}) \cdot k_x v_{rx}^2 \\ -\text{sign}(v_{ry}) \cdot k_y v_{ry}^2 \\ -\text{sign}(v_{rz}) \cdot k_z v_{rz}^2 \end{bmatrix}, \tag{2}$$

where k_x, k_y, k_z are the viscous drag constants along the body axes and $\vec{v}_{r,\text{b}} = [v_{rx}, v_{ry}, v_{rz}]^{\text{T}}$ is the relative velocity between the robot and water stream, in the body frame. Essentially, in each direction, the drag force is proportional to the square of this relative velocity and acts in a direction opposite to it. $\vec{v}_{r,\text{b}}$ is computed by appropriate rotation of the said relative velocity in the ground frame as

$$\vec{v}_{r,\text{b}} = \mathbf{q}^{-1} \left(\vec{v}_{\text{g}} - \vec{s}_{\text{g}} \right) \mathbf{q}, \tag{3}$$

where $\vec{v}_{\text{g}}$ is the estimated velocity of the robot and $\vec{s}_{\text{g}} = [s_x, s_y, s_z]$ is the water stream velocity, both in the ground frame. $\mathbf{q}$ is the quaternion representing robot attitude, and the operation $\mathbf{q}^{-1} (\vec{v}) \mathbf{q}$ rotates the vector $\vec{v}$, transforming it from ground frame to body frame representation.

We assume that the drag constants k_y and k_z are equal, and we refer to it with k_{yz}. This is warranted since the AUV geometry along those axes is largely similar, and ASV has no vertical motion. We also assume $s_z = 0$.

The AUVs are deliberately trimmed to be slightly positively buoyant, and ASVs are floating devices. We assume that the weight of the robot nearly cancels out buoyancy, and hence, both terms are ignored. The net force is then a function of five quantities.

$$\vec{F}_{\text{net,b}}(k, k_x, k_{yz}, s_x, s_y) = \left(\vec{F}_{\text{thrust}} + \vec{F}_{\text{drag}} \right). \tag{4}$$

2) Rotational dynamics: The shape of the AUV is streamlined only along its longitudinal (x) axis. As a result, rotation about all three axes is highly damped. Further, the AUV is trimmed in a way that the centers of mass and buoyancy coincide. As such, the rotational motion of the AUV is slow enough that the update rate of the IMU (100 Hz) is sufficient to compute orientation. Therefore, we do not consider rotational forces in the dynamics model, instead relying only on the orientation computed internally and reported by the Xsens IMU module. A correction is applied to the yaw orientation, as explained in the next section.

E. EKF framework

1) State Vector: The state variable comprising of the position and velocity in ground frame and acceleration in the body frame is formulated as

$$\mathbf{X} \triangleq \left[\vec{x}_{\text{g}}^{\text{T}}, \vec{v}_{\text{g}}^{\text{T}}, \vec{a}_{\text{b}}^{\text{T}} \right]^{\text{T}} \tag{5}$$

We do not include the variables representing attitude since they are not predicted by the model in this work. When predicting angular motion in the model, the state can be expanded correspondingly to include the attitude variables.

2) Attitude: The orientation reported by the Xsens IMU is used directly. A correction θ_e is applied to the yaw (heading) orientation, since it is both unreferenced (since we do not use the magnetometer), as well as drifts over time. The corrected attitude is computed as

$$\mathbf{q} = \mathbf{q}(\theta_e) \mathbf{q}_{\text{raw}}, \tag{6}$$

where $\mathbf{q}(\theta_e)$ is the quaternion corresponding θ_e, which is estimated separately as a parameter.

3) Process update: A process model $f(.)$ is used to compute the future state as

$$\hat{\mathbf{X}}^{t+dt} = f\left(\mathbf{X}^t, dt, \mathbf{P} \right), \tag{7}$$

where t denotes time and $\mathbf{P}$ is the set of parameters which is introduced later in Section III-A. Some of the parameters were used earlier in Eq. (2). The individual state components are computed as

$$\hat{x}_{\text{g}}^{t+dt} = \vec{x}_{\text{g}}^{t} + \vec{v}_{\text{g}}^{t} \cdot dt, \tag{8}$$

$$\hat{v}_{\text{g}}^{t+dt} = \vec{v}_{\text{g}}^{t} + \mathbf{q}\left(\vec{a}_{\text{b}}^{t} \right) \mathbf{q^{-1}} \cdot dt, \tag{9}$$

$$\hat{a}_{\text{b}}^{t+dt} = (1/m) \cdot \vec{F}_{\text{net, b}}, \tag{10}$$

where $\mathbf{q}$ is the attitude quaternion introduced earlier in Eq. (6), used here for the inverse operation. The acceleration is calculated using the dynamics model in Eq. (4). The covariance matrix representing motion noise (not shown here) is determined empirically.

4) Measurements: Measurement updates are performed individually for IMU, GNSS or acoustic information, using independent measurement models. The 'update error' e_i is defined as the product of the Kalman gain and innovation.

$$e_i = K_i \left(z_i - h(\hat{\mathbf{X}}) \right), \tag{11}$$

where z_i is the ith measurement, K_i is the Kalman gain computed during this measurement, and $h(.)$ is the measurement model of a particular sensor. The sum of errors corresponding to a set of measurements is used later in the optimization framework.

Variable	Description
θ_e	Offset between actual and estimated heading
s_x	Water stream velocity x (ground frame)
s_y	Water stream velocity y (ground frame)
$k' = k/m$	Ratio of thrust constant to mass
$k'_x = k_x/m$	Ratio of drag constant to mass
$k'_{yz} = k_{yz}/m$	Ratio of drag constant to mass

TABLE I: List of variables to be estimated.

5) Initialization: The initial position is obtained form GNSS before launching the robots. The initial velocity and acceleration can be assumed to be zero. The initial heading, if unknown, will result in a large θ_e, which will be estimated as a parameter eventually. The gyro biases are internally estimated by the XSENS IMU module by leaving the robot stationary for a few seconds.

III. Methodology

This section formulates parameter estimation as an optimization problem, and describes our approach in detail.

A. Parameters to estimate

The set of parameters to be estimated, $\mathbf{P}$, consists of the following variables.

$$\mathbf{P} \triangleq \left[\theta_e, s_x, s_y, k', k'_x, k'_{yz}\right]. \tag{12}$$

Table I provides a description of the parameters. The last three variables correspond to the dynamic model parameters in Eq. (4) and always appear as a ratio with mass. Therefore, the ratio itself is estimated.

The parameters can be categorized into two types – *fixed-parameters* do not change often or during operation, such as the vehicle model parameters, while *variable-parameters* change depending on the situation or during operation, such as the yaw drift and water stream velocities.

B. History management

We maintain a timestamped, chronological history of information pertaining to robot state updates, as illustrated in Fig. 3. This includes all the measurements (IMU, acoustics, GNSS etc.) as well as motor commands output by the controller. Each of these entities is called an *update node*, denoted by u_i^t, where i denotes the index of the node, and t the timestamp. Note that multiple update nodes may have the same timestamp, for example, when there are concurrent measurements from two sensors. We also record the initial state of the robot at $t = 0$. Additionally, we record the state at points of time when absolute position measurements (GNSS or acoustics) are received. These are called *anchor states*.

Starting from an appropriate anchor state, we can use the chain of update nodes for successive EKF updates to compute the estimated robot state, $\hat{\mathbf{X}}$, at any point of time in history. We will also need the values of parameters in $\mathbf{P}$.

$$\hat{\mathbf{X}}^{t_2} = \mathcal{F}_{t_1}^{t_2}\left(\mathbf{X}_a^{t_1}, \mathcal{U}, \mathbf{P}\right), \tag{13}$$

where $\mathcal{F}_{t_1}^{t_2}$ is some function that performs this computation from t_1 to t_2, and $\mathcal{U} \coloneqq \{u_i^t \mid t \in [t_1, t_2]\}$ is the sequence of all the update nodes between the two points of time. In

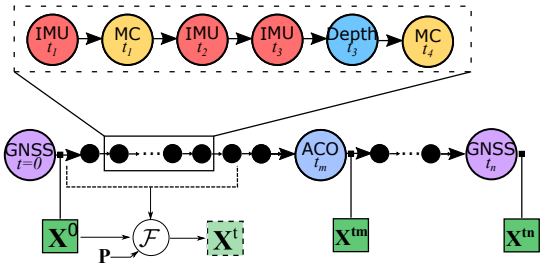


Fig. 3: Timestamped history consisting of update nodes representing state transition information. Each colored circle corresponds to an update node, for e.g., IMU measurements, Motor Commands (MC), etc. Anchor states, shown in dark green squares, are recorded after an acoustic or a GNSS position update. The function $\mathcal{F}$ computes the state at t using the previous anchor state and the intermediate sequence of update nodes.

Fig. 3, the computation of state at t is illustrated. Note that we treat motor commands as a type of measurement, and define the corresponding update error to be zero.

C. Optimization

A series of EKF updates are performed in Eq. (13). The net error in calculation of the estimated state is defined as a function of the parameters, $\mathbf{P}$ as

$$E_{[t_1,t_2]}(\mathbf{P}) = \sum_{\mathcal{U}} \|e_i\|^2, \tag{14}$$

where $\mathcal{U}$ is the set of nodes within the time window from t_1 and t_2, and e_i is the update error corresponding to node i, as computed in Eq. (11). Note that e_i is computed as a product of Kalman gain and innovation. The key advantage of this is that the uncertainties in the measurement and the state estimate are accounted for, which is important for acoustic positioning measurements, which have non-constant error characteristics. For a measurement with a larger uncertainty, the resulting error will be lower.

We seek to find values of parameters $\mathbf{P}$ that result in the most accurate inertial positioning. To do so, we need the inertial estimates to be in agreement with absolute position information. We achieve this by minimizing the measurement update error in Eq. (11) over a time window $[t_1, t_2]$, spanning several GNSS or acoustic measurements. Formally, we seek to solve the following optimization problem:

$$\mathbf{P}^* = \arg\min_{\mathbf{P}} \; E_{[t_1,t_2]}(\mathbf{P}). \tag{15}$$

For a valid solution, at least two absolute position measurements are required within the time window.

D. Past trajectory correction

Once optimal parameter estimates are obtained for a particular window $[t_1, t_2]$, the past states are recomputed again with the newly obtained parameters, starting from an appropriate anchor state. Thus, the past trajectory of the robot is also corrected.

E. Window size, online and offline execution

The computational complexity increases with the size of $\mathbf{P}$ and the length of the time window $[t_1, t_2]$. A larger optimization is more likely to contain sufficient motion and

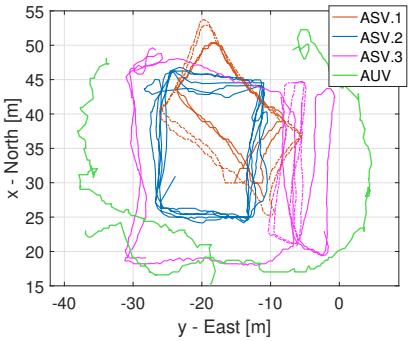


Fig. 4: A subset of the trajectories of robots in Lake Geneva from which data is used for parameter estimation. Experiments were performed with three ASVs and one AUV on several different days and two different locations near the shore of Lake Geneva.

measurements to render all parameters sufficiently observable. It results in better estimation of parameters that are likely to be static or change slowly. However, for parameters that may change quickly (e.g., local stream velocity), longer time windows result in slow or no convergence.

Given the constrained online computational budget, online estimation is performed for variable parameters with small time windows. Offline estimation is preferred for fixed parameters, which are unlikely to change during normal operation of the robot. These parameters are then saved and used as fixed values during normal operation.

F. Suboptimal or locally optimal estimates

Convexity of the optimization problem with respect to all the parameters is not guaranteed. Therefore, the obtained parameters may be a local optimum. However, if they result in a lower inertial positioning error in the short term, they are acceptable. A solution causing large errors will be updated in the subsequent optimization steps.

In practice, with windows containing insufficient motion (e.g., if robot was at rest), parameters show large variation between successive optimization steps before converging eventually. To avoid this, parameter bounds need to be chosen carefully, for example, corresponding to physical limitations. Further, we use the standard deviation between the last $n = 3$ parameter estimates as a measure of non-convergence.

IV. Experiments

Experiments as well as data gathering missions were performed in Lake Geneva with three ASVs and an AUV. The ASVs are identical but are in varying states of wear after use. In particular, the brushed motors are prone to wear and rusting. They also have slightly different mechanical attachments for external peripherals such as acoustic transceivers. Robot trajectories were pre-defined and specified as a sequence of waypoints. They were programmed to travel towards the next waypoint until the last waypoint is visited. A waypoint is considered to have been visited if the robot reaches within a radius of 3 m.

Experiments were performed on days with mild-to-no wind. The GNSS positions are available at around 1 Hz, depending on the quality of the received signal. The true and estimated (inertial) positions were recorded. Errors in inertial position estimates were due to wind/water flow, unknown initial heading, heading drift and other external factors.

A. Offline parameter estimation

We first recorded control inputs, INS measurements and GNSS positions from operation of robots on the surface on several trajectories. A plot of a subset of robot trajectories is shown in Fig. 4. We used the recorded data to perform offline estimation of fixed parameters, $[k, k_x, k_{yz}]$ for each robot separately. The optimization window was set to the entire length of a combination of several trajectories, for a particular robot *i.e.*, using all the measurement nodes from several mission datasets. We found slight differences in model parameters between robots, as expected. Results in Section V-A show significant reduction in error in the inertial trajectory computed with optimal parameters.

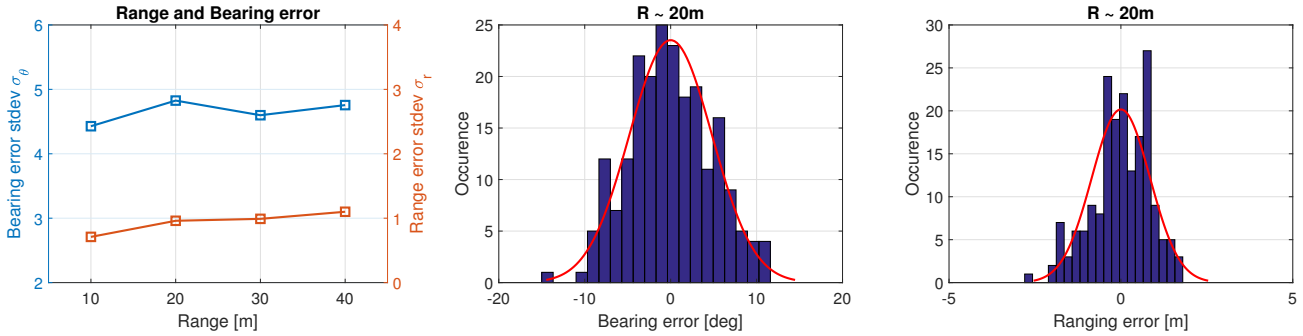
B. Online parameter estimation

By using the optimal values of fixed parameters obtained in the previous step, we ran the estimation procedure online to estimate the variable parameters, $[\theta_e, s_x, s_y]$. We were able to improve the accuracy of estimation of the robot trajectory between GNSS updates, as shown in Section V-B. To demonstrate this clearly and to partially emulate the use of acoustic positioning devices, we restricted access to GNSS information to once in 8 s, a lower update rate than that of an acoustic beacon. We show in Section V-C that we are able to calculate an updated, improved estimate of the past trajectory using the newly estimated parameters.

C. Online estimation using acoustic range and bearing

Finally, we performed online parameter estimation using acoustic range and bearing measurements. We operated the robots on the surface and recorded GNSS positions for ground truth. The results are shown in Section V-D. The acoustic positioning system consists of a base station with a USBL receiver array deployed by attaching it to a rigid, submerged mount near the shore, and a single channel transceiver mounted on each robot. The base station uses a query-response cycle to measure range (from two way time-of-flight) and bearing (from angle of arrival). It then transmits this information back to the robot. Therefore, each range and bearing measurement requires three transmissions in total, and takes 4.5 s, including idle time to avoid interference. With N robots served by the base station, the update interval for each robot would be $4.5N$ s.

The error characteristics of the system (after outlier rejection) are shown in Fig. 5. We found a very small change in accuracy with increasing distance with this system. Note, however, that the for a given error in bearing angle, the uncertainty in the calculated position increases with increasing range. For example, with increasing distance from the beacon, the uncertainty in the measured position increases along the tangential direction, while that in the radial direction remains largely the same. This error characterization is used to set the measurement uncertainty parameters in the



(a) Std. dev. of error in bearing and range. (b) Bearing error dist. for range $= 20$ m. (c) Range error dist. for range $= 20$ m.

Fig. 5: (a): Characteristics of error in range and bearing measurements for various distances (range). (b,c): Distribution in error in range and bearing measurements at 20 m range, with a Gaussian fit, after outlier rejection.

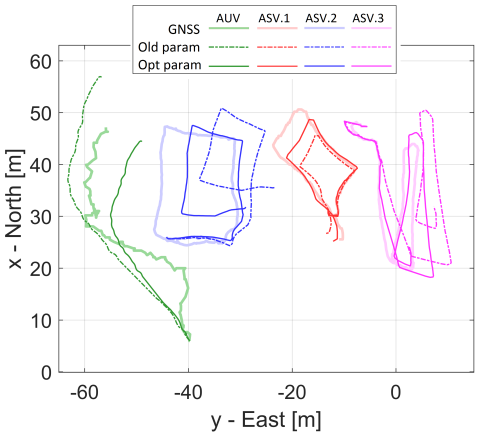


Fig. 6: Comparison of trajectories estimated with old non-optimal as well as optimal parameters. The trajectory estimation as well as parameter optimization was done offline by using real robot data. The RMSE error in both kinds of estimated trajectories is shown in Table II. Note: The trajectories have been offset in the plot to avoid overlap for clarity.

	k'	k'_x	k'_{yz}	RMSE of estimated traj.	
ASV.Old	8.0	2.0	0.3	Old param	Opt. param
ASV.1	8.1	1.71	0.155	7.62 m	3.57 m
ASV.2	10.33	1.85	0.15	3.04 m	1.66 m
ASV.3	9.26	1.82	0.14	5.72 m	1.96 m
AUV	4.42	0.55	0.07	12.95 m	6.53 m

TABLE II: Left: Values of estimated parameters for different robots. The three ASVs have slightly different parameters due to different attachments and wear over time. The old, pre-optimization parameters are also shown. Right: RMS error of inertial trajectory estimated with old parameters as well as newly obtained optimal parameters.

corresponding EKF measurement model, and accounted for by the error metric used for minimization.

V. Results

A. Offline estimation

Table II shows the combination of fixed parameters for each robot obtained after offline estimation. To demonstrate the effectiveness of our approach, we tested the new parameters on a separate testing dataset. We recomputed the inertial trajectory of the robot without using any GNSS measurements (except for initial positions) with old as well as new parameters. Fig. 6 shows a comparison of the estimated

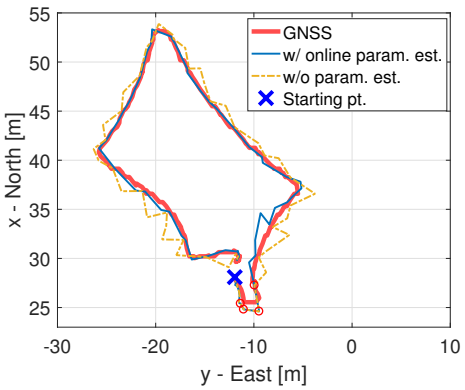


Fig. 7: Online estimation of variable parameters. The trajectory estimates with and without parameter optimization are identical initially, shown by red circles. After a few optimization steps, optimal values of variable parameters result in a more accurate trajectory estimate.

trajectories. The RMS error in the trajectories computed with old and optimal parameters is shown in Table II. Optimal parameters result in a significant reduction in error.

B. Online estimation

Fig. 7 shows a plot of the estimated trajectory with and without online estimation of variable parameters. GNSS updates were restricted to once in 8 s (but were continually recorded for ground truth). We deliberately chose a trajectory with a high initial heading error ($\sim 30°$). The optimization window was set to include the latest three GNSS updates (roughly 24 s long). The optimization was performed after each new GNSS update (roughly every 8 s). The error in the estimate is nearly identical initially, both with and without online parameter estimation. However, after the first three optimization steps, optimal parameters result in a significantly better trajectory estimate. The initial heading error does not get corrected without parameter estimation.

C. Past trajectory correction

To demonstrate this, we delayed the optimization process until after a few GNSS updates. At this point, we performed the optimization and used the newly estimated variable parameters ($[\theta_e, s_x, s_y]$) to trace the recorded measurement history and recompute the past trajectory. We used the same scenario as the previous section – GNSS updates once in 8 s,

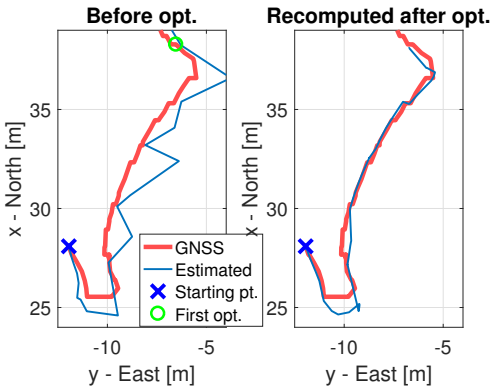


Fig. 8: Correction of past trajectory after an optimization step. Left: The section of the trajectory where the first online parameter estimation is performed at the point marked by the green circle. Right: Past trajectory recomputed with the newly estimated optimal parameters.

optimization window of 24 s. Fig. 8 shows the section of the trajectory after which the first optimization is performed, and the newly corrected past trajectory.

D. Parameter estimation with acoustic positioning

Trajectories estimated with the acoustic positioning system are shown in Fig. 9. GNSS was available to the robot only before launch, to provide the initial position. In the rest of the experiment, only acoustic range and bearing measurements were used for position updates, and GNSS was recorded for ground truth. Parameter estimation was performed after every acoustic measurement, and the window size for estimation was set to 15 s (encompassing three last range and bearing measurements). The estimated trajectory with parameter optimization has better accuracy, mainly as a result of correct estimation of heading error. Note that in Fig. 9b, the heading drift increases significantly in the inertial trajectory towards the end. This is estimated by the optimization framework, as shown by the increasing error in the estimated heading error in Fig. 9c, resulting in better inertial estimation in between acoustic updates.

E. Evolution of trajectory error

The error in estimated position (distance between true and estimated positions) increases in between external position updates, when the robot is performing inertial estimation. Minimizing this error will allow for longer intervals between position updates, while still maintaining the error below a specified level. Fig. 10 shows the evolution of instantaneous error in the estimated position for different GNSS update intervals. This data corresponds to the trajectory in Section V-B, which had a high initial heading error. While GNSS was continuously available, we restricted access to it to once in 4, 8 or 12 s, and recomputed the estimated trajectory with and without parameter estimation. The window size for parameter estimation is set to include the last three GNSS updates, *i.e.* $3\times$ the update interval.

With online parameter estimation enabled, once the parameter values converge, the estimated trajectory shows significantly lower estimation error. Results in Fig. 10 show that for a specified level of acceptable error, optimal parameter values can tolerate longer intervals between absolute position updates. This is important in the context of acoustic positioning systems, which have long update intervals especially when shared by several robots.

F. Computational overhead

We used gradient descent to solve the problem in Eq. (15) and obtain optimal parameter estimates. The most expensive component of this operation is the evaluation of the error $E_{[t_1,t_2]}(\mathbf{P})$ defined in Eq. (14). It involves computation of the EKF process and measurement updates corresponding to the time window $[t_1, t_2]$. For a time window of 15 s, this evaluation takes about 10 ms. In practice, online estimation of the three variable parameters $[\theta_e, s_x, s_y]$ typically requires 30-50 evaluations of the error before convergence. We have used a 15 s time window with an acoustic update interval of 4 to 5 s. This amounts to about 300-500 ms of time spent on the computation. Since the computation is performed every 5 s, this delay is affordable.

Our system consists of a microcontroller which handles low-level control in real-time, and a companion computer, which performs the parameter estimation. The companion computer is a Raspberry Pi Zero with a single core 1 GHz processor and 512 MB of RAM, running C++ code compiled with `-O2` optimization flag. For experiments of usual length of about 15-20 minutes, the RAM is sufficient to store the measurement and control history.

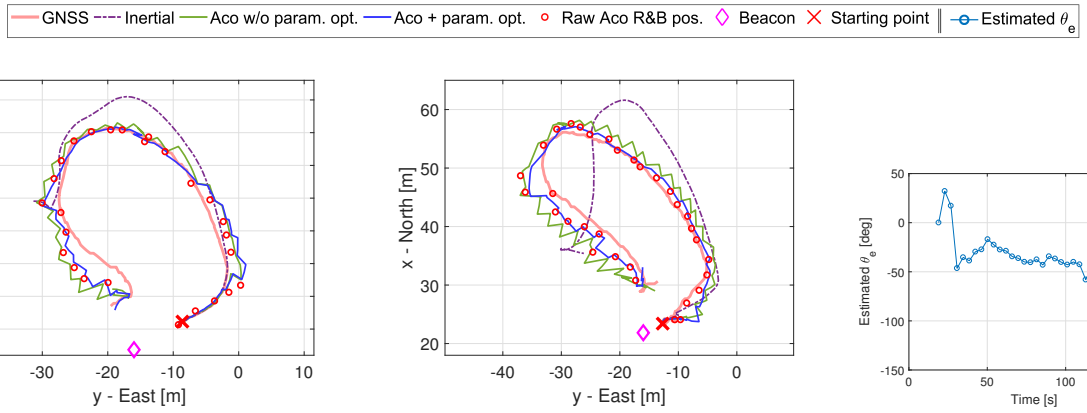
VI. Conclusion

In this paper, we have presented an approach for estimating parameters for aquatic robots, using only GNSS or acoustic absolute positioning information. It can run both online during operation, or offline using recorded data. We performed estimation of fixed parameters offline for several robots. The parameter values differ for identical robots due to differing levels of wear after use and different on-board peripherals.

We evaluated our approach with real-world experiments using GNSS as well as acoustic range and bearing measurements. We showed that optimal parameter estimates result in improved inertial localization accuracy. We also showed that by estimating variable parameters online, we can tolerate longer intervals between position measurements for a given level of acceptable position error. The presented method can also correct the past trajectory online after a parameter optimization step. This is useful in experiments where improved estimates of past trajectory can enhance performance, such as adaptive sampling.

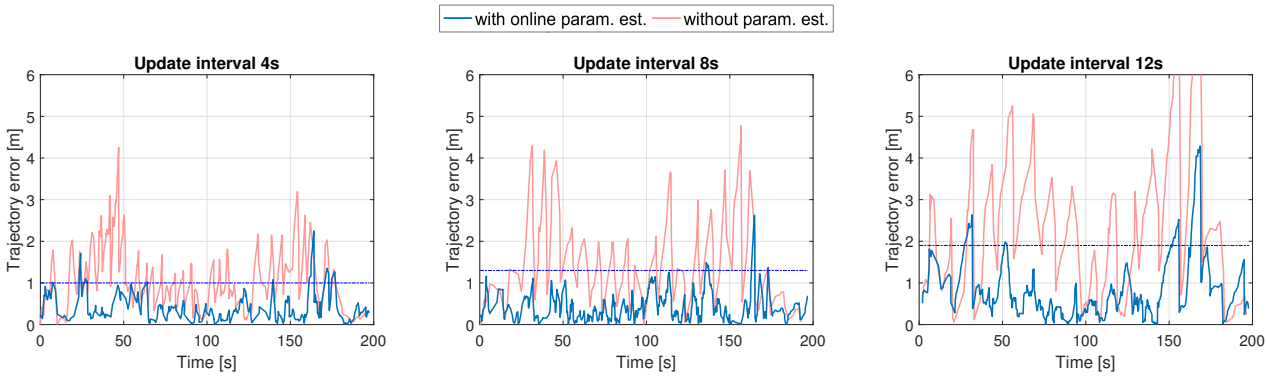
In the context of aquatic robotics, communication and external position references are sparse resources. Our approach addresses this challenge by exploiting computation, which has become inexpensive in both price and power consumption in the recent years.

We used the Kalman update error as the residual to be minimized for theoretical reasons. In future, we will evaluate other formulations of the minimization problem, in particular using Maximum Likelihood. Further, we have not considered



(a) Trajectory 1 (b) Trajectory 2 (c) Heading error θ_e in Trajectory 2

Fig. 9: (a,b) Two trajectories estimated with absolute positioning only from acoustic range and bearing measurements. GNSS data was recorded for ground truth. Estimated trajectories with and without parameter optimization are shown. (c) Estimation of heading error parameter, θ_e, in the trajectory in (b). The inertial trajectory shows increasing heading drift towards the end in (b), which is correctly estimated.



(a) GNSS interval 4s (b) GNSS interval 8s (c) GNSS interval 12s

Fig. 10: Evolution of error in position estimates for various GNSS position update intervals. Note that the error drops each time there is an external position update. Optimal parameters result in lower error in estimated position, with 95 % of the samples below the level indicated by the blue horizontal line.

the relative positions between the GNSS receiver, IMU and acoustic receiver, assuming them to coincide with the center of mass. Given that the localization errors in aquatic robots are of the order of few meters, this assumption is warranted. However, accounting for these geometric relationships can potentially result in further improvement.

References

[1] R. Wang, M. Veloso, and S. Seshan, "Active sensing data collection with autonomous mobile robots," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 2583–2588.

[2] D. E. Soltero, M. Schwager, and D. Rus, "Decentralized path planning for coverage tasks using gradient descent adaptive control," *The International Journal of Robotics Research*, vol. 33, no. 3, pp. 401–425, 2014.

[3] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1271–1287, 2014.

[4] F. S. Schill, A. Bahr, and A. Martinoli, "Vertex: A New Distributed Underwater Robotic Platform for Environmental Monitoring," in *International Symposium on Distributed Autonomous Robotic Systems, 2016*, vol. 6. Springer Proceedings in Advanced Robotics, 2018, pp. 679–693.

[5] M. Khaghani and J. Skaloud, "Autonomous Vehicle Dynamic Model-Based Navigation for Small UAVs," *J. Inst. Navigation*, vol. 63, no. 3, pp. 345–358, 2016.

[6] S. Kemna, J. G. Rogers, C. Nieto-Granda, S. Young, and G. S. Sukhatme, "Multi-robot coordination through dynamic Voronoi partitioning for informative adaptive sampling in communication-constrained environments," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 2124–2130.

[7] J. Kelly and G. S. Sukhatme, "Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration," *The International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, 2011.

[8] M. Li, H. Yu, X. Zheng, and A. I. Mourikis, "High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 409–416.

[9] Z. Taylor and J. Nieto, "Motion-based calibration of multimodal sensor arrays," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 4843–4850.

[10] T. Schneider, M. Li, C. Cadena, J. Nieto, and R. Siegwart, "Observability-Aware Self-Calibration of Visual and Inertial Sensors for Ego-Motion Estimation," *IEEE Sensors Journal*, vol. 19, no. 10, pp. 3846–3860, 2019.

[11] V. Wüest, V. Kumar, and G. Loianno, "Online Estimation of Geometric and Inertia Parameters for Multirotor Aerial Vehicles," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1884–1890.

[12] J. Svacha, J. Paulos, G. Loianno, and V. Kumar, "IMU-Based Inertia Estimation for a Quadrotor Using Newton-Euler Dynamics," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3861–3867, 2020.

[13] M. Dhaybi and N. Daher, "Accurate real-time estimation of the inertia tensor of package delivery quadrotors*," in *2020 American Control Conference (ACC)*, 2020, pp. 1520–1525.

[14] M. Burri, M. Bloesch, Z. Taylor, R. Siegwart, and J. Nieto, "A framework for maximum likelihood parameter identification applied on MAVs," *Journal of Field Robotics*, vol. 35, no. 1, pp. 5–22, 2018.

[15] "DiveNET Commander Ultra-Short Base Line (USBL) Positioning and Navigation," https://www.divenetgps.com/commander.