System Identification and Adaptive Input Estimation on the Jaiabot Micro Autonomous Underwater Vehicle

Ioannis Faros[1†] and Herbert G. Tanner[1†]

[1]Mechanical Engineering Department, University of Delaware, 130 Academy St, Newark, 19716, DE, USA.

Contributing authors: ifaros@udel.edu; btanner@udel.edu;
[†]These authors contributed equally to this work.

Abstract

This paper reports an attempt to model the system dynamics and estimate both the unknown internal control input and the state of a recently developed marine autonomous vehicle, the Jaiabot. Although the Jaiabot has shown promise in many applications, process and sensor noise necessitates state estimation and noise filtering. In this work, we present the first surge and heading linear dynamical model for Jaiabots derived from real data collected during field testing. An adaptive input estimation algorithm is implemented to accurately estimate the control input and hence the state. For validation, this approach is compared to the classical Kalman filter, highlighting its advantages in handling unknown control inputs.

Keywords: Input estimation, System identification, Underwater Vehicles

1 Introduction

The effects of climate change are disrupting marine life and contribute to a decline in water quality especially around coastal areas (Liu et al., 2016; Zereik et al., 2018). The ability for on-demand environmental observation with in situ measurements is thus becoming increasingly important. Traditional water quality observation methods typically involve measurement from stationary platforms or through manual in-situ data collection, both of which methods are costly, time-consuming and often yield sparse data sets.

These challenges can in part be overcome with automation and the utilization of one or multiple autonomous underwater vehicles (AUVs) (Lim et al., 2023; Peng et al., 2021; Li and Du, 2021). Some AUVs can even be deployed from shore,

Fig. 1: The first generation of the Jaiabot AUV.

with additional advantages in terms of affordability and open source access. One such AUV is the Jaiabot (Fig. 1). This vehicle has a slender, torpedo-like body and is approximately one meter long, with a hull diameter of 7 cm, weighs about 3 kg, and is capable of achieving speeds of nearly 5 m/s. The Jaiabot is equipped with a single propeller and rudder, and features GPS, and compass, inertial measurement unit (IMU),

in addition to potentially application-specific sensor payloads, which can monitor at a minimum salinity and temperature.

Few attempts at constructing dynamical models for such micro-AUVs have been reported. The first known dynamical system for a Jaiabot is known for its vertical dive motion (Tanner et al., 2024), based on which a control for diving is blackesigned and developed with new safety constraints, specifically with respect to overshoot in terms of assigned depth. Subsequent design modifications made on this AUV require new system parameter identification, while the aforementioned approach applies still.

In this paper the Jaiabot is operated mainly as an autonomous surface vehicle (ASV), and the focus is on filtering the data from its navigation sensors (GPS and IMU, specifically) in order to provide more accurate position and orientation estimation. In this context, the Kalman filter and its variants are well-established techniques for estimating unmeasured states by taking advantage of knowledge about the system dynamics, its input, as well the process and sensor noise. However, while the PID control loop that regulates thruster speed and rudder configuration of this ASV is known, the actual thrust generated to propel the ASV or the torque generated to make it turn is unknown. This fact implies that the control input for whatever dynamical model considered is also unknown and as a result the application of a Kalman filter is problematic.

Still, mismatches between commanded and realized control input are not uncommon in robotic deployments. Sometimes this situation arises due the lack of model fidelity, actuator limitations, or external disturbances, and can significantly degrade the performance of state estimation algorithms. The particular challenge this paper addresses for the case of the Jaiabot is the lack of control information needed for the implementation of a Kalman filter on the surface maneuvering dynamics. Specifically, here both the state of the system and its input have to be estimated simultaneously.

This is not a new problem. Numerous methods have been used for state estimation in noisy environments (Paull et al., 2013). However, when the input signal is deterministic but unknown, the need for obtaining unbiased state estimates arises. Solutions to this problem include unbiased Kalman filters, unknown input observers, and sliding-mode observers (Kitanidis, 1987; Darouach and Zasadzinski, 1997; Veluvolu and Soh, 2009).

An alternative input estimation approach considers the unknown input as the output of an auxiliary system with known dynamics, perturbed by white noise. This estimated input can then be incorporated into the state estimator to yield more accurate state estimates. Along these lines, a new technique known as retrospective cost input estimation (RCIE) has been developed (Ansari and Bernstein, 2018; Sanjeevini and Bernstein, 2022). RCIE formulates a *retrospective cost optimization* problem, where the coefficients of the input estimator are recursively adjusted to minimize a (retrospective) cost function. By doing so, RCIE effectively builds an internal model of the unknown input that estimates the later, which is subsequently fed into the Kalman filter.

The estimator coefficients are continuously adapted using the innovations (differences between predicted and observed measurements) as the error metric, thereby enhancing the accuracy and robustness of state estimation. RCIE has been extensively studied and modified for nonminimum-phase discrete time systems (Ansari and Bernstein, 2019), linear time varying (Sanjeevini and Bernstein, 2024) and invariant systems (Rahman et al., 2016). It has also been applied in the area of signal processing, especially in the context of signal numerical differentiation (Verma et al., 2024) and integration (Sanjeevini and Bernstein, 2023). RCIE has also shown promise in target tracking (Verma and Bernstein, 2025b), (Han et al., 2017) and trajectory prediction (Verma and Bernstein, 2025a).

The *contributions* of the work reported in this paper are in simulation and validation domain and are outlined as follows:

- *Modeling surge and heading dynamics identification*: Simplified surge and heading dynamics for the Jaibot are identified and validated directly from experimental data. In addition, a state augmentation is introduced in order to track temporal differences between functions of sensor measurements.
- *Adaptive input estimation (AIE):* Adaptive input estimation based on RCIE is applied to marine vehicles to estimate both system states and internal control inputs.

- *Field validation*: The RCIE implementation on the Jaiabot AUV is validated using field data.

The rest of the paper is organized as follows. Section 2 frames the process of modeling the surge and heading dynamics from real data. Section 3 provides the simulation results using real data in support of the derived models, the implemented AIE algorithm, and give a comparative analysis between AIE and the classical Kalman filter. Section 4 summarizes the research outcomes and outlines directions for future work.

2 Technical Approach

2.1 System Identification

The Jaiabot is a miniature marine vehicle that can move fast on the surface of the water. To perform principled control design for automated maneuvering for this vehicle we constructed models based on experimental data. There can be several general model templates for surface vehicle kinematics and dynamics (Fossen, 2021), and most advanced such models incorporate coupling between longitudinal (surge) and lateral (yaw) vehicle motions (Panagou et al., 2015). Given the choice of parameterizing the model based on *experimental data*, we opted here to ignore the coupling between surge and yaw (see also (Fossen, 2021)) and built separate models for each motion direction.

2.1.1 Surge dynamics

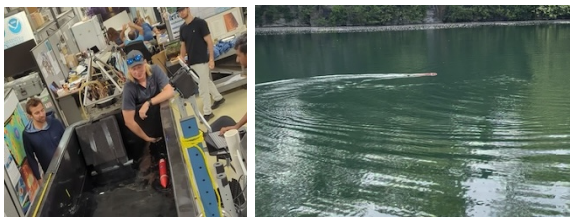
The surge dynamics of the vehicle are assumed (Fossen, 2021) in the form of a second order linear system

$$m\,\ddot{x} + d\,\dot{x} = u \tag{1}$$

where m denotes the mass of the vehicle, d is in the role of a hydrodynamic drag coefficient along the surge direction, and u expresses forward thrust. Equivalently, (1) can be regarded as a first order system in surge speed; either way, the step response of this model is known analytically and thus knowing the mass of the vehicle, the coefficient d can be directly determined through a least squares process from experimental data (Fig. 2a).

To identify the model parameters, we consider a parameterized first order transfer function from input to surge of the form $\frac{S(s)}{U(s)} = \frac{K}{s+T}$ and we measure the speed of the vehicle as it moves along the elongated water tank of Fig. 2a when given a thrust actuator reference rate of 0.5. The speed is estimated by measuring distance traveled within a second at regular time intervals.

To assess the robustness of the identified model, we performed experimental trials in which the input amplitude is varied around this nominal value and confirmed that the form of transient response curve remained the same.



(a) Surge tests (b) Yaw tests

Fig. 2: Maneuvering experiments for motion dynamics system identification. (a) Experimental data for surge dynamics were obtained from indoor tests in a long water tank. (b) Data for yaw dynamics were collected during outdoor tests in lake Allure, PA.

The limited length of the water tank, and given the speed of the vehicle at the particular thrust rate input command, allows for only few (four) data points (Fig. 3). The nonlinear model least-squares fit to the analytical first order step response yields parameter values for (1) as follows: $m = 0.469$, and $d = 0.311$.

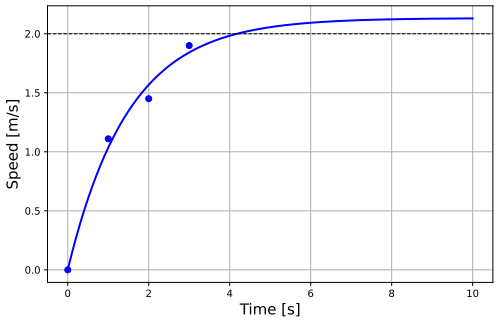


Fig. 3: Parameter fit for surge model parameter identification.

2.1.2 Heading dynamics

The identification of the heading (yaw) dynamics of the Jaiabot is performed using a standard maneuvering test for marine vehicles (Fossen, 2021). The vehicle moves with a fixed rudder configuration over a period of time (Fig. 2b), and the response of the yaw rate is measured (in this case, based on the vehicle's IMU and GPS readings). Assuming a similar second order model for the yaw dynamics

$$I\,\ddot{\theta} + c\,\dot{\theta} = r \tag{2}$$

where I is in the role of the vehicle's moment of inertia along the vertical axis, c is a hydrodynamic drag coefficient and r expresses the rudder input command, a least squares fitting approach on the ramp response for this model gives $I = 4.896$ and $c = 9.087$. The parameter identification process follows the same principles as that for surge, only now we have the luxury of a larger data set to which we can fit the ramp response of (2) to a rudder reference of 1500 (Fig. 4).

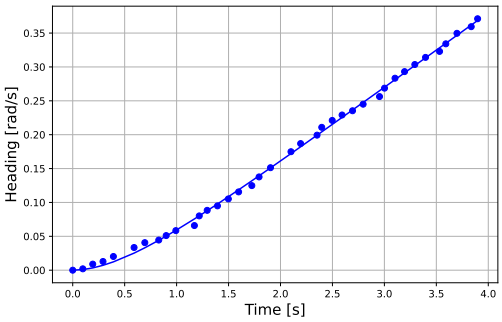


Fig. 4: Parameter fit for heading model parameter identification.

In the section that follows, the surge and yaw dynamics, (1) and (2) respectively, are expressed in discrete time, in order to be integrated into a Kalman filter formulation.

2.2 Discrete-time Vehicle Dynamics

As expected of any marine surface vehicle, the Jaiabot's motion is rarely a pure translation or a pure rotation. Thus none of the models of (1) or (2) can, *in isolation*, describe the motion of the vehicle. What is more, to be integrated into a Kalman filter, these models need to be associated with measurable outputs. And while the vehicle's orientation is directly measured via its IMU or compass, individual GPS measurements do not directly inform about the length of the path traveled, which is practically what is being tracked by (1). To overcome the latter challenge we construct discrete-time models that describe the evolution of path length traversed by the vehicle and its change in orientation between consecutive *time steps*.

2.2.1 Heading

We start with the incremental heading dynamics because they are more straightforward. If we denote h_1 the vehicle's current heading and h_2 its yaw rate, then the discrete-time heading dynamics derived by analytic integration of (2) for a time step T[1] results in

$$\begin{bmatrix} h_1 \\ h_2 \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0.5487(1-\mathrm{e}^{-1.82249T}) \\ 0 & \mathrm{e}^{-1.82249T} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}_k + \begin{bmatrix} 0.5487+0.301072(\mathrm{e}^{-1.82249T}) \\ 0.5487(1-\mathrm{e}^{-1.82249T}) \end{bmatrix} r_k.$$

We now augment the heading state vector at step $k+1$ to include the heading at step k as so

$$\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0.5487(1-\mathrm{e}^{-1.82249T}) & 0 \\ 0 & \mathrm{e}^{-1.82249T} & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}_k + \begin{bmatrix} 0.5487+0.301072(\mathrm{e}^{-1.82249T}) \\ 0.5487(1-\mathrm{e}^{-1.82249T}) \\ 0 \end{bmatrix} r_k. \tag{3}$$

The purpose of augmenting the system is to allow the discrete-time linear model to capture the state's temporal difference over time –specifically, the difference between heading at time $k+1$ - heading at time k. This change, denoted as $\Delta\bar{\theta}$ is defined as:

$$\Delta\bar{\theta}_{k+1} = 0.200474 \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}_{k+1} \tag{4}$$

This equation shows that the output of the system is essentially the the difference of the heading states h_1 and h_3, and corresponds to the actual compass heading difference between two timesteps. Thus, it can be directly obtained from sensor measurements.

[1] In implementation, T is set to 0.546 seconds.

2.2.2 Surge

The discrete-time augmented dynamics for surge is constructed along the same lines. We denote s_1 the length of the path traveled by the vehicle, s_2 its speed along this path, and s_3 the path length at the previous time step.

Just as in the preceding section, the reason we augment the system's dynamics is to capture the temporal difference of the system's state, specifically, path length at time $k+1$ minus path length at time k. This difference now can be expressed in the model's output and linked directly with the sensor's measurements. Note that, in this case of surge dynamics, this temporal difference is more critical, because GPS latitude /longitude readings do not directly reflect the distance traveled, and without this manipulation, a direct association of system's outputs to sensor measurements is not straightforward. With these we arrive at

$$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 1.50625(1-\mathrm{e}^{-0.66397T}) & 0 \\ 0 & \mathrm{e}^{-0.66397T} & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}_k + \begin{bmatrix} 1.50625T+2.26879(\mathrm{e}^{-0.66397T}) \\ 1.50625(1-\mathrm{e}^{-0.66397T}) \\ 0 \end{bmatrix} u_k \quad (5)$$

Similarly, the output for this discrete-time system is defined as the difference ΔS of path lengths between consecutive steps

$$\Delta S_{k+1} = 1.4162 \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}_{k+1} \quad (6)$$

The challenge here, however, is that ΔS is not provided as a sensor measurement. Nonetheless, this quantity can be derived directly through geometric conditions that depend explicitly on compass and GPS measurements. Without excessive loss, therefore, we approximate the incremental path length, ΔS, assumed at the scale of the time step to be adequately captured by a circular arc with zero order hold (ZOH) over r_k, with the distance between the vehicle at consecutive steps, $\Delta\bar{S}$, and express the latter as

$$\Delta S_{k+1} \approx \Delta\bar{S}_{k+1} = \sqrt{\Delta S_x{}^2 + \Delta S_y{}^2} \quad (7)$$

where ΔS_x and ΔS_y can be computed from GPS measurement differences after projection to the body frame at time step $k+1$ (see Fig. 5).

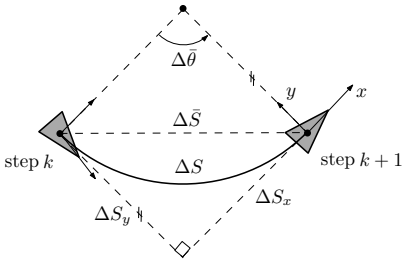


Fig. 5: Geometric relation between surge model output and sensor measurements.

A remaining challenge, however, is that during deployment and due to the vehicle's control architecture and interface, neither the input thrust u in (1) nor the input moment r in (2) are directly known. Hence, traditional filtering methods, such as Kalman filtering would not work properly for state estimation, because the rely on access to the true executed control inputs.

This is where retrospective cost input estimation comes in to provide real-time estimates of u_k and r_k through an adaptive estimation process.

2.3 Adaptive Input Estimation

This section provides a very brief mathematical overview of the AIE technique according to Verma et al. (2024). Consider the linear discrete-time system

$$x_{k+1} = A\,x_k + B u_k$$
$$y_k = C\,x_k + V$$

where $x_k \in \mathbb{R}^n$ is the system state, $u_k \in \mathbb{R}$ is the control input (assumed unknown), and $V \in \mathbb{R}$ is zero-mean Gaussian sensor noise. Matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times 1}$, and $C \in \mathbb{R}^{1\times n}$ are known. The goal of AIE is to estimate simultaneously both u_k and x_k. AIE is practically comprised of an input estimation subsystem, and a Kalman filter (Thacker and Lacey, 1998).

2.3.1 Input estimation subsystem

To obtain the estimated input $\hat{u}_k$, we construct the input estimation subsystem of order $n_e > 1$ as follows

$$\hat{u}_k = \sum_{i=1}^{n_e} M_{i,k}\,\hat{u}_{k-i} + \sum_{i=1}^{n_e} N_{i,k}\,z_{k-i} \quad (8)$$

where the $M_{i,k} \in \mathbb{R}$, $N_{i,k} \in \mathbb{R}$, and z_k is the residual in the prediction step of Kalman filter. The *input estimate* in subsystem (8) can be written in form of

$$\hat{u}_k = \Phi_k \, \theta_k \tag{9}$$

where the *regressor matrix* Φ_k is defined as

$$\Phi_k \triangleq [\hat{u}_{k-1} \cdots \hat{u}_{k-n_e} \; z_k \cdots z_{k-n_e}] \in \mathbb{R}^{1 \times l_\theta} \tag{10}$$

and the *coefficient vector* θ_k is

$$\theta_k \triangleq [M_{1,k} \cdots M_{n_e,k} \; N_{0,k} \cdots N_{n_e,k}] \in \mathbb{R}^{l_\theta} \tag{11}$$

with $l_\theta \triangleq 2n_e+1$. The order of this subsystem, n_e, must be chosen large enough to properly develop the internal model for the estimation of the input. The objective now becomes to update the coefficient vector θ_k, in order to derive an estimate $\hat{u}_k$. To do so, we first define the backward-shift operator for a discrete signal Y_k

$$\boldsymbol{q}^{-1} \, Y(k) \triangleq Y(k-1)$$

and we this definition we express the following filtered signals

$$\begin{aligned} \Phi_{f,k} &\triangleq G_{f,k}(\boldsymbol{q}^{-1}) \, \Phi(k-i) \\ \hat{u}_{f,k} &\triangleq G_{f,k}(\boldsymbol{q}^{-1}) \, \hat{u}(k-i) \end{aligned}$$

By defining now $G_{f,k}(\mathbf{q}^{-1}) \triangleq \sum_{i=1}^{n_f} \mathbf{q}^{-i} H_i(k)$, taking $n_f \geq 1$ to be the window length of filter, the aforementioned filtered signals can be written as

$$\begin{aligned} \Phi_{f,k} &= \sum_{i=1}^{n_f} H_i(k) \, \Phi(k-i) \\ \hat{u}_{f,k} &= \sum_{i=1}^{n_f} H_i(k) \, \hat{u}(k-i) \end{aligned}$$

where now $H_i(k)$ is defined as

$$H_i(k) \triangleq \begin{cases} CB & k \geq i = 1 \\ C\bar{A}_{k-1} \cdots \bar{A}_{k-(i-1)} B & k \geq i \geq 2 \\ 0 & i > k \end{cases}$$

with $\bar{A} \triangleq A(I + K_k C)$, and K_k being the Kalman filter gain that is included in the update step of the state filtering process. To find the coefficient vector θ_k, we construct an optimization problem; more specifically, a retrospective optimization problem where the coefficient vector will denote the optimization variable. To this end, define the *retrospective variable* as

$$z_{r,k}(\hat{\theta}) \triangleq z_k - (\hat{u}_{f,k} - \Phi_{f,k}\hat{\theta})$$

where now $\hat{\theta}$ represents the decision variable. The *retrospective cost function* is defined as

$$J_k(\hat{\theta}) \triangleq (\hat{\theta}-\theta_0)^\intercal R_\theta (\hat{\theta}-\theta_0) + \sum_{i=0}^{k} R_z z_{r,i}^2(\hat{\theta}) + R_d(\Phi_i \hat{\theta})^2$$

where $R_z \in (0, \infty)$ and $R_d \in (0, \infty)$ are scalar optimization gains, and $R_\theta \in \mathbb{R}^{l_\theta \times l_\theta}$ is a positive definite gain matrix. Note that the regularization term $(\hat{\theta}-\theta_0)^\intercal R_\theta (\hat{\theta}-\theta_0)$ weighs the initial estimate and ensures that the θ_{k+1} has a unique global minimizer (Islam and Bernstein, 2019). Define now $P_0 \triangleq R_\theta^{-1}$. Then for all $k \geq 1$, the unique global minimizer θ_{k+1}, is given by the recursive least squares (RLS) update

$$\begin{aligned} P_{k+1} &= P_k - P_k \tilde{\Phi}_k^\intercal \Gamma_k \tilde{\Phi}_k P_k \\ \theta_{k+1} &= \theta_k - P_k \tilde{\Phi}_k^\intercal \Gamma_k (\tilde{z}_k + \tilde{\Phi}_k \theta_k) \end{aligned} \tag{12}$$

where

$$\Gamma_k \triangleq \big(\tilde{R}^{-1} + \tilde{\Phi}_k P_k \tilde{\Phi}_k^\intercal\big)^{-1} \qquad \tilde{\Phi}_k \triangleq \begin{bmatrix} \Phi_{f,k} \\ \Phi_k \end{bmatrix}$$

$$\tilde{z}_k \triangleq \begin{bmatrix} z_k - \hat{u}_{f,k} \\ 0 \end{bmatrix} \qquad \tilde{R} \triangleq \begin{bmatrix} R_z & 0 \\ 0 & R_d \end{bmatrix}$$

By using (12) and replacing the $k+1$ with k in (9) we derive the estimated input. We choose $\theta_0 = 0$ which implies $\hat{u}_0 = 0$.

Applying the AIE algorithm to (3)–(4) and (5)–(6) is not straightforward due to several hyperparameters that require careful manual tuning. The remarks that follow outline some of the insights we developed in this process.

Remark 1. *To properly implement the AIE algorithm, it is essential to first specify all the hyperparameters n_e, n_f, R_z, R_d, R_θ, typically done empirically through trial and error. This issue has been recognized in literature, where the introduction of variable-rate forgetting into the recursive*

least squares can potentially alleviate the tuning burden (Verma et al., 2024).

Remark 2. *In the process of tuning the hyperparameters one needs to be cognizant that the associated subsystem is highly sensitive to hyperparameter variations, and small changes in them can either yield the desired results or result in significantly high values for the estimated input.*

Remark 3. *A reasonable choice for gain matrix R_θ is in the form of $R_\theta = 10^{-\alpha}$, where α is a positive number. However, in some cases like the one in this paper a choice in the form of $R_\theta = 10^{-a/100}$, where α is a positive number, appears more likely to yield convergent θ values.*

Remark 4. *Large values of the hyperparameter n_f or n_e do not necessary lead to better filtering of the signals or better estimate control input. On the contrary, they might cause divergence of the input estimate.*

3 Validation

This section presents simulation and experimental results and numerical analysis that supports the theoretical predictions on (a) estimation of the deterministic control input as it is presented in Section 2.3, and (b) the effect of the estimated input to the Kalman filter to estimate the pose of the Jaiabot. The approach is applied to the problem of estimating the states of surge (5)–(6) and heading model (3)–(4). To achieve this, the control inputs for both the surge and heading dynamics must first be estimated, because the operator has no direct knowledge of the thrust and torque inputs.[2] For comparison reasons, a standard Kalman filter is also implemented with the assumption that the nominal operator control input is the one that is implemented by the Jaiabot.

For the estimation process, we rely on IMU and GPS sensor readings, both of which are affected by noise and bias. While well structured noise models and statistics exist in the literature for both sensors, RCIE literature offers little coverage of noise statistics. A key challenge in our setup is the lack of the ground truth for IMU and GPS readings during field deployments, as no reference system was available to directly validate the accuracy of position or attitude estimates.

Part of the implementation of the algorithm for estimation of control inputs and the states of the two systems, is the process of tuning all the hyperparameters. In our tests, we set the following values. For the *surge dynamics*: $n_e = 4$, $n_f = 8$, $R_z = 1$, $R_d = 50$, $R_\theta = 10^{-0.01} I_9$. For the *heading dynamics*: $n_e = 3$, $n_f = 4$, $R_z = 1$, $R_d = 0.1$, $R_\theta = 10^{-2} I_7$.

The algorithms used experimental data collected in Lake Allure, PA, where the Jaiabot was deployed in a series of turning maneuvers (see Fig. 13). By assuming zero mean noise statistics we can estimate the noise covariance in IMU and GPS data that is $V_{\text{IMU}} = 4172$ (rad/s)^2 and $V_{\text{GPS}} = 2.23$ m^2, respectively.

3.1 Results for Surge Dynamics

The AIE algorithm was first implemented for the surge dynamics. Figure 6 gives a comprehensive view of the results obtained. The coefficient vector θ_k in (11) converges after approximately 100 iterations (Fig. 6, top). Nine coefficients ($l_\theta = 2n_e + 1$) were included to achieve convergence for this subsystem. Figure 6 (middle) illustrates the estimated input for the surge dynamics. The dashed red line shows the average of the values that the input takes. Note that the control input fluctuates within the $(-0.05, 0.30)$ range, an observation that also aligns with the behavior of the estimated state, shown in Fig. 6 (bottom). Recall that the output (and measurement) for the surge dynamics represents the length of the chord of the arc along which the Jaiabot is moving. Hence, small values in this context indicate small steps along the motion path.

Figure 7 gives a visual comparison of the state estimation results. The red dashed curve represents measurements, the blue curve shows the state estimate obtained from the Kalman filter that uses a nominal (step) control input set at $u_k = 1$, and the black curve illustrates the output of AIE. Arguably, AIE provides significantly more effective noise filtering compared to the standalone Kalman filter.

Due to the lack of ground truth for the GPS sensor, we perform a simulation-based validation study for the surge dynamics using a synthetic

[2] The relationship between operator commands and thrust/-torque inputs can alternatively be empirically established via hydrodynamic experiments.

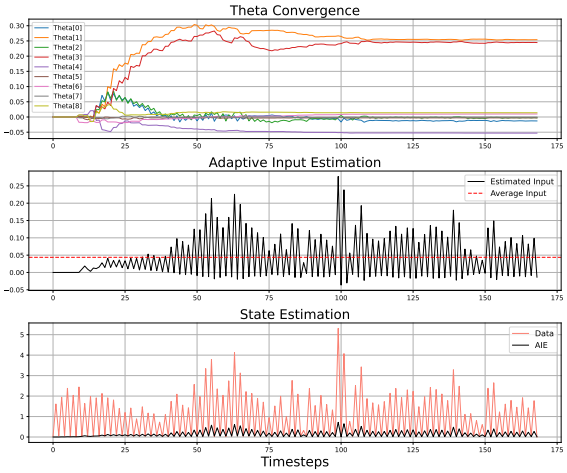


Fig. 6: Adaptive input estimation results for surge.

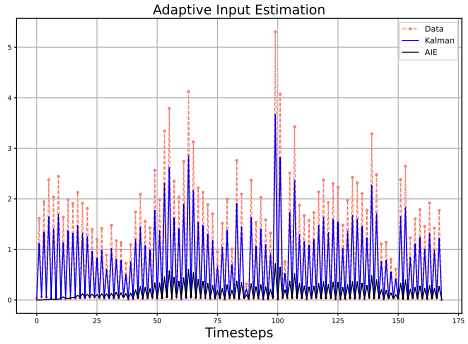


Fig. 7: Comparison of the estimated state for surge against experimental data, for the Kalman filter and AIE.

dataset constructed by augmenting the original experimental data with structured noise and known sinusoidal patterns. This analysis provides additional evidence that AIE achieves better performance than the Kalman filter regarding (i) the trajectory tracking RMSE (Fig. 8, top) and (ii) the reconstruction of a smoother signal of the state (Fig. 8, middle). The convergence of the coefficient vector θ_k is achieved after some retuning of the hyperparameters (Fig. 8, bottom).

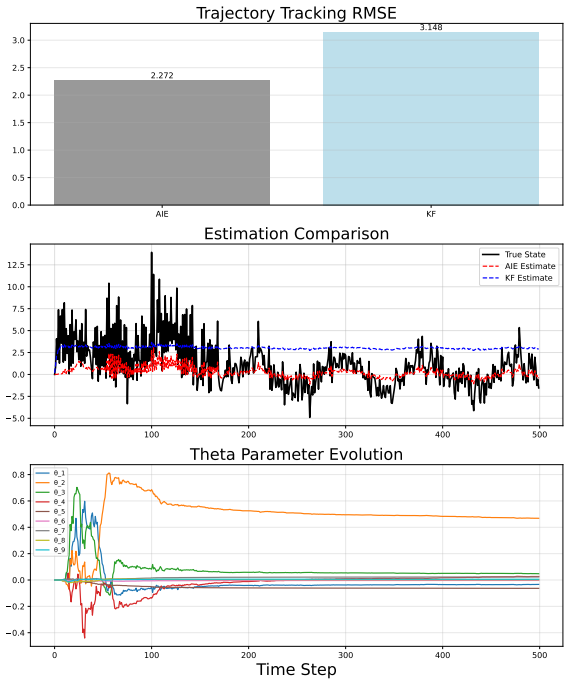


Fig. 8: Simulation-based validation results for surge.

3.1.1 Validation on decoupled surge dynamics

A separate data set, which allows the independent validation of the implementation of RCIE on the surge dynamics in isolation, is now utilized. This data set corresponds to an experiment involving a Jaiabot moving straight (in open loop) for 15 seconds with the same thruster input command as that featured in Fig. 3. Figure 9 illustrates the results of AIE applied on this dataset. For this case, we selected a coefficient vector θ_k with shorter length and retuned the hyperparameters, resulting in convergence of the coefficient vector θ_k and effective filtering of the noise during state estimation.

3.2 Results for Heading Dynamics

In this subsection we report on the AIE implementation for the heading dynamics. Figure 10 summarizes the findings. Note that for this model the measurement is the heading (angle) which is obtained directly from the AUV sensors. Similarly to the AIE implementation for the surge dynamics, the coefficient vector θ_k in (11) converges after

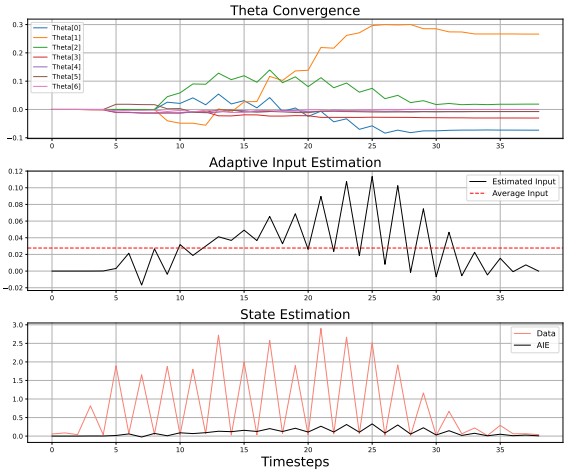


Fig. 9: Validation of adaptive input estimation results for decoupled surge motion.

approximately 100 iterations (Fig. 10, top). For heading estimation, only 7 coefficients are needed to achieve proper convergence. Figure 10 (middle) shows the estimated input for the heading dynamics. The dashed red curve represents the average of the values that the input takes over the whole period of application. In this execution, it can be observed that the control input fluctuates within a $(-100, 170)$ range, in agreement with the behavior of the estimated state, shown in Fig. 10 (bottom) which depicts the estimated state. As seen, during instances of significant fluctuations (red curve), such as within the range $(80, 110)$, the initial noisy behavior is effectively filtered (black curve).

Figure 11 offers a comparative analysis of the state estimation results. In Fig. 11, the red dashed line represent system measurements, while the blue lines show the estimated states obtained from the Kalman filter, with the nominal control input set to $u_k = 1$. The black line shows the estimated state from the AIE algorithm. It can be observed that while the Kalman filter provides a good estimate for the heading, it struggles to properly filter out significant fluctuations without accurate knowledge of the system input. On the other hand, AIE demonstrates improved performance in scenarios involving large fluctuations, providing more robust and accurate estimates compared to the standalone Kalman filter.

Application of AIE suggests that the DC gain of the surge dynamics (5) is 4.68 while that for the heading dynamics (3) is 0.125.

The absence of ground truth for IMU measurements motivates a synthetic data-based analysis for the heading dynamics similar to that presented for surge. The synthetic dataset is based on the original experimental data by introducing controlled sinusoidal patterns and additive noise. The simulation analysis of the synthetic data set reveals that the AIE method outperforms the standalone Kalman filter. Figure 12 illustrates the RMSE values relative to the true and the estimated signal, the reconstruction of a smoother state trajectory, and the convergence of the coefficient vector θ_k after hyperparameter tuning.

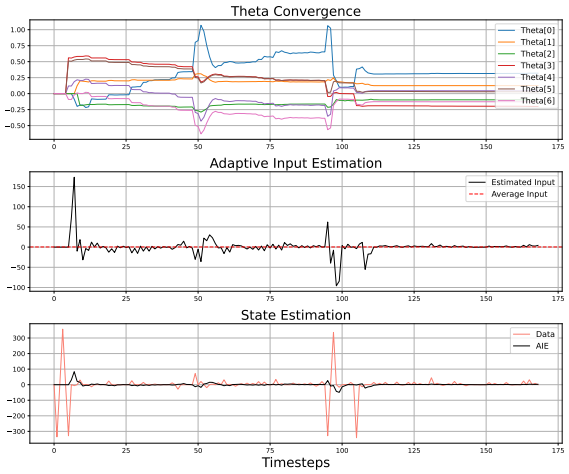


Fig. 10: Adaptive input estimation results for heading.

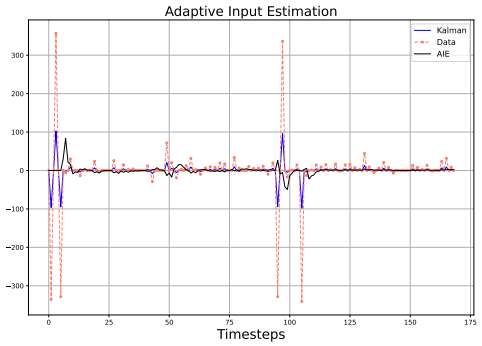


Fig. 11: Comparison of the estimated state for heading against experimental data, for the Kalman filter and AIE.

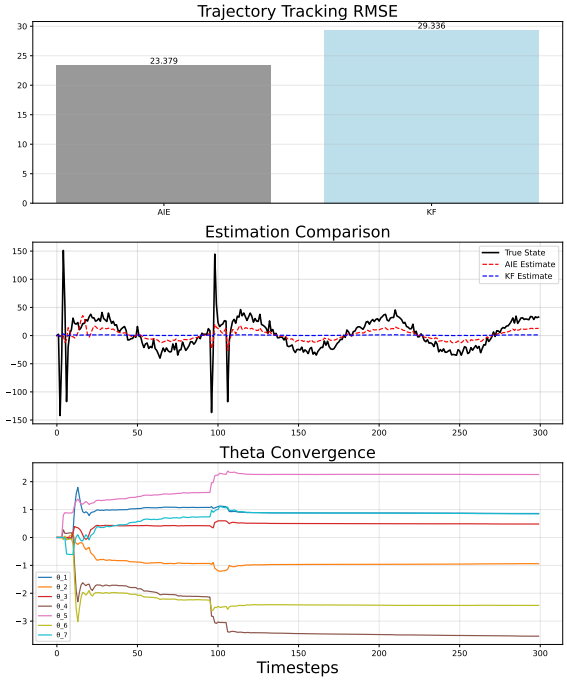


Fig. 12: Simulation-based validation results for heading.

3.3 Estimate-based Trajectory Reconstruction

This section makes a combined spatial comparative analysis between AIE and standalone Kalman filter, as the ASV motion path is reconstructed based on the corresponding estimates and compared to experimental (GPS) data, the same one that was used for the surge dynamics in the AIE process. In both cases (AIE and standalone Kalman filter) the trajectory is reconstructed as

$$X_{k+1} = X_k + \Delta S \sin\left(\frac{\Delta\theta}{2}\right)$$
$$Y_{k+1} = Y_k + \Delta S \cos\left(\frac{\Delta\theta}{2}\right)$$

where X_k and Y_k are the north-south and east-west coordinates of the ASV at time step k. Here, $\Delta\theta$ is derived from the heading dynamics, either using the AIE or the Kalman process, while the ΔS is estimated similarly. Figure 13 depicts the results of this reconstruction for the ASV trajectory. It can be observed that AIE outperforms the Kalman filter since it is adjusting the control input to be suitable for estimation of the state. On the other hand, Kalman filter presents big fluctuations along the trajectory, primarily due to the control input that was selected from the user, which does not accurately reflect the actual internal control input implemented by the ASV.

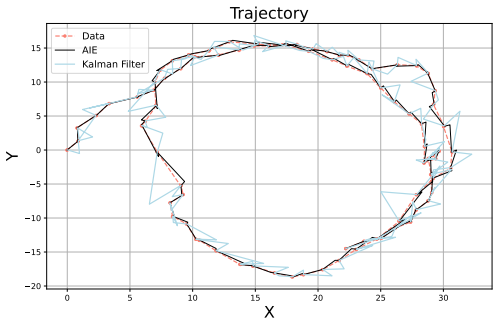


Fig. 13: Comparison of the trajectory reconstruction based on the Kalman filter and the AIE relative to experimental data.

Following up on the analysis of Section 3.1.1 that utilized a "surge-only" field deployment data set, here we reconstruct the motion path of the ASV based on the AIE and the standalone Kalman filter (using a preselected control input) algorithms, and compare the two. Figure 14 depicts this comparison. The figure indicates that the AIE gives better state estimates as it adjusts the unknown control input and reconstructs the trajectory without the abrupt fluctuations that the Kalman filter yields.

4 Conclusion

Just as an appropriate choice of the control input in a system is critical for effective control of a dynamical system, so is knowledge of the actual control input applied for accurate state estimation. In real-world application scenarios, however, there can be a mismatch between the commanded control inputs and the ones actually realized. Because of this mismatch, both estimation and control performance suffer.

Consequently, in cases where the control input is unknown, it becomes necessary to estimate

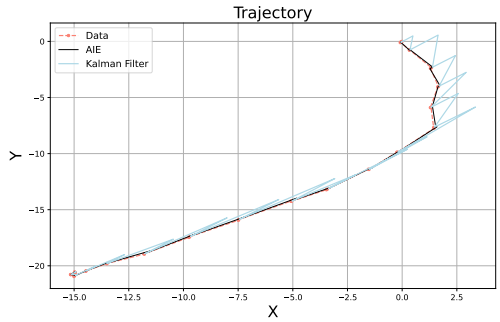


Fig. 14: Adaptive input estimation results for surge.

it *in parallel* to the state of the system. This paper implements this concept for the first time on an ASV to generate filtered pose (position & orientation) estimates using linear models for its surge and heading dynamics, identified independently based on experimental data. Results indicate promise for AIE as a method for providing accurate and consistent real-time state and input estimates for an ASV, which are especially useful in cases where the actual control inputs are opaque to its operator.

Acknowledgements. The authors thank Chanaka Bandara for collecting and sharing the ASV data at Lake Allure PA. Funding for this work is provided in part through the USGS Next Generation Water Observing System (NGWOS) Research and Development program via award #G23AC00149-00 and by NSF through award #2234869.

References

Ansari, A., Bernstein, D.S.: Input estimation for nonminimum-phase systems with application to acceleration estimation for a maneuvering vehicle. IEEE Transactions on Control Systems Technology **27**(4), 1596–1607 (2018)

Ansari, A., Bernstein, D.S.: Input estimation for nonminimum-phase systems with application to acceleration estimation for a maneuvering vehicle. IEEE Transactions on Control Systems Technology **27**(4), 1596–1607 (2019)

Darouach, M., Zasadzinski, M.: Unbiased minimum variance estimation for systems with unknown exogenous inputs. Automatica **33**(4), 717–719 (1997)

Fossen, T.I.: Handbook of Marine Craft Hydrodynamics and Motion Control, 2nd edn. John Wiley & Sons, Chichester (2021)

Han, L., Ren, Z., Bernstein, D.S.: Maneuvering target tracking using retrospective-cost input estimation. IEEE Transactions on Aerospace and Electronic Systems **52**(5), 2495–2503 (2017)

Islam, S.A.U., Bernstein, D.S.: Recursive least squares for real-time implementation [lecture notes]. IEEE Control Systems Magazine **39**(3), 82–85 (2019)

Kitanidis, P.K.: Unbiased minimum-variance linear state estimation. Automatica **23**(6), 775–778 (1987)

Li, D., Du, L.: AUV trajectory tracking models and control strategies: A review. Journal of Marine Science and Engineering **9**(9), 1020 (2021)

Lim, H.S., Filisetti, A., Marouchos, A., Khosoussi, K., Lawrance, N.: Applied research directions of autonomous marine systems for environmental monitoring. OCEANS 2023-Limerick, 1–10 (2023)

Liu, Z., Zhang, Y., Yu, X., Yuan, C.: Unmanned surface vehicles: An overview of developments and challenges. Annual Reviews in Control **41**, 71–93 (2016)

Paull, L., Saeedi, S., Seto, M., Li, H.: AUV navigation and localization: A review. IEEE Journal of Oceanic Engineering **39**(1), 131–149 (2013)

Panagou, D., Tanner, H.G., Kyriakopoulos, K.J.: Nonholonomic control design via reference vector fields and output regulation. ASME Journal of Dynamic Systems, Measurement and Control **137**(8), 081011–0810119 (2015)

Peng, Z., Wang, J., Wang, D., Han, Q.-L.: An overview of recent advances in coordinated control of multiple autonomous surface vehicles. IEEE Transactions on Industrial Informatics

17(2), 732–745 (2021)

Rahman, Y., Xie, A., Hoagg, J.B., Bernstein, D.S.: A tutorial and overview of retrospective cost adaptive control. In: Proceedings of the 2016 IEEE American Control Conference, pp. 3386–3409 (2016)

Sanjeevini, S., Bernstein, D.S.: Decomposition of the retrospective performance variable in adaptive input estimation. In: Proceedings of the of the 2022 IEEE American Control Conference, pp. 242–247 (2022)

Sanjeevini, S., Bernstein, D.S.: Random-walk elimination in numerical integration of sensor data using adaptive input estimation. In: Proceedings of the 2023 IEEE American Control Conference, pp. 4791–4795 (2023)

Sanjeevini, S., Bernstein, D.S.: Performance-variable decomposition in retrospective cost adaptive control of linear time-varying systems. IEEE Systems and Control Letters **185**, 105744 (2024)

Tanner, H.G., Bandara, C.T., Gyves, M.C.: Partial system identification and sensor fusion with the jaiabot micro autonomous underwater vehicle. In: Proceedings of the 2024 IEEE Mediterranean Conference on Control and Automation, pp. 167–172 (2024)

Thacker, N., Lacey, A.: Tutorial: The kalman filter. Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester **61** (1998)

Verma, S., Bernstein, D.: Prediction-based target tracking using adaptive input and state estimation for real-time numerical differentiation. In: AIAA SCITECH 2025 Forum, p. 0348 (2025)

Verma, S., Bernstein, D.S.: Target tracking using the invariant extended kalman filter with numerical differentiation for estimating curvature and torsion. arXiv preprint arXiv:2501.04262 (2025)

Veluvolu, K.C., Soh, Y.C.: High-gain observers with sliding mode for state and unknown input estimations. IEEE Transactions on Industrial Electronics **56**(9), 3386–3393 (2009)

Verma, S., Sanjeevini, S., Sumer, E.D., Bernstein, D.S.: Real-time numerical differentiation of sampled data using adaptive input and state estimation. International Journal of Control **97**, 2962–2974 (2024)

Zereik, E., Bibuli, M., Mišković, N., Ridao, P., Pascoal, A.: Challenges and future trends in marine robotics. Annual Reviews in Control **46**, 350–368 (2018)