Tracking individual fish from a moving platform using a split-beam transducer

Nils Olav Handegard,[a] Ruben Patel, and Vidar Hjellvik
Institute of Marine Research, P.O. Box 1870 Nordnes, N-5817 Bergen, Norway

(Received 16 May 2003; revised 11 July 2005; accepted 11 July 2005)

Tracking of individual fish targets using a split-beam echosounder is a common method for investigating fish behavior. When mounted on a floating platform like a ship or a buoy, the transducer movement often complicates the process. This paper presents a framework for tracking single targets from such a platform. A filter based on the correlated fish movements between pings is developed to estimate the platform movement, and an extended Kalman filter is used to combine the split-beam measurements and the platform-position estimates. Different methods for gating and data association are implemented and tested with respect to data-association errors, using manually tracked data from a free-floating buoy as a reference. The data association was improved by utilizing the estimated velocity for each track to predict the location of the next observation. The data association was more robust when estimates of platform tilt/roll were used. Other techniques to estimate position and velocity, like linear regression and smoothing splines, were implemented and tested on a simulated data set. The platform-state estimation improved the estimates for methods like the Kalman filter and a smoothing spline with cross validation, but not for robust methods like linear regression and smoothing spline with a fixed degree of smoothing. © *2005 Acoustical Society of America.* [DOI: 10.1121/1.2011410]

PACS number(s): 43.30.Sf [KGF] Pages: 2210–2223

I. INTRODUCTION

The split-beam echosounder has become an important tool for investigating fish behavior. It provides four-dimensional observations (range, two angles, and time), making it possible to track objects over a period, thus obtaining information about fish movements in the acoustic beam.[1,2] The technique has several applications, including studies of migratory behavior, vessel-avoidance effects, and counting fish. Tracking objects with a split-beam echosounder is fairly easy for well-defined targets observed by a fixed transducer, but is more difficult if the targets are weak and/or the transducer platform is unstable.

The concept of the "single echo" often arises in fisheries acoustics. This is the echo formed by one fish in isolation. The split-beam echosounder measures the range and the angular direction to the target, and an algorithm for single-echo detection (SED) is used to detect these echoes, discarding any that have overlapping contributions from more than one fish. Different single-target algorithms have been tested, and it was found that algorithms based on the phase stability (measured by the standard deviation of the phase angle) rejected multiple targets most efficiently, and a method utilizing amplitude differences between the split-beam transducer elements performed well for strong targets.[3]

After the single targets are detected they must be combined into tracks. This can be achieved using an algorithm for multiple-target tracking (MTT). This is an automatic procedure that can handle several tracks simultaneously. MTT is used in several applications, and the literature is extensive.[4,5]

Any motion of the transducer complicates the tracking of fish. Data association is the process of combining the single echoes into useful tracks. Transducer motion makes this more difficult and causes error in the velocity estimates. In this paper, we investigate various methods of data association and velocity estimation under these conditions, leading to improved techniques for fish tracking.

II. MATERIALS

Observations made from a free-floating buoy[6] are used to test the performance of the tracker. The buoy contains a Simrad EK60 split-beam echosounder with an ES38-12 split-beam transducer of approximately 11.5° symmetrical beam width between the −3 dB directions. The transducer is mounted on a cable with floats and weights to stabilize it during operations.[7] The resulting depth of the transducer is 40 m. A compass is mounted on the transducer house to determine its alongship direction.

The test data set was recorded during a fish avoidance study in the Barents Sea in March 2001. The buoy was passed several times by a trawling vessel, and the purpose of the study was to investigate the behavior of the fish (mainly cod) in response to the vessel and gear.[7] Data from one such passing comprise the test set whose total duration is 10 min. The ping rate is 1 s^{-1} and the data include 5700 single-echo detections (see the echogram shown in Fig. 1).

The EK60 Mark 1 software (Ver. 1.3.0.54) was used to detect single targets within the echo beam. The settings for the SED algorithm are given in Table I.

[a]Electronic mail: nilsolav@imr.no

 0001-4966/2005/118(4)/2210/14/$22.50

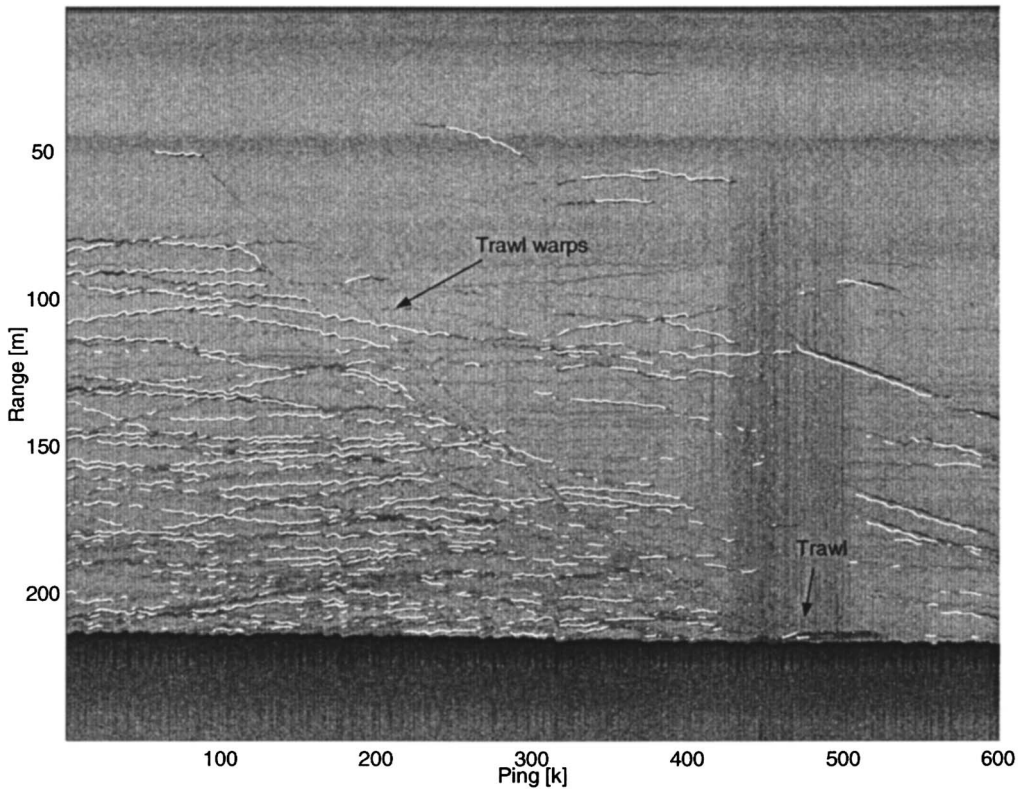

FIG. 1. Tracked buoy data. White lines indicate the tracks. The gray-scaled background is the volume backscattering values. In the middle of the echogram the trawl and the warps are seen. Vertical lines above the trawl are noise from the trawl, trawl sensors, and trawl doors.

III. METHODS

Our method of tracking single fish observed from a moving platform is divided into three subtasks: initial tracking, platform-state estimation, and final tracking. The initial tracking and the final tracking use the same MTT algorithm. The initial tracking gives a first association of measurements to tracks, from which we subsequently estimate the platform state (i.e., the position and direction of the acoustic axis). In the final tracking, the platform state is used to estimate the geo-relative position and velocity for each track at each time step. This set of positions and velocities is called the track state $\mathbf{x}_k$ which is obtained using several alternative methods. The extended Kalman filter (EKF) is only one method for estimating the track state, but since predictions of track state are easily obtained from the EKF, the EKF is presented together with the MTT in Sec. III A. The alternative track-state estimators require previously associated data points and are presented in Sec. III B. The technique for platform-state estimation is described in Sec. III C.

TABLE I. EK60 single echo detection (SED) settings according to the Simrad EK60 Scientific echosounder manual, p. 141. The echo lengths are given as a factor τ multiplying the pulse length (or pulse duration time).

Description	Value
Minimum Echo Length	0.8τ (m)
Maximum Echo Length	1.8τ (m)
Maximum Phase Deviation	8.0 (phase steps)
Maximum Gain Compensation	6.0 (dB)

Methods for measuring data-association errors based on manually associated data, and track-state-estimation errors based on simulated data, are presented. These are described in Secs. III D and III E, respectively. In addition, the sensitivities to the various tracking parameters are quantified for both measures.

A. Multiple target tracking (MTT) using extended Kalman filtering (EKF)

The MTT process associates measurements to tracks and estimates the track state based on the single-echo detections and the transducer platform state (measured or estimated). The MTT consists of several elements: track-state estimation, track-state prediction, gating, data association, and track support. The track-state estimation gives the location and velocity for each track indicated by the current and previous measurements. Based on this information, we predict where the next measurement is likely to be. The prediction is then used in the gating and in the data association. Gating is the process of removing unlikely track/observation pairs, and data association is the actual pairing of predictions and observations into tracks. The last element of the MTT is the track-support algorithms. These take care of initiating, terminating, and validating tracks.

1. The extended Kalman filter and track-state estimation and prediction

One method for estimating the track state is the extended Kalman filter (EKF). The advantage of the EKF is that the

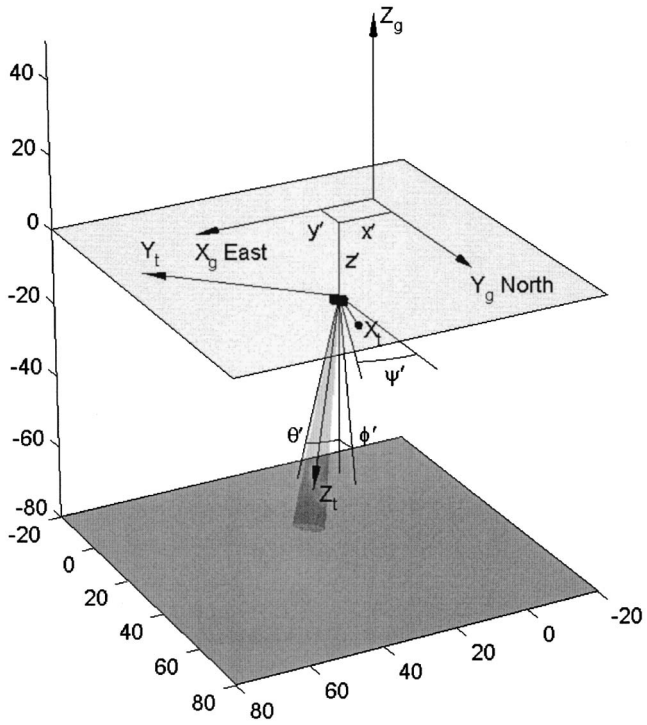

FIG. 2. The platform state shown in the geographical coordinate system, e_{gc}. The eastward, northward, and vertical directions are indicated with X_g, Y_g, and Z_g, respectively. The platform state $\mathbf{z}_k=\hat{\mathbf{z}}_k+\mathbf{v}_{y,k}=[x' \quad y' \quad z' \quad \theta' \quad \phi' \quad \psi']^T_{e_{gc}}$ is indicated in the figure. The alongship, athwartship, and vertical transducer axes are given by X_t, Y_t, and Z_t, respectively. The compass reading ψ' is the angle off Y_g for the projection of X_t onto the horizontal plane.

state is estimated by weighting the previous state with the new observation, thus it is unnecessary to recalculate the whole track. See Sec. III C for a description of alternative track-state estimators. The true position and velocity for target i at ping k and time t_k are given by the track-state vector

$$\mathbf{x}_k=[x_k \quad y_k \quad z_k \quad \dot{x}_k \quad \dot{y}_k \quad \dot{z}_k \quad \mathrm{TS}_k]^T_{e_{gc}},$$

where (x_k,y_k,z_k) is the position vector, $(\dot{x}_k,\dot{y}_k,\dot{z}_k)$ is the velocity vector, TS_k is the target strength, T denotes matrix transposition, and e_{gc} denotes the geo-referenced coordinate system (see Fig. 2). The corresponding estimate is denoted by $\hat{\mathbf{x}}$. In general there are several tracks i and measurements j for each ping k, but for easier readability we have here omitted i and j from the notation. Under a constant-velocity assumption, the track state at time $t_{k+1}=t_k+\Delta T_k$ is given by

$$\mathbf{x}_{k+1}=\Phi(\Delta T_k)\mathbf{x}_k+\mathbf{w}_\Phi, \tag{1}$$

where $\Phi(\Delta T_k)$ is the matrix of the track-state transition for the time interval ΔT_k and $\mathbf{w}_\Phi=\mathbf{w}(t_k)$ is an additive system error component (see Appendix A for details). The error is assumed to be normally distributed and the target-model-covariance matrix is defined as $\Sigma_\Phi=E(\mathbf{w}_\Phi\mathbf{w}_\Phi^T)$ (see Appendix A).

From the estimated track state at time step $k-1$, we can predict the state at step k. The predicted track state $\tilde{\mathbf{x}}$ is found by setting the model error $\mathbf{w}_\Phi=\mathbf{0}$ in Eq. (1),

$$\tilde{\mathbf{x}}_k=\Phi(\Delta T_k)\hat{\mathbf{x}}_{k-1}. \tag{2}$$

We have also implemented a "zero-velocity" prediction. This is achieved by setting $\Delta T_k=0$ in Eq. (2), which yields $\tilde{\mathbf{x}}_k=\hat{\mathbf{x}}_{k-1}$ since $\Phi(0)=I$. Note that this prediction is used in the data association algorithm only, and not in the Kalman-filter estimates (see below).

A split-beam echosounder with a SED algorithm calculates the alongship angle α_k, the athwart ship angle β_k, the range r_k, and the target strength TS_k, for each target at every time step k. These measurements are represented by

$$\mathbf{y}_k=[\alpha_k \quad \beta_k \quad r_k \quad \mathrm{TS}_k]^T_{e_{tp}}+\mathbf{v}_{y,k}.$$

Here, $\mathbf{v}_{y,k}$ is an additive error component that depends on the accuracy of the transducer, and e_{tp} denotes the transducer coordinate system.

The platform state $\mathbf{z}_k$ describes the acoustic axis of the transducer, which may move in space and point in any direction. It is given by

$$\mathbf{z}_k=\hat{\mathbf{z}}_k+\mathbf{v}_{z,k}=[\hat{x}'_k \quad \hat{y}'_k \quad \hat{z}'_k \quad \hat{\theta}'_k \quad \hat{\phi}'_k \quad \hat{\psi}'_k]^T+\mathbf{v}_{z,k}, \tag{3}$$

where $\hat{\mathbf{z}}_k$ is the estimated platform state, $(\hat{x}'_k,\hat{y}'_k,\hat{z}'_k)_{e_{gc}}$ is the transducer position at time t_k, $\hat{\psi}'_k$ is the corresponding transducer compass reading, $(\hat{\theta}'_k,\hat{\phi}'_k)$ are the tilt angles in east-west and north-south directions (see Fig. 2), and $\mathbf{v}_{z,k}$ is an additive error component. The errors are combined in $\mathbf{v}_{R,k}=\begin{bmatrix}\mathbf{v}_{z,k}\\ \mathbf{v}_{y,k}\end{bmatrix}$ and are assumed to be additive and normal with covariance $\Sigma_{R,k}=E(\mathbf{v}_{R,k}\mathbf{v}_{R,k}^T)$ (see Appendix B). In our case, all the off-diagonal elements of $\Sigma_{R,k}$ are assumed to be zero. The relationship between the track state $\mathbf{x}_k$, the platform state $\hat{\mathbf{z}}_k$, and the measurement $\mathbf{y}_k$ is defined by

$$\mathbf{y}_k=\mathbf{h}(\mathbf{x}_k,\hat{\mathbf{z}}_k,\mathbf{v}_{R,k}), \tag{4}$$

where $\mathbf{h}$ is a nonlinear function defined in Appendix B. Since the relationship between the track state and the measurement is nonlinear, an EKF must be used instead of an ordinary Kalman filter. The EKF uses a linear approximation of Eq. (4) about the predicted track state, $\tilde{\mathbf{x}}_k$, the estimated platform state, $\hat{\mathbf{z}}_k$, and the measurement-error vector set to zero, $\mathbf{v}_{R,k}=\mathbf{0}$. The linear relationship between the measurement and the track state is

$$\mathbf{y}_k\sim\mathbf{h}(\tilde{\mathbf{x}}_k,\hat{\mathbf{z}}_k,\mathbf{0})+H(\mathbf{x}_k-\tilde{\mathbf{x}}_k)+V\mathbf{v}_{R,k},$$

where

$$H_{[l,m,k]}=\frac{\partial h_{[l]}}{\partial x_{[m]}}(\tilde{\mathbf{x}}_k,\hat{\mathbf{z}}_k,\mathbf{0})$$

and

$$V_{[l,m,k]}=\frac{\partial h_{[l]}}{\partial v_{[m]}}(\tilde{\mathbf{x}}_k,\hat{\mathbf{z}}_k,\mathbf{0}).$$

The predicted measurement is found from the predicted track state by setting the measurement error $\mathbf{v}_{R,k}=\mathbf{0}$ in Eq. (4),

$$\tilde{\mathbf{y}}_k=\mathbf{h}(\tilde{\mathbf{x}}_k,\hat{\mathbf{z}}_k,\mathbf{0}). \tag{5}$$

The error covariance of the predicted estimate is given by

$$\tilde{\Sigma}_{\mathbf{x},k} = \Phi(\Delta T_k)\hat{\Sigma}_{\mathbf{x},k-1}\Phi(\Delta T_k)^T + \Sigma_\Phi,$$

where $\hat{\Sigma}_{\mathbf{x},k-1}$ is the estimated track-state-error covariance from the previous time step and Σ_Φ is the track-state-transition covariance. Initially $\hat{\Sigma}_{\mathbf{x},k-1} = \Sigma_{\mathbf{x},0}$ where

$$\Sigma_{\mathbf{x},0} = [P^2_{0\ xy}, P^2_{0\ xy}, P^2_{0\ z}, P^2_{0\ dxdy}, P^2_{0\ dxdy}, P^2_{0\ dz}, P^2_{0\ TS}]I. \quad (6)$$

The elements of $\Sigma_{\mathbf{x},0}$ are parameters. The estimated track state is found by weighting the predicted track state and the observation by the Kalman gain, K_k,

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + K_k(\mathbf{y}_k - \mathbf{h}(\tilde{\mathbf{x}}_k, \hat{\mathbf{z}}_k, 0)),$$

where

$$K_k = \tilde{\Sigma}_{\mathbf{x},k} H_k^T (H_k \tilde{\Sigma}_{\mathbf{x},k} H_k^T + V_k \Sigma_{R,k} V_k^T)^{-1}.$$

The estimated track-state-error covariance is

$$\hat{\Sigma}_{\mathbf{x},k} = (I - K_k H_k)\tilde{\Sigma}_{\mathbf{x},k},$$

the linearized innovation, $\hat{\boldsymbol{\epsilon}}_k = \mathbf{y}_k - \tilde{\mathbf{y}}_k$, is

$$\hat{\boldsymbol{\epsilon}}_k = H(\mathbf{x}_k - \tilde{\mathbf{x}}_k) + V\mathbf{v}_{R,k},$$

and the linearized covariance of the innovation is

$$\Sigma_{\epsilon,k} = H_k \tilde{\Sigma}_k H_k^T + V_k \Sigma_{R,k} V_k^T.$$

2. Gating and data association

The next step in the MTT algorithm is to associate the observations with existing predictions. The challenge is to avoid associating a false prediction-observation pair, and to avoid *not* associating a true prediction-observation pair (track splitting). The innovation $\hat{\boldsymbol{\epsilon}}_{ijk}$ is the difference between the predicted measurement from track i and the observation j at ping k; it is used in the gating and data-association algorithms. Innovations based on both constant and "zero-velocity" predictions are implemented and tested. Gating is the initial step in the data association, where unlikely pairs of predictions and observations are removed. The distance indicated by the innovation is calculated, and any pair separated by more than a set amount is deemed to be outside the gate. The gate can be specified in several ways. We have implemented two types. Firstly, a static gate accepts pairs within a fixed ellipsoidal volume around the prediction:

$$d^2_{ijk} = \hat{\boldsymbol{\epsilon}}_{ijk} G \hat{\boldsymbol{\epsilon}}^T_{ijk} < 1, \quad (7)$$

where

$$G = \begin{bmatrix} \alpha_G^2 & 0 & 0 \\ 0 & \beta_G^2 & 0 \\ 0 & 0 & r_G^2 \end{bmatrix}^{-1}. \quad (8)$$

Here α_G, β_G, and r_G act as gates in angles and range, respectively. Note that different gate widths may be used in the initial and final tracking. These are indexed by numbers, i.e., r_{G1} and r_{G2}. If no measurement is associated with a given track at time step k, the algorithm continues the search at time step $k+1$. However, the volume likely to contain the next measurement of the track is now larger, and it will increase for each time step when no association is made. This suggests the use of a dynamic gate whose size increases with the prediction covariance. Such a gate requires *a priori* knowledge of the detection probabilities and measurement errors.[8] In order to test the concept of a dynamic gate, suppose the detection probabilities are constant, so that

$$d^2_{ijk} = \hat{\boldsymbol{\epsilon}}_{ijk}\, \Sigma^{-1}_{\boldsymbol{\epsilon},ijk}\, \hat{\boldsymbol{\epsilon}}^T_{ijk} \leqslant 2 \ln\left[\frac{c_G}{\sqrt{|\Sigma_{\boldsymbol{\epsilon},ijk}|}}\right] + \ln[|\Sigma_{\boldsymbol{\epsilon},ijk}|], \quad (9)$$

where c_G is a constant. The last term is a penalizing term to prevent tracks with high innovation covariance (many missing pings) being preferred over those with low innovation covariance.[8]

When gating is performed, a sparse matrix is obtained containing the d^2's for all combinations of predictions (i) and observations (j) for ping k that fall inside the gates. The association algorithm assigns observations to predictions based on the elements in this matrix. Several data-association methods are described in the literature.[8] One of the most common is the global nearest neighbor (GNN) method, which assigns observations to tracks by minimizing the total sum of distances. In this case an assignment with a higher d^2 may be chosen at a given time step if the total cost is lower. This assignment problem is solved by the Bertsekas auction algorithm.[9] We also implemented a simpler algorithm that first assigns the best pairing at each time step, then the next best, continuing until all observations inside the gates are assigned. In this case the total sum of distances may be higher than for the GNN method. We have compared the methods with respect to their data-association errors.

3. Track support

The last part of the MTT algorithm is track support. This starts, terminates, and validates the tracks. When an observation is not connected to an existing track, a new one is spawned. The new track state starts with the position indicated by the observation, the velocity set to zero, and $\Sigma_{\mathbf{x},0}$ as an assumed error covariance. A track is terminated at the last ping before the first sequence of MP consecutive missing pings. The track is rejected if it is less than ML (minimum track length), or if the number of missing pings divided by the track length is more than MN. The track length is simply the number of pings from start to finish, including any missing pings.

B. Other track-state estimators

In the previous section, the EKF was used both for prediction and estimation of the track state. Since the EKF estimates the new track state based on the new associated measurements and the present track prediction only, this technique is well suited to be integrated with the gating and data-association algorithms. However, after the measurements have been associated, other techniques can be used to estimate the position and velocity of the targets. The additional techniques we have tested are the Kalman-smoothing algorithm, linear regression, and a smoothing spline.

After estimating the tracks with the Kalman filter and computing the track-support functions, a Kalman smoothing algorithm may be applied. The advantage over the Kalman-

filter algorithm is that the influence of the initial values (zero velocity and $\Sigma_{\mathbf{x},0}$) is avoided. This is achieved by using the Kalman-filter predictions and estimates to compute the Kalman smoothed estimate

$$\hat{\hat{\mathbf{x}}}_k = \hat{\mathbf{x}}_k + J_k(\hat{\hat{\mathbf{x}}}_{k+1} - \Phi(\Delta T_k)\hat{\mathbf{x}}_k),$$

with covariance

$$\hat{\hat{\Sigma}}_{\mathbf{x},k} = \hat{\Sigma}_{\mathbf{x},k} - J_k(\hat{\hat{\Sigma}}_{\mathbf{x},k+1} - \tilde{\Sigma}_{\mathbf{x},k+1})J_k^T,$$

by moving backwards through the track. Here

$$J_k = \hat{\Sigma}_{\mathbf{x},k}\Phi(\Delta t_k)^T\hat{\Sigma}_{\mathbf{x},k}^{-1}.$$

Both the Kalman filter and the Kalman smoother estimate the position and velocity errors. However, the comparison with other methods is based on the position and velocity estimates only, not the errors. For the Kalman-filter estimate we define position and velocity vectors

$$\hat{\mathbf{s}}_k^{\mathrm{KF}} = \hat{\mathbf{x}}_{k[1\cdots3]}, \quad \hat{\mathbf{u}}_k^{\mathrm{KF}} = \hat{\mathbf{x}}_{k[4\cdots6]}$$

and for the Kalman smoothed estimate

$$\hat{\mathbf{s}}_k^{\mathrm{KS}} = \hat{\hat{\mathbf{x}}}_{k[1\cdots3]}, \quad \hat{\mathbf{u}}_k^{\mathrm{KS}} = \hat{\hat{\mathbf{x}}}_{k[4\cdots6]}.$$

The linear regression and the smoothing spline methods require us to map the measurements $\mathbf{y}_k$ to Cartesian coordinates (e_{gc}). This is achieved using the estimated platform state $\hat{\mathbf{z}}$,

$$\mathbf{s}_k = \mathbf{g}(\mathbf{y},\hat{\mathbf{z}}) = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{e_{gc}},$$

where $\mathbf{g}$ is found from $\mathbf{h}$ (see Appendix B). A constant-velocity track,

$$\mathbf{s}_k^L = \mathbf{s}_0^L + \mathbf{u}^L t_k, \tag{10}$$

is then fitted to $\mathbf{s}_k$ by minimizing $SS=\Sigma_{\mathbf{k}}\|\hat{\mathbf{s}}_k^L-\mathbf{s}_k\|_2^2$, and $\hat{\mathbf{s}}_0^L$ and $\hat{\mathbf{u}}^L$ are found. However, a straight regression line through the $\mathbf{s}_k$'s may be a rather crude approximation. Therefore, we also used a smoothing spline, which minimizes a compromise between the exact fit and the smoothness of the track. This is implemented in the R function[10] smooth.spline, where the degree of smoothness can be chosen automatically by cross validation, or by setting the parameter "spar" to a given value. We have used both the nonparametric (cross validation) and parametric (spar=0.7) methods, denoted by s_k^{SNP} and s_k^{SP}, respectively. For details see the documentation of smooth.spline in the R program,[10] and the standard-S literature.[11]

C. Platform-state estimation

When tracking targets from vessels and buoys, the position and the direction of the acoustic beam may change from ping to ping. Knowledge of the transducer position and orientation is required for accurate results. There are standard notation and sign conventions for the motion of a submerged body.[12] However, the angles defined in this section are relative to the east-west and north-south directions, not the vessel heading, i.e., the measurements of ship or buoy motion have to be mapped into $\mathbf{z}_k$ before evaluating Eq. (3). If direct measurements of transducer position and orientation were available, the initial tracking and platform-state estimation would be unnecessary.

However, this is often not the case. We have therefore developed a method to extract correlated fish movements relative to the acoustic beam, based on the association from the initial tracking. This common movement is attributed to movement of the transducer platform. In our test data, the transducer compass direction and the GPS positions are measured, while the tilt, roll, and heave are unknown. The unknown platform-state variables are set to zero before initial tracking. The measurements are mapped from e_{tp} coordinates to e_{gp} coordinates (see Appendix B for details), and the mean differences in angles and range between successive pings are used to estimate the transducer movement. For the east-west angle the mean difference is computed as

$$\delta_{\theta_k} = \frac{1}{N_k}\sum_{i=1}^{N_k}(\theta_{ki} - \theta_{k+1,i}), \tag{11}$$

where k is the ping number, i is the track number, N_k is the number of tracks with measurements in both ping k and $k+1$, and θ is the east-west angle to the target relative to the transducer position. Some tracks are short and are therefore rejected by the track-support functions. However, in this procedure these short tracks are retained since the platform movement may split tracks.

If fish are swimming in one direction through the beam, consecutive δ_{θ_k} will be positive. If the transducer is moving cyclically, consecutive δ_{θ_k} will vary from positive to negative within the cycle. To extract this cyclical movement, the differences are summed cumulatively to get $\theta'_{\mathrm{RAW},k}=\theta'_{\mathrm{RAW},k-1}+\delta_{\theta_k}$, starting with $\theta_0=0$. To compensate the effect of polarized swimming, a running-mean filter is used to remove the drift component (platform translation and fish migration) from $\{\theta'_{\mathrm{RAW},k}\}$,

$$\hat{\theta}_k = \theta'_{\mathrm{RAW},k} - \frac{1}{2M+1}\sum_{l=k-M}^{k+M}\theta'_{\mathrm{RAW},l},$$

where M is here the length of the running-mean filter. We set M equal to $1/f_0$ rounded up to the nearest integer, where f_0 is a constant defining the lowest tilt/roll frequency that the algorithm can detect. To estimate the north-south and range components of the transducer movement, the same process is applied to ϕ' and z'. Then the tracking is repeated utilizing the obtained estimates of transducer tilt, roll, and heave.

D. Testing data association

To test the performance of the data association algorithms, the buoy data set is first tracked manually (true tracks), and then compared with the tracks obtained from the various algorithms (auto tracks). The auto-track identifiers are compared with the true-track identifiers. If the auto-track identifier changes along a true track, a split error has oc-

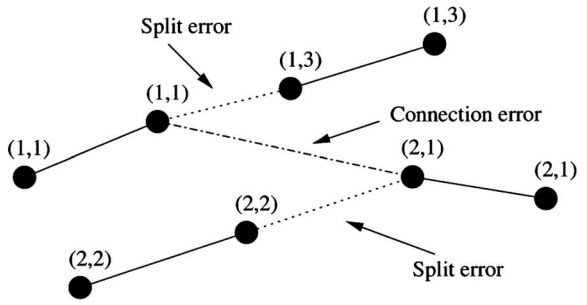

FIG. 3. The splits and false associations of a true track compared with an auto track. The pairs of numbers show (true-track number, auto-track number). The split error is found by counting the number of auto-track number changes along the true tracks (1+1) divided by possible number of changes (3+3). The connection error is the number of true-track changes along the auto tracks (1+0+0) divided by the possible number of changes (3+1+1).

curred. If the true-track identifier changes along an auto track, a connection error has occurred. The measure for track splitting is defined as

$$J_{\text{split}}(\mathbf{p}) = \frac{\Sigma_i C_i^s}{\Sigma_i (L_i - 1)}, \tag{12}$$

where $\mathbf{p}$ gives the parameter settings, C_i^s is the number of changes in the auto-track identifier along the true track i, and L_i is the length of the true track i. An example is given in Fig. 3. The measure for false associations is similarly defined as

$$J_{\text{connect}}(\mathbf{p}) = \frac{\Sigma_i C_i^c}{\Sigma_i (L_i - 1)}, \tag{13}$$

where C_i^c is the number of changes in the true-track identifier along auto track i and L_i is the length of auto track i. The two measures are combined into a single measure of the association error

$$J_{\text{alloc}} = \tfrac{1}{2}(J_{\text{split}} + J_{\text{connect}}). \tag{14}$$

To test the parameter sensitivity to the data association error, the tracker is run with one parameter perturbed by ±10% at each run. The sensitivity measure is defined as

$$S_{pa} = 0.5\left(\frac{|\Delta J_{+10\%}|}{J_{\text{alloc}}} + \frac{|\Delta J_{-10\%}|}{J_{\text{alloc}}}\right)\left(\frac{|\Delta p|}{p}\right)^{-1}, \tag{15}$$

where $\Delta J_{+10\%}$ and $\Delta J_{-10\%}$ are the changes in J_{alloc} when perturbing parameter $p \pm 10\%$, $\Delta p/p = 0.1$, except for MP, which is an integer. To test the sensitivity to MP, this parameter is increased or decreased by one. A relative measure is still used, i.e., $\Delta p/p = 1/\text{MP}_0$, where MP_0 is the unperturbed value.

E. Testing position and velocity estimates

A simulated data set is used to test the validity of the position and velocity estimates for each track. Different fish trajectories are simulated, including transducer tilt/roll effects. The track states estimated by the various techniques are then compared with the known positions and velocities from the simulations. The simulated data comprise four constant-speed tracks: two straight lines and two half circles (see Table II). Instead of simulating transducer tilt/roll/heave, we estimated the transducer platform state from the test data, $\hat{\mathbf{z}}_{k,\text{testdata}}$. These estimates are taken as the true platform state in the simulations, i.e., $\mathbf{z} = \hat{\mathbf{z}}_{k,\text{testdata}}$. This is applied only to the tilt/roll/heave; the transducer translation and compass heading are set to zero. The simulated tracks are mapped to measurement space and an error is added according to

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{z}_k, \mathbf{0}) + \mathbf{v}_{k[7\cdots10]}. \tag{16}$$

To simulate missing pings, randomly selected data points are removed from the tracks. Short tracks are simulated by termination after the desired track length (L). Distributions of the track lengths and the ratio of missing pings to the track length are calculated from the test data. To make the test more realistic, we simulated 1230 data sets based on these distributions. An overview of the simulated data sets is given in Table III.

The mean along-track errors in position and velocity,

$$\text{ME}_s = \frac{1}{L}\sum_{k=1}^{L} \|\hat{\mathbf{s}}_k^E - \mathbf{x}_{k[1\cdots3]}\|_2 \tag{17}$$

and

$$\text{ME}_u = \frac{1}{L}\sum_{k=1}^{L} \|\hat{\mathbf{u}}_k^E - \mathbf{x}_{k[4\cdots6]}\|_2, \tag{18}$$

are evaluated as measures of the fit between the true and estimated position and velocity, respectively. Here L is the track length and E denotes the estimation technique (KS, KF, L, SNP, or SP). The means ME_s and ME_u are calculated for each of the four tracks in all the simulated data sets, using all the estimation techniques. Note that for the constant-velocity tracks, $\hat{\mathbf{u}}_k$ is the same for all k's.

The sensitivity of the estimated position and velocity to the tracking parameters is tested. The manually associated data set is the reference. Thus we can compare the impact of different parameter settings on the track estimates of the described algorithms. The sensitivity measure is

TABLE II. The simulated tracks.

Track	Curve	Speed (m/s)	Direction	Depth
1	Line	0.53	NE	200
2	Line	0.37	E	150
3	Semicircle	0.589	N to S	230
4	Semicircle	0.589	N to S	170

TABLE III. The number of simulated data sets for each track length (L) and each missing ping to track length ratio (MN). There are several duplicates for each setting due to the added normally distributed error.

		L						
		15	25	35	45	55	65	75
MN	0.1	90	120	60	60	30	30	30
	0.3	120	160	80	80	40	40	40
	0.5	60	80	40	40	10	10	10

$$S_{ps} = \frac{1}{\sum_{i=1}^{M} L_i} \sum_{i}^{M} \sum_{k}^{L_i} \|\hat{\mathbf{s}}_{ik}^{E} - \hat{\mathbf{s}}_{ik}^{E_p}\|_2 \tag{19}$$

for the position estimates and

$$S_{pv} = \frac{1}{\sum_{i=1}^{M} L_i} \sum_{i}^{M} \sum_{k}^{L_i} \|\hat{\mathbf{u}}_{ik}^{E} - \hat{\mathbf{u}}_{ik}^{E_p}\|_2 \tag{20}$$

for the velocity estimates. Here M is the number of tracks, L_i is the track length for track i, $\hat{\mathbf{s}}_{ik}^{E}$ and $\hat{\mathbf{u}}_{ik}^{E}$ are the position and velocity estimates for track i at time t_k using estimation technique E and the optimal parameter setting for the data association error, and $\hat{\mathbf{s}}_{ik}^{E_p}$ and $\hat{\mathbf{u}}_{ik}^{E_p}$ are the position and velocity estimate where parameter p is perturbed. The parameters are perturbed by ±10%, one at a time. The mean S_{ps} and S_{pv} are calculated for each parameter and each estimation technique.

IV. RESULTS

The various methods for data association and track estimation are examined by comparing the tracker results obtained, respectively, with simulated data and the test data. The questions are: how do the different data association algorithms perform, how accurate are the velocity and position estimates, and does the estimation of platform state improve the results?

TABLE IV. The test cases for the manually associated data.

Case	Description
1	Kalman prediction with a fixed Euclidean gate (static gate)
2	"Zero-velocity" prediction with a fixed Euclidean gate (static gate)
3	Kalman prediction with a maximum likelihood gate (dynamic gate)
4	As case 1, but with no correction for platform movement
5	As case 2, but with no correction for platform movement
6	As case 3, but with no correction for platform movement

A. Data association

We tested the performance of six different cases of data association on the manually associated data sets (see Table IV). The association measures defined in Eqs. (12)–(14) are used in the tests. All cases were tested with both GNN and BPF data-association methods.

TABLE V. List of parameters and the optimal parameter settings for the different cases. See Appendix A and Appendix B for details of the error model parameters. N/A indicates "not applicable." Note that all angles (θ, ϕ, ψ, α, and β) are given in degrees.

		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Σ_{Φ}	Q_{xy}	0.100	0.100	0.100	0.100	0.100	0.100
	Q_z	0.050	0.050	0.050	0.050	0.050	0.050
	Q_{dxdy}	0.150	0.149	0.151	0.150	0.148	0.151
	Q_{dz}	0.100	0.100	0.100	0.100	0.100	0.099
$\Sigma_{\mathbf{x},0}$	$P_{0\ xy}$	0.200	0.200	0.200	0.200	0.200	0.200
	$P_{0\ z}$	0.100	0.100	0.100	0.100	0.100	0.100
	$P_{0\ dxdy}$	0.300	0.300	0.300	0.300	0.298	0.300
	$P_{0\ dz}$	0.200	0.200	0.200	0.200	0.200	0.200
Σ_R	$R_{x'}$	0.100	0.100	0.100	0.100	0.100	0.100
	$R_{y'}$	0.100	0.100	0.100	0.100	0.100	0.100
	$R_{z'}$	1.000	1.000	1.000	1.000	0.980	1.005
	$R_{\Theta'}$	1.000	1.000	0.995	1.000	1.005	0.995
	$R_{\phi'}$	1.000	1.000	1.000	1.000	1.005	0.995
	$R_{\psi'}$	0.500	0.500	0.500	0.500	0.500	0.500
	R_{α}	0.500	0.500	0.500	0.500	0.500	0.497
	R_{β}	0.500	0.500	0.500	0.500	0.502	0.500
	R_r	0.100	0.100	0.100	0.100	0.100	0.100
	R_{TS}	0.000	0.000	0.000	0.000	0.000	0.000
Track support	f_0	0.050	0.049	0.050	0.050	0.049	0.050
	MN	0.400	0.400	0.400	0.400	0.400	0.400
	MP	4.000	5.000	5.000	4.000	5.000	4.000
	ML	15.000	15.000	15.000	15.000	15.000	15.000
Initial gate	$\alpha_{G1}\beta_{G1}$	3.940	4.020	N/A	3.940	4.020	N/A
	r_{G1}	1.000	1.000	N/A	1.000	1.000	N/A
	c_{G1}	N/A	N/A	1.000	N/A	N/A	1.000
Final gate	$\alpha_{G2}\beta_{G2}$	6.862	4.018	N/A	6.724	4.996	N/A
	r_{G2}	1.082	1.000	N/A	1.082	1.040	N/A
	c_{G2}	N/A	N/A	1.000	N/A	N/A	1.000

TABLE VI. The data-association error for the six cases. The optimal parameter settings for each case are given in Table V.

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
J_{alloc}	0.017	0.022	0.041	0.017	0.026	0.061
J_{split}	0.008	0.020	0.041	0.014	0.038	0.060
J_{connect}	0.025	0.024	0.041	0.021	0.014	0.062

To do a fair comparison of the methods, we use the parameters that give the lowest J_{alloc} for each case. These parameters are found initially by a searching over the parameter space, then a gradient method is used to minimize J_{alloc}. The resulting parameters are given in Table V. Note that MN and ML are not in the optimization since the association measure does not contain a penalizing term for excluding short tracks. If the longest track is correctly associated, the optimal ML will be equal to the length of the longest track. The parameter estimation procedure is performed using the BPF data-association algorithm.

The split error, the connect error, and the association error for the six cases are given in Table VI. The GNN and BPF methods showed no difference in the data association error, and BPF is used since it is computationally less demanding. The association error is lower for a static gate compared to a dynamic gate. Velocity prediction (cases 1 and 4) decreases the association error. Note that the optimal final horizontal gates (α_{G2} and β_{G2}) are larger when velocity prediction is included (see Table V). The platform-state estimation gives little improvement. When velocity prediction is used (cases 1 and 4), the connect error increases by 25%, while the split error is reduced by 50% by the platform estimation.

Another way to evaluate the effect of platform-state estimation is to look at the distribution of the innovations (i.e., the differences between predictions and observations). The innovation distributions for the initial and final tracking results for case 1 are shown in Fig. 4. To compare the distributions, the initial gate is set equal to the final gate. The innovation distribution is shifted left, indicating a better fit between prediction and measurement. Examples of data with and without correction for platform movements are given in Figs. 5 and 6.

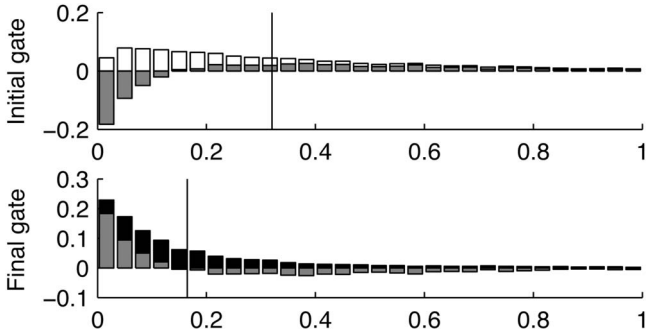

FIG. 4. The white bars (upper panel) and black bars (lower panel) show the gate distance (d^2) distribution for case 1 without and with platform-state estimation, respectively. The gray bars indicate the difference between the distributions. The vertical lines indicate the means for the distributions.

As indicated above, changes in the various parameters influence the association error. The sensitivity index defined in Eq. (15) is used to test this, and the results are shown in Fig. 7. The results are sensitive to MP and MN in all cases. When using static gates, the gate parameters are important. In particular, note the high sensitivity when platform-state estimation is omitted. This is seen for the vertical gates in cases 4 and 5, and especially for the horizontal gates in case 5. The sensitivity to horizontal gates is lower when velocity

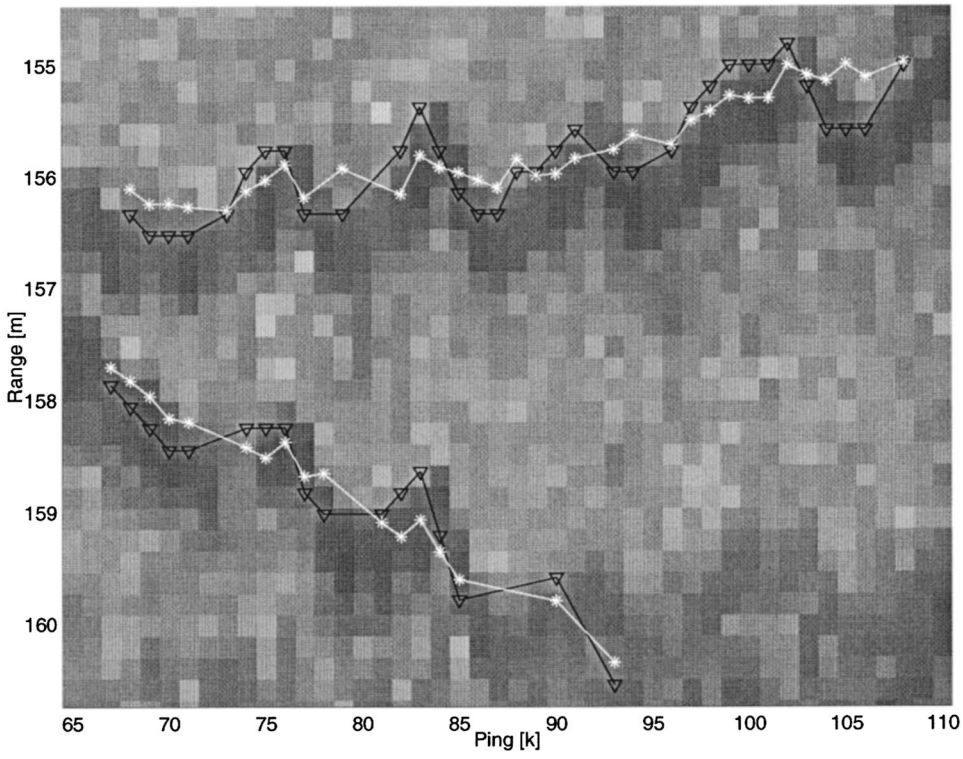

FIG. 5. Range as a function of ping number for two tracks. Asterisks (white) and triangles (black) show the tracks with and without buoy movement estimation, respectively.

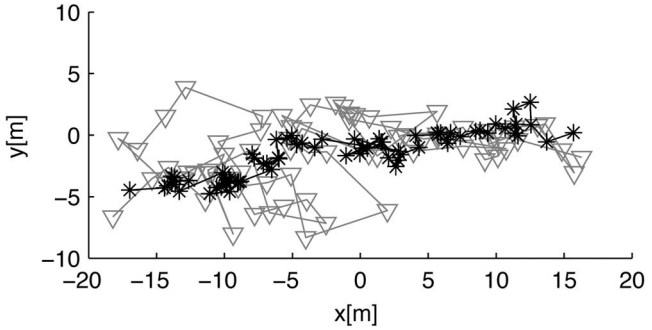

FIG. 6. Horizontal positions for a single track. Asterisks and triangles show the track with and without buoy movement estimation, respectively.

prediction is used. The sensitivity to the error model parameters (R) is higher when using dynamic gates. Note also that the detection probabilities, hidden in c_G, are less important compared to the error model parameters.

B. Track state estimation

We used the simulated data sets described in Sec. III E to investigate the accuracy of the various track-state estimators. The data sets are tracked, with and without platform-state estimation, and the position and velocity estimates are evaluated using the measures defined in Eqs. (17) and (18).

Tables VII and VIII show the mean position and velocity errors for the five estimation techniques and for three different scenarios. Firstly, the data are simulated with platform movement, and platform-state estimation is applied. Secondly, the platform movement is simulated but not estimated. The last simulation is the "control," where platform movement is neither simulated nor estimated. Comparing the results with and without platform movement estimation, the position estimate error is reduced for the Kalman methods KS and the KF when the platform movement is estimated, but for the L and SP methods it increases. For the velocity estimates, the errors of KS indicate divergence in all cases.

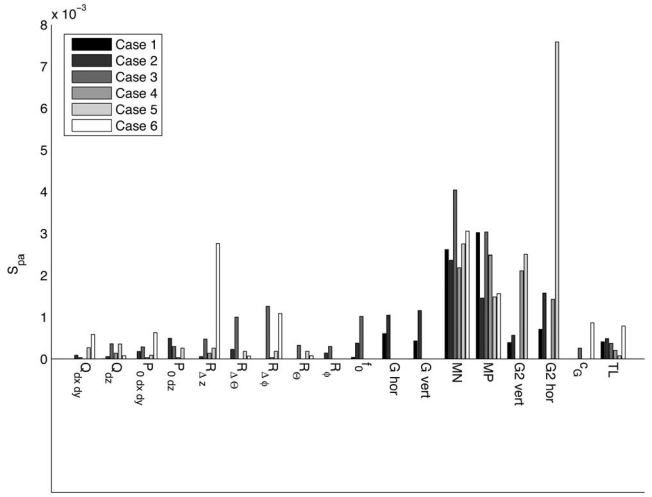

FIG. 7. Sensitivity indices for association error, S_{pa}. Nonsensitive parameters are not included.

TABLE VII. Mean absolute position errors for different track-state estimators.

	KS	KF	L	SNP	SP
PlatMove, PlatEst.	1.082	1.368	1.628	1.336	1.158
PlatMove, No PlatEst.	1.929	2.577	1.303	3.295	0.829
No PlatMove, No PlatEst.	0.627	1.138	1.029	0.417	0.346

For KF and the SNP the velocity error is reduced by estimating the platform state, whereas for L and SP the error levels are similar.

The sensitivity to the estimated velocity and position is tested by calculating the measures S_{ps} and S_{pv}, using the various track-state estimators on the test data set described in Sec. II. The position sensitivity measure, S_{ps}, is the mean difference in position. The position estimates deviate at the most 0.3 m when perturbing the parameters ±10% (Fig. 8), and they depend mainly on f_0. This parameter controls the platform estimation. All the techniques are equally sensitive, except that SP is also sensitive to the smoothing parameter spar.

As regards the velocity estimates, there are differences between the techniques, although all except SP are most sensitive to changes in f_0 (Fig. 9). The sensitivity to f_0 is extremely high for KS and SNP, while it is lower for KF and SP, with L being the most robust method. Note also the high sensitivity to *spar* for the SP method.

V. DISCUSSION

This paper presents a general framework for tracking fish with a split-beam echosounder when the transducer is moving. The performance of various methods of data association and track estimation is investigated.

A. The extended Kalman filter

The extended Kalman filter serves two purposes in the MTT. It links observations to predictions, taking into account errors due to the prediction, the echosounder, and the platform movement, and it is used to estimate positions and velocities. The extended version of the Kalman filter is required due to the nonlinear relationship between the measurement space and the state space. The main contribution here is a framework for tracking when the error models are known. The track-state-transition error may be seen as the deviation in swimming speed from a straight line (constant velocity) and has nothing to do with the measurement errors. This enables us to separate the observation error from that due to nonconstant fish velocity. Track state transition models based on the observed behavior, the coordinated turn model, for

TABLE VIII. Mean relative velocity errors for different track-state estimators.

	KS	KF	L	SNP	SP
PlatMove, PlatEst.	315%	52%	31%	68%	32%
PlatMove, No PlatEst.	721%	88%	28%	815%	35%
No PlatMove, No PlatEst.	428%	51%	23%	45%	16%

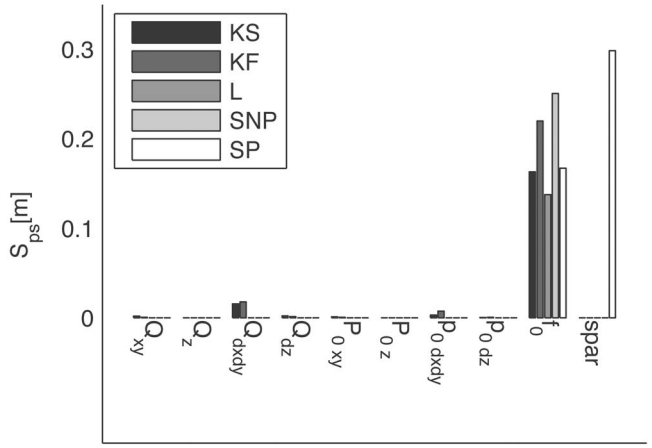

FIG. 8. Sensitivity indices for estimated position, S_{ps}. The P and Q parameters are the elements of the parametrized $\Sigma_{\mathbf{x},0}$ and Σ_Φ, respectively. See Eqs. (6) and (A1) for details. Nonsensitive parameters are not included.

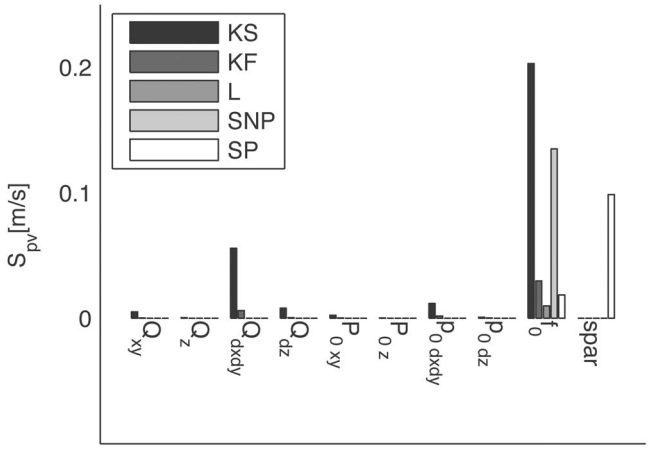

FIG. 9. Sensitivity indices for estimated velocity, S_{pv}. The P and Q parameters are the elements of the parametrized $\Sigma_{\mathbf{x},0}$ and Σ_Φ, respectively. See Eqs. (6) and (A1) for details. Nonsensitive parameters are not included.

example,[5] might improve the track-state transition (prediction). We decided against that approach because of the large errors in our observations. The convergence of the Kalman filter depends on the error model. This is discussed in Sec. IV D.

To take full advantage of the EKF, proper error models for the measurements and state transitions should be established. This would allow us to estimate the track state including both error models. This is not a feature of the alternative track-state estimators, except KS, and is the reason for the central role the EKF has in this work.

B. Platform-state estimation

Since we had no measurements of the platform tilt and roll, there was a need to estimate this movement somehow. We used a simple approach to solve this problem, but other techniques like fitting a model of wave-induced movement to the data could be considered. We tried using a bandpass filter to remove low and high frequencies, but the association error was more than that of our simple algorithm (which may be compared to a high-pass filter). We therefore chose not to proceed with more elaborate models. The reason why the simple filter works well in our example is probably due to the rigging of the transducer. The transducer hangs from a cable, and the dynamics of this arrangement are complicated. The motion of a transducer attached to a ship's hull is probably simpler to model, and more complex models may improve performance in that case. It is important to note that our method does not detect any constant tilt of the acoustic axis; only the periodic movement about the (unknown) mean is estimated. If measurements of the platform state are available, the platform estimation technique and the second tracking are unnecessary.

The algorithm seems to estimate the platform state quite well, but what are the consequences of utilizing these estimates? This is discussed below, both for data-association error and track-state estimation.

C. Data association

Correct data association is crucial for successful tracking. If the data-association algorithm fails, the platform state and the tracks will be poorly estimated. An important part of the data association is accurate prediction. We have tested two kinds of predictions, the constant-velocity prediction and the "zero-velocity" prediction. When comparing the constant-velocity and "zero-velocity" predictions, we see that the split error is smallest in the former prediction (cases 1 and 4). This indicates a better result from the constant-velocity prediction, which allows the data-association algorithm to handle more candidates within a gate, hence the larger optimal horizontal gates for cases 1 and 4. However, the larger horizontal gates cause more connection error, although the total association error is still improved. There is a trade-off between connect and split errors. If one desires to reduce the connect error, narrower horizontal gates may be applied. The total association error is then increased, but it is still lower than that of the "zero-velocity" prediction. We conclude that it is better to base predictions on the actual velocity estimates rather than the zero-velocity assumption.

The platform-state estimation has a similar effect: The split error is reduced, while the connect error is increased, but here the total association error remains the same. The benefit of platform-state estimation in the data-association process is not evident. However, if we decrease the horizontal gates from 6° to 4°, the platform-state estimation does improve the results. The J_{alloc} is then 0.07 and is reduced to 0.02 after platform-state estimation. For some parameter values the platform-state estimation is important, but not when using the optimal settings for our example. However, when considering the innovation distribution, it is seen that the platform-state estimation reduces the distance between the predictions and the observations. In addition, the sensitivity to the gate parameters is higher without platform-state estimation. Thus, the platform-state estimation makes the association algorithm more robust.

We tried to implement maximum-likelihood track-support functions.[5] This would avoid *ad hoc* parameters like missing pings, MN ratio, and predetermined gates. The idea

is to use the prediction covariance to define the gate, but it is necessary to estimate the detection probabilities. We tested whether we could improve the results by setting constant detection probabilities and then optimizing the parameters with respect to the data-association error. These probabilities are in fact not constant, and we were unable to come close to the performance of a fixed Euclidean static gate and *ad hoc* parameters. One reason for the failure could be the narrow acoustic beam. The transducer's pendulum movement causes the detection probabilities to change rapidly, thus the use of constant parameter values may be incorrect. Consequently, we cannot rule out the possibility that the maximum-likelihood track-support functions may work. However, with the accuracy of the detection probabilities and the error models in mind, we believe that the *ad hoc* approach is more robust.

A similar situation is found for the data-association algorithms. Although the GNN algorithm is quite simple, the even simpler method of picking the nearest target first works equally well. This indicates that the system has low vulnerability to the choice of the data-association algorithm. This may be explained by the nature of the SED which detects only one target at a given range, and rejects closely spaced targets. It is concluded that the choice of data-association algorithm is not crucial to the result. These conclusions are based on the data-association test. A crucial part in the test is the validity of the true tracks. These are obtained by first applying the tracker and then correcting the tracks manually. The manual data association is subjective to some extent, and only the target range, not the angles, is used in the manual process. Therefore some false track connections might occur. In addition, our results are based on only one data set, and the result is, strictly, only valid for this data set. The convergence of the optimization algorithm is also important. If the algorithm sticks on a local J_{alloc}, wrong conclusions may be drawn. Also, the relative weights of J_{split} and J_{connect} influence the results, as illustrated by the above-mentioned example of increased horizontal gates.

For our data, the constant-velocity prediction improves data association, while the platform-state estimation reduces the sensitivity to the gate parameters. In general, these effects are connected, and we argue that both methods improve the quality of the data association.

D. Track-state estimators

The validity of the estimated positions and velocities are central to the success of the tracker. Tracking fish from a free-floating and moving buoy with high noise in the positioning may require a different approach compared to the analysis of fine-scale behavior from a known or stable platform state.

The Kalman filter (KF) is an integral part of the MTT, and error models of the measurements and fish movement are taken into account. The KF works well for the position estimates. Each position includes an estimate of the error, assuming that the error models are correct. The velocity estimate is less good. One reason is that the initial velocity is set to zero, and it takes several pings for the velocity estimate to converge. This depends on the initial state estimate error and the measurement error model. An important feature of the KS estimate is that the tracker runs backwards through the tracks, thus eliminating the impact of the initial estimate covariance. The problem with KS is that accurate error models and platform-state estimates are usually necessary to prevent the results from diverging. In our case this was not so. KS is excellent for the position estimates, which are greatly improved when the platform estimation technique is applied.

A linear regression fit to the track loci may be a crude approximation. This is seen in the rather poor position estimates obtained from the regression method. However, the linear regression L is a robust method for estimating velocity. The velocity estimate from L is the same with or without platform-state estimation. This occurs because the periodic buoy movement is removed by the straight-line regression, provided the tracks are longer than half the period of the platform movement. Another advantage is the low sensitivity to the tracking parameters. The simple linear regression is a good choice when the noise level is high and a robust method is required, although the amount of detail in the results is limited.

The nonparametric spline works fairly well for the position estimates. Cross validation is used to set the parameters in the interpolation. This approach automatically determines the relative weights according to smoothness and fit and has the advantage that *ad hoc* parameters are avoided. However, the estimated positions are rather bad if the platform-state estimation is omitted. The cross-validation technique detects the periodic movement of the platform and tends to follow the observations and the buoy movement closely. This leads to bad position estimates. The velocity estimate of SNP is unsatisfactory. The derivative of the spline is used, and if the degree of smoothness is too low, the velocities will be grossly over estimated. On the other hand, SP seems to work well with spar=0.7. This velocity estimate is the best, but it is highly sensitive to spar. When a high degree of smoothing is applied, the spline smooths the buoy movement in a similar manner to the linear regression. In that case, there is no evidence that estimating the buoy movement is useful.

As discussed earlier, the various track-state estimators have different properties. However, these conclusions are based on the simulated data set, which is similar to our test data set. For our test data the robust linear regression seems to be the best choice. Other data sets may have different properties, like higher ping rates, fewer missing pings, etc. It might then be feasible to incorporate more detailed track-state estimators. A general statement of what is the best estimator is therefore difficult.

E. Other tracking issues

Although we have said little about the algorithm for single-echo detection (SED), the efficiency of this algorithm is an important factor in target tracking. Traditional SED methods are discussed in the literature.[3] For weak targets, systematic angle measurement errors may occur.[13] If the SED were improved, the maximum allowed number of missing pings could be reduced with consequent benefits for the

platform-state estimation, the data association, and the track estimation. There are promising developments in this field.[14]

The orientation of the transducer may change between the transmitted and the received pulses. The adverse effect on echo integration has been investigated.[15] The same problem may affect the SED efficiency, and also the target-angle measurements. This requires further investigation, as regards the effects on target detection algorithms and the performance of target trackers.

VI. CONCLUSIONS

We have shown that compensating for platform movement improves the quality of the data association, especially when that is combined with predictions. Consequently, the platform state should be taken into account whenever possible, and predictions should be based on the estimated velocity. The various data-association algorithms we tested performed very similar in our case, and we conclude that the tracker is not sensitive to the choice of data-association algorithm. On the other hand, the choice of gate had a large influence, and we found that the simple static gate worked best in our case. However, we cannot rule out the idea that dynamic gates might perform better, given better knowledge of detection probabilities and improved error models.

The choice of state estimation technique depends on the quality of the data and how they are used. For example, more detailed methods like SNP and KS require more accurate estimates of platform state in order to perform well. Coarse-scale estimators like linear regression are less vulnerable to incorrect platform-state estimates. Velocity is more difficult to estimate than position, and this should be taken into consideration when choosing the track-state estimator. For our data, the linear regression yielded a robust and relatively accurate estimate of the velocity.

The EKF approach differs from the other techniques for estimating the track state, since it takes account of errors both in the measurements and in the prediction model, and it models the two error types separately. To take full advantage of this approach, better error models must be developed. This work presents the framework for such an approach which, in our opinion, is the natural next step.

ACKNOWLEDGMENTS

We are grateful to Professor Egil Ona for useful comments on the manuscript and ideas and suggestions during the development of the method. The Norwegian Research Council has financially supported the work (Project 133426/120).

APPENDIX A: THE TRACK-STATE TRANSITION

Under the assumptions of constant velocity, the track state at time $t_{k+1}=t_k+\Delta T$ is given by

$$\mathbf{x}_{k+1}=\Phi(\Delta T)\mathbf{x}_k+\mathbf{w}_\Phi,$$

where $\Phi(\Delta T)$ is the track-state-transition matrix for the time interval ΔT and $\mathbf{w}_\Phi$ is an additive system error component. The track-state-transition matrix is given by

$$\Phi(\Delta T)=\begin{bmatrix} 1 & 0 & 0 & \Delta T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta T & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta T & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

and the target-model-covariance matrix is defined as $\Sigma_\Phi=E(\mathbf{w}_\Phi\mathbf{w}_\Phi^T)$. The error is assumed normal, independent, and constant for all track-state variables and

$$\Sigma_\Phi=[Q_{xy}^2 \quad Q_{xy}^2 \quad Q_z^2 \quad Q_{dxdy}^2 \quad Q_{dxdy}^2 \quad Q_{dz}^2 \quad Q_{TS}^2]I \tag{A1}$$

where the elements are parameters and I is the identity matrix.

APPENDIX B: THE MEASUREMENTS

A split-beam echosounder with an algorithm for single echo detection calculates the alongship angle α_k, the athwart ship angle β_k, the range r_k, and the target strength TS_k, for each target inside the sampled volume at every time step k. These measurements are represented by

$$\mathbf{y}_k=[\alpha_k \quad \beta_k \quad r_k \quad \mathrm{TS}_k]_{e_{tp}}^T+\mathbf{v}_{y,k}.$$

The measurement has an additive error component, $\mathbf{v}_{y,k}$, given by the accuracy of the transducer.

The transducer platform state is described by the position, $\hat{x}'$, $\hat{y}'$, and $\hat{z}'$ relative to geo-referenced coordinates (e_{gc}), the transducer yaw (transducer compass reading) ψ', and the tilt angles θ' and ϕ' relative to east-west and north-south directions, respectively (see Fig. 2). Note that this is not the same as the usual vessel tilt and roll angles,[12] which are relative to vessel heading. The translation, tilt, and twist of the transducer are given by

$$\mathbf{z}_k=\hat{\mathbf{z}}_k+\mathbf{v}_{y,k}=[\hat{x}' \quad \hat{y}' \quad \hat{z}' \quad \hat{\theta}' \quad \hat{\phi}' \quad \hat{\psi}']_{e_{gc}}^T+\mathbf{v}_{z,k},$$

where $\mathbf{v}_{z,k}$ is the additive error component. The error terms are combined into

$$\mathbf{v}_{R,k}=\begin{bmatrix}\mathbf{v}_{z,k}\\ \mathbf{v}_{y,k}\end{bmatrix}$$

and assumed additive and normal with covariances $\Sigma_{R,k}=E(\mathbf{v}_k\mathbf{v}_k^T)$. The covariances are given as

$$\Sigma_{R,k}=\begin{bmatrix}\Sigma_{z,k} & 0\\ 0 & \Sigma_{y,k}\end{bmatrix},$$

where

$$\Sigma_{z,k}=[R_{x'}^2 \quad R_{y'}^2 \quad R_{z'}^2 \quad R_{\Theta'}^2 \quad R_{\phi'}^2 \quad R_{\psi'}^2]I$$

and

$$\Sigma_{y,k}=[R_\alpha^2 \quad R_\beta^2 \quad R_r^2 \quad R_{TS_k}^2]I.$$

Here the elements are parameters and I is the identity matrix.

The mapping from state space to measurement space, $\mathbf{h}:(\hat{\mathbf{x}}_k,\hat{\mathbf{z}}_k,\mathbf{v}_k)\rightarrow\mathbf{y}_k$, is a central part in the EKF. The mapping involves Cartesian translation and rotation, and changing from Cartesian to transducer coordinates. Note that the positive component of the acoustic axis is pointing away from the transducer. The relationship between the transducer Cartesian coordinates (e_{tc}) and the transducer polar coordinates (e_{tp}) is given by

$$\mathbf{y}_{e_{tp}}=\begin{bmatrix}\arctan\dfrac{x}{z}\\ \arctan\dfrac{y}{z}\\ \sqrt{x^2+y^2+z^2}\\ \mathrm{TS}\end{bmatrix}_{e_{tp}}+\mathbf{v}_{y,k}, \tag{B1}$$

where x,y,z are the target position in e_{tc} coordinates and TS is the target strength. The target position in e_{tc} is given by the mapping between the geo-referenced track-state coordinates (e_{gc}) and transducer platform Cartesian coordinates (e_{tc}), defined by

$$\begin{bmatrix}x\\ y\\ z\\ \mathrm{TS}\\ 1\end{bmatrix}_{e_{tc}}=A_cA_t\begin{bmatrix}\mathbf{x}_{e_{gc}}\\ 1\end{bmatrix}, \tag{B2}$$

where A_t is the translation matrix and A_c is the rotation matrix, defined as

$$A_t(\mathbf{z}_{k[1\cdots3]})=\begin{bmatrix}1&0&0&0&0&0&0&-x'_k\\ 0&1&0&0&0&0&0&-y'_k\\ 0&0&1&0&0&0&0&-z'_k\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1\end{bmatrix},$$

and

$$A_c(\mathbf{z}_{k[4\cdots6]})=\begin{bmatrix}&&&0&0\\ \{\mathbf{e}_x\}&\{\mathbf{e}_y\}&\{\mathbf{e}_z\}&0&0\\ &&&0&0\\ 0&0&0&1&0\\ 0&0&0&0&1\end{bmatrix}^{-1}$$

$$=\begin{bmatrix}\{\mathbf{e}_x^T\}&&0&0\\ \{\mathbf{e}_y^T\}&&0&0\\ \{\mathbf{e}_z^T\}&&0&0\\ 0&0&0&1&0\\ 0&0&0&0&1\end{bmatrix}.$$

The unity vectors are defined by

$$\mathbf{e}_x=k_2\begin{bmatrix}\sin\psi'\\ \cos\psi'\\ k_1\end{bmatrix}_{e_{gc}},$$

$$\mathbf{e}_y=k_2k_3\begin{bmatrix}k_1\tan\phi+\cos\psi\\ -\sin\psi-k_1\tan\theta\\ \tan\theta\cos\psi-\tan\phi\sin\psi\end{bmatrix}_{e_{gc}},$$

$$\mathbf{e}_z=k_3\begin{bmatrix}\tan\theta'\\ \tan\phi'\\ -1\end{bmatrix}_{e_{gc}},$$

where

$$k_1=\tan\theta'\sin\psi'+\tan\phi'\cos\psi',$$

$$k_2=(\sin^2\psi'+\cos^2\psi'+k_1^2)^{-1/2}, \tag{B3}$$

$$k_3=(\tan^2\theta'+\tan^2\phi'+1)^{-1/2}.$$

Note that the platform-state variables are the sum of the error and the estimate, i.e., $x'=\hat{x}'+v_{z,k[1]}$, thus giving the dependence of $\mathbf{v}_z$ in the mapping.

In order to spawn a new track and to facilitate the use of the alternative track-state estimators, the mapping from measurement $\mathbf{y}_k$ to position $\mathbf{s}_k$ is required, i.e., $\mathbf{g}:(\mathbf{y}_k,\hat{\mathbf{z}}_k)\rightarrow\mathbf{s}_k$. First the transducer coordinates (e_{tp}) are converted to transducer Cartesian coordinates,

$$z=r/\sqrt{\tan^2\alpha+\tan^2\beta+1},$$
$$x=z\tan\alpha,$$
$$y=z\tan\beta.$$

Rotation and translation are applied to arrive at the geographical Cartesian coordinates (e_{gc}),

$$\mathbf{s}_k=A_{ti}A_c^{-1}\begin{bmatrix}x\\ y\\ z\\ \mathrm{TS}\\ 1\end{bmatrix}, \tag{B4}$$

where A_c^{-1} is the inverse of A_c and

$$A_{ti}(\mathbf{z}_{k[1\cdots3]})=\begin{bmatrix}1&0&0&0&+x'_k\\ 0&1&0&0&+y'_k\\ 0&0&1&0&+z'_k\\ 0&0&0&1&0\\ 0&0&0&0&1\end{bmatrix}.$$

[1]R. Brede, F. H. Kristensen, H. Solli, and E. Ona, "Target tracking with a split-beam echo sounder," Rapp. P.-v. Réun. Cons. Int. Explor. Mer. **189**, 254–263 (1990).

[2]E. Ona, Chap. 10. Recent Developments of Acoustic Instrumentation in Connection with Fish Capture and Abundance Estimation, in *Marine Fish Behaviour*, edited by A. Fernö and S. Olsen (Fishing News Books, Oxford, England, 1994),

[3]M. Soule, I. Hampton, and M. Barange, "Potential improvements to current methods of recognizing single targets with a split-beam echosounder," ICES J. Mar. Sci. **53**, 237–243 (1996).

[4]S. S. Blackman, *Multiple Target Tracking with Radar Applications* (Artech House, Boston, MA, 1986).

[5]S. S. Blackman, *Design and Analysis of Modern Tracking Systems* (Artech House, Boston, MA, 1999).

[6]O. R. Godø, D. Somerton, and A. Totland, "Fish behaviour during sam-

pling as observed from free floating buoys - application for bottom trawl survey assessment," ICES CM 1999/J:10 (1999).
[7]N. O. Handegard, K. Michalsen, and D. Tjøstheim, "Avoidance behaviour in cod (*Gadus morhua*) to a bottom-trawling vessel Aquatic Living Resources **16**, 265-270 (2003).
[8]S. S. Blackman, in *Design and Analysis of Modern Tracking Systems* (Artech House, Boston, MA, 1999), Chap. 6.
[9]D. Bertsekas, "The auction algorithm for assignment and other network flow problems: A tutorial," Interfaces **20**, 133–149 (1990).
[10]R Development Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2003.
[11]J. M. Chambers and T. J. Hastie, *Statistical Models in S* (Wadsworth/Brooks Cole, Pacific Grove, CA, 1992).
[12]SNAME, "Nomenclature for treating the motion of a submerged body through a fluid," Technical Report Bulletin 1-5. Society of Naval Architects and Marine Engineers, New York (1950).
[13]R. Kieser, T. Mulligan, and J. Ehrenberg, "Observation and explanation of systematic split-beam angle measurement errors," Aquatic Living Resources **13**, 275–281 (2000).
[14]H. Balk and T. Lindem, "Improved fish detection in data from split beam transducers," Aquatic Living Resources **13**(5), 297–303 (2000).
[15]T. Stanton, "Effects on transducer motion on echo-integration techniques," J. Acoust. Soc. Am. **72**, 947–949 (1982).