

# Experimental Evaluation of Tracking Algorithms used for the Determination of Fish Behavioral Statistics

Chad Schell, *Member, IEEE*, and Stephen Paul Linder, *Member, IEEE*

**Abstract**—The fine scale swimming behavior of fish can now be studied because of the development of sophisticated measurement devices such as multibeam sonar and stereo video systems. However, even with these sensors, improved methods are still required to generate quality estimates of swimming speeds and turn rates. Biologists have commonly relied on point-wise differentiation of noisy position measurements, while engineers have focused on Bayesian algorithms to track underwater vehicles. A comparative evaluation of the performance of these tracking algorithms for the analysis of fine scale behavior of fish was performed using a data set of 100 fish track tracks recorded simultaneously with a multibeam sonar and a stereo video camera system. The Segmenting Track Identifier, a non-Bayesian curve fitting and segmenting tracker, is shown to be most effective for tracking the unpredictable and complex horizontal motion of fish, while a Kalman smoother using a constant-velocity model is shown to be most effective for tracking the more predictable and piecewise linear vertical motion of fish. Both are shown to be more effective than point-wise differentiation. Criteria for selecting an appropriate algorithm for a given motion study are provided.

**Index Terms**— Biological system modeling, Estimation, Marine animals, Sonar tracking, Tracking filters.

## I. INTRODUCTION

THE swimming behavior of individual fish on a fine scale can now be studied because of the increasing use of sophisticated measurement devices, such as multibeam sonars and video camera systems, which allow the precise tracking of individual fish. Examples of studies at these scales include Johnson and Moursund's [12] study of fish behavior at the entrance to a surface collector at Bonneville Dam using multibeam sonar, and the single [22] and stereo [4-6, 20, 21] camera based experiments conducted to study the energetic costs of free-swimming. However, even when using high resolution sensors, as used in these studies, quantifying animal

Manuscript received May 5<sup>th</sup>, 2004. This material is based in part upon work supported under a National Science Foundation Graduate Fellowship.

C. Schell was with the San Diego Electrical and Computer Engineering Department and the Scripps Institution of Oceanography, University of California, La Jolla, CA 92093 USA. He is now with Rincon Research Corporation, Tucson, AZ 85711 USA (e-mail: chad@schells.com).

S. P. Linder is with the Department of Computer Science, Dartmouth College, Hanover, NH 03755-3510 USA (e-mail: spl@alum.mit.edu).

behavior at this scale requires careful processing of the data to ensure quality results.

The most common data processing technique used to estimate fish behavior in current studies is point-wise differentiation of the noisy measurements, where fish velocities and accelerations are calculated by differentiating the noisy position measurements. The limitations of point-wise differentiation to estimate the swimming speeds of fish, including its tendency to generate biased estimates with unnecessarily large variance was shown by Mulligan and Chen [16] in a response to study exploring the use of split-beam sonar to estimate swimming speeds [1]. Regression was instead suggested as the preferred methodology for estimating swimming speed.

This paper examines the use of tracking algorithms as an alternative method of processing position only measurements to obtain behavioral statistics, as tracking algorithms have the advantage that they can be used in complex data association frameworks to extract individual fish tracks from data sets with multiple fish or fish swimming in clutter. The performance of the tracking algorithms is shown to be superior to that of point-wise differentiation.

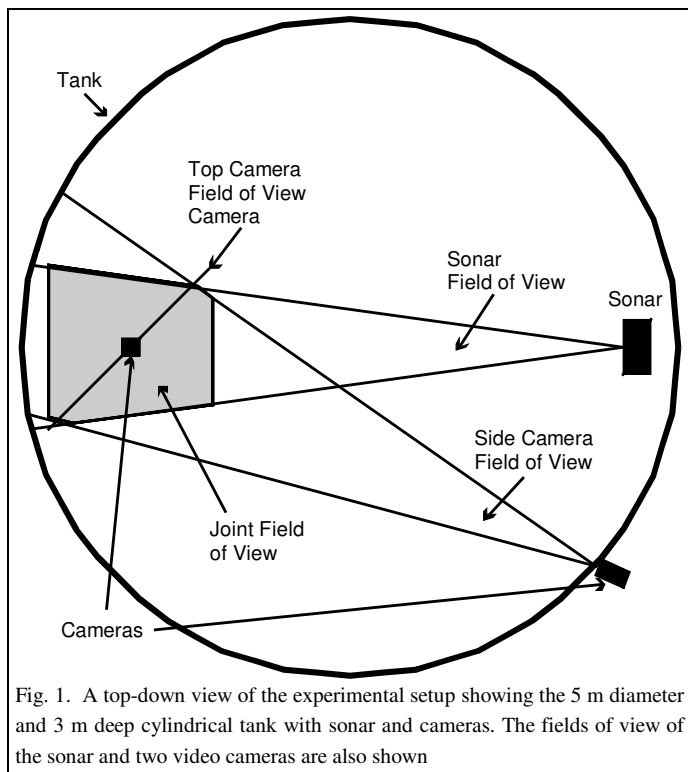
## II. METHODS

### A. Experimental Design

The tracking algorithms were evaluated using a laboratory data set of 100 fish tracks simultaneously recorded by a multibeam sonar and a stereo video camera system. The more accurate and higher frame rate video data was used as the "ground truth" when evaluating the efficacy of the tracking algorithms in processing the sonar data.

The fish consisted of a mix of common goldfish (*Carassius auratus*), a silver dollar fish (*Metynnis argenteus*), and an unknown species of Australian rainbow fish (*Melanotaenia sp.*) all swimming freely in a cylindrical tank 5 m in diameter and 3 m deep shown in Fig. 1. The majority of the tracks recorded by the sonar were rainbow tracks as they were the only fish that regularly swam through the middle of the tank. The other fish tended to stay close either to the side of the tank, the tank bottom, or the water surface, areas outside the sonar's field of view.

The sonar system was mounted at one end of the tank at mid



water level, and the cameras were placed to provide top down and side looking coverage of the sonar's observation area. The tank's pumps and filters were shut off shortly before and during recording periods so there were effectively no currents in the tank.

Sonar measurements were recorded at 4 Hz using FishTV [11], a 445 kHz multibeam sonar whose observation volume is a  $16^\circ$  by  $16^\circ$  wedge divided into an 8 by 8 grid of  $2^\circ$  by  $2^\circ$  beams. FishTV's angular resolution was increased to  $0.25^\circ$  using the interpolation algorithm described in [10], effectively dividing the observation volume into a 64 by 64 grid of  $0.25^\circ$  by  $0.25^\circ$  beams.

The stereo camera system used two JVC GR-DVF21U digital video camcorders. The video pixel coordinate positions of the side and top cameras were converted to 3-D world coordinates using the technique of Hughes and Kelly [9]. This method consists of videotaping a grid of points at known locations in the field of view of both cameras, extracting the pixel coordinate pairs for the points from each camera, and performing a non-linear regression to match world locations and pixel coordinate pairs. This method has the advantages that camera specifications are not required and there is an automatic compensation for the refraction effects at the air-water interface seen by the top camera and the air-plastic-water interfaces of the camera housing seen by the side camera. The accuracy of the conversion was tested using 100 additional points on the grid not used in the regression. The mean error (total radial distance in all three dimensions) between known 3-D grid location and calculated location was 1.08 cm with a standard deviation of 0.70 cm. Mean error along the sonar's three Cartesian axes, distance from sonar,

and distance left/right of sonar, and height, were -0.11, -0.03, and 0.07 cm with standard deviations of 0.82, 0.66, and 0.74 cm respectively.

The coordinate system of the cameras and sonar were collocated by measuring the position of the sonar with respect to the camera calibration grid, and then fine tuning the offsets to minimize the mean error between the sonar positions and the video positions of the 100 tracks. Using the points from all 100 tracks with the video positions as the true positions, the sample mean,  $\mu$ , and standard deviation,  $\sigma$ , of the sonar measurement errors in polar coordinates are:  $\mu_R = 0.35$  cm,  $\sigma_R = 2.05$  cm for range,  $\mu_B = -0.61^\circ$ ,  $\sigma_B = 0.39^\circ$  for bearing angle, and  $\mu_E = 0.045^\circ$ ,  $\sigma_E = 0.56^\circ$  for elevation angle. It is believed that the increased noise in the elevation angle is caused by the fact that the fish were not completely stationary between pulses on the individual transmitters, which sampled the vertical dimension. The lack of stationarity likely decreased the effectiveness of the interpolation algorithm. Time synchronization between the cameras and sonar was accomplished by projecting FishTV's frame count into the view of each camera.

### B. Experimental Data

This experiment was designed to study the efficacy of tracking algorithms at estimating fish behavior rather than the data association aspects of target tracking. Therefore, 100 tracks were manually extracted from the data. This process consisted of overlaying the sonar data on the video images by projecting the sonar's 3-D coordinates onto 2-D camera pixel coordinates. The sonar measurements corresponding to a single fish track were then collected from the time it entered the field of view until it left. If sonar data could not be associated with certainty to a single fish, as when multiple fish were closely grouped, no measurement was accepted for that frame, and it was marked as a missed measurement.

After extracting a fish track from the sonar data, the corresponding track was extracted from the video data at the full video frame rate of 29.97 Hz. Video pixel positions were selected by hand at roughly the center of mass of the fish, approximately 1/3 of the distance from the tip of the head to the tip of the tail. The paired pixel coordinate points from both cameras were then converted to 3-D positions.

The sonar data consists of noisy range, bearing angle and elevation angle triplets. The data was converted to Cartesian coordinates using the unbiased conversion method presented in [15], which also calculates an appropriate Cartesian measurement covariance for each measurement given the previously calculated polar measurement error standard deviations of the sonar. (The Cartesian measurement errors from a polar sensor are range and angle dependent.) The Cartesian coordinate system used in this paper is expressed relative to the sonar location, with  $\varepsilon$  representing distance in front of the sonar,  $\eta$  representing distance to the left or right of the sonar's center from the sonar's point of view (positive values to the left), and  $\zeta$  representing distance above or below

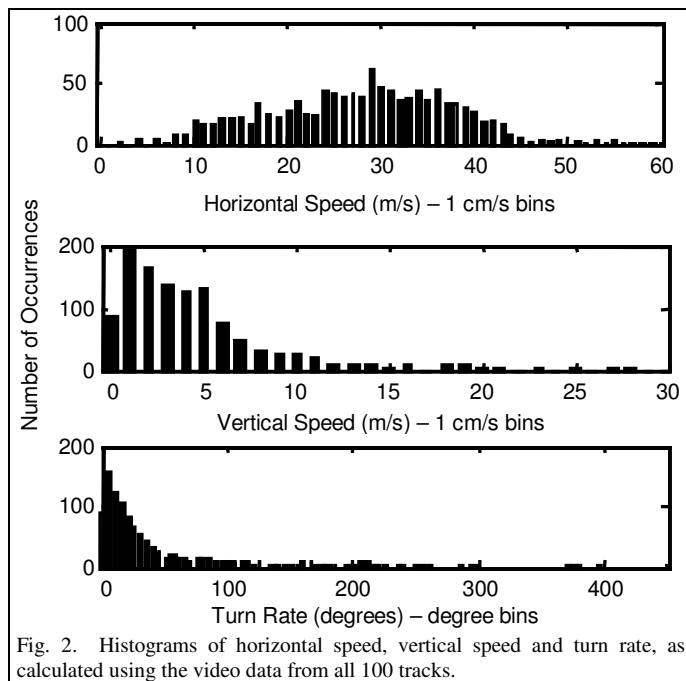


Fig. 2. Histograms of horizontal speed, vertical speed and turn rate, as calculated using the video data from all 100 tracks.

the sonar's center.

### C. Data Analysis

To simplify the data analysis and the design of the tracking algorithms, fish motion was separated into horizontal and vertical components. This decomposition of fish motion is supported by Webb [22] who argues that fish motion is often limited to a thin horizontal plane due in part to dependence on gas based buoyancy regulations and the high flexibility of vertebrates in only one direction leading to large translations occurring mostly in one plane. Our experimental observations also showed little correlation between horizontal and vertical motion.

Position only measurements can provide behavioral quantities in terms of the positions themselves, and the derived higher order derivatives of the position data such as velocity and acceleration. However, Harper and Blake [8] discuss how both framing and measurement errors can corrupt tangential (in the direction of travel) acceleration estimates obtained from position only measurements, and concludes a combination of high frame rates and highly accurate measurements are required to generate quality acceleration estimates. As this study uses low frame rate sonar data, accurate tangential acceleration estimates are impossible to achieve, so they and all other higher derivatives will not be generated. Instead the study will restrict itself to considerations of position, velocity (speed), and angular acceleration in the form of turn rates.

Fish tracks extracted manually from video data were used as the "ground truth" against which all the tracking algorithms which used sonar data are benchmarked. These video tracks were smoothed along each Cartesian axis using a fifth order Savitzky-Golay filter over 29 data points. Fig. 3 presents typical results of smoothing a typical video track and demonstrates that the sharp features of the tracks are not being

over-smoothed. Velocities along each Cartesian axis were then generated by point-wise differentiating the smoothed data, and horizontal target heading angle was computed as  $\arctan(\dot{\eta}/\dot{\epsilon})$ . The turn rate was then calculated by point-wise differentiating the target heading after it was again smoothed using a fifth order Savitzky-Golay filter.

Performance metrics are used to quantify how well each algorithm performs at estimating fish position, swimming speed, and horizontal turn rate. The root mean squared error (RMSE) between the estimated position and the true position (as provided by the video data) is used as the position metric. This is a commonly used statistic to evaluate tracking algorithms. The best algorithm will usually have the smallest RMSE. RMSE is also used to evaluate the tracking algorithms' speed estimates, but not their turn rate estimates. Because fish can almost instantaneously change their turn rate, tracking algorithms often produce outliers (large individual errors) in turn rate estimates when fish initiate a turn, as the turn rate estimate either leads or lags the true change in turn rate. Because of the squaring involved in the RMSE, these outliers can lead to very high turn rate RMSE for an estimator that is actually doing a reasonable job of tracking turn rates outside of these outliers. For this reason median absolute deviation (MAD), which is less sensitive to these outliers, will replace RMSE in evaluating turn rate performance.

However, when trying to characterize the distribution of swimming speeds or turn rates, the RMSE and MAD statistics by themselves are insufficient as performance metrics because low RMSE or MAD statistics do not necessarily imply that the distribution of swimming speeds or turn rates over several tracks has been accurately estimated. A tracking algorithm that has a slight bias or averages a rapidly changing signal can still have a low RMSE or MAD. As the distribution of swimming speeds and turn rates are critical indicators of behavior from an energetics standpoint [13, 20, 21], it is vital that the estimated distributions be correct. Therefore, a test must be done to insure that the histogram of distributions shown in Fig. 2, as calculated from the video data from one hundred tracks, is accurately estimated by a tracking algorithm. The two-sample Kolmogorov-Smirnov (KS) test was used to test the probability that the estimated distribution for all 100 tracks and the true distribution are statistically similar.

The KS test evaluates the estimation of the overall distribution of values and is most sensitive to changes close to the median value of the distributions (and thus is good at detecting bias). It is insufficient as a performance metric by itself as it can never state that in fact two distributions are equivalent, only that they are different, and because it does not provide a point-wise metric of algorithm performance. An algorithm could provide a distribution of values over several tracks that closely matches the true distribution, but never once accurately measure the desired quantity at any given point in time. However, an algorithm that performs well on both the KS test and RMSE or MAD is likely a good estimator since the overall distribution of the estimates is statistically similar

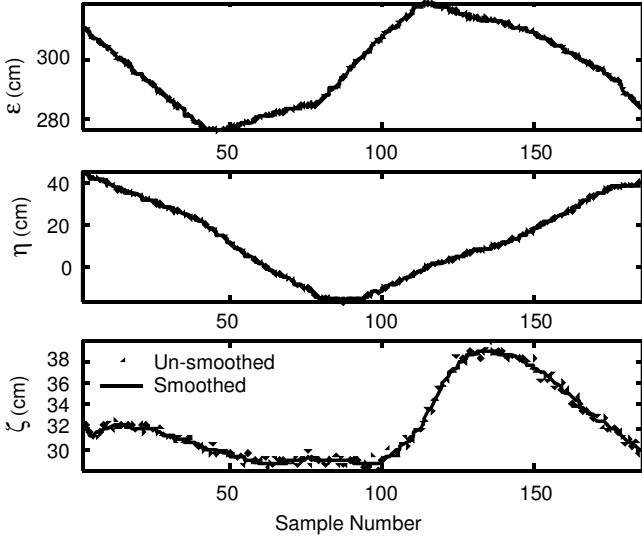


Fig. 3. Example of smoothing of raw video data for one track using fifth order Savitzky-Golay filter over 29 data points separately on each horizontal axis,  $\varepsilon$  and  $\eta$ , and the vertical axis  $\zeta$ . Note that the noise is not actually higher along the  $\zeta$  axis, but rather the total range of motion is smaller so the plot is scaled larger.

to the true distribution, and its point-wise performance at individual samples is also good.

In addition to these quantitative metrics, algorithms will also be judged on their sensitivity to their tuning parameters, with the desired behavior being a lack of sensitivity. A tracker must provide a stable and accurate estimate of behavior over a range of the tuning parameters in order for a researcher to have confidence in the estimates obtained from in-situ observations. To demonstrate both the peak performance of each algorithm, and that algorithm's sensitivity to its tuning parameters, graphs of the performance metrics versus tuning parameters will be shown. The variation in performance seen on the graphs provides a qualitative means of assessing the sensitivity of the algorithm to proper tuning. The specific tuning parameters for each algorithm, and the range over which they were tested, will be described along with the algorithms themselves in the following section.

### III. THE ALGORITHMS

Six algorithms were used to process the horizontal data for this paper. They are: point-wise differentiation; a Kalman filter and two Kalman smoothers all using a piecewise constant white acceleration model; and an extended Kalman filter (EKF) and the Segmenting Track Identifier (STI) using a constant velocity coordinated turn model. The vertical data was processed using: point-wise differentiation; Kalman filters and smoothers using a piecewise constant white acceleration model; and the STI using a constant velocity model.

Kalman filters are very commonly used tracking algorithms and many examples of their use are readily available. However, the STI algorithm is relatively new, and this study represents the first application of the STI algorithm to real data. Therefore, the Kalman algorithm's description is brief, while the STI algorithm is presented in greater detail.

#### A. Point-wise Differentiation

Point-wise differentiation refers to the technique of computing instantaneous velocities by differentiating the noisy measurements. The velocity along the  $\varepsilon$  Cartesian axis is computed as

$$\dot{\varepsilon}(k) = [\varepsilon(k) - \varepsilon(k-1)] / [t(k) - t(k-1)] \quad (1)$$

where  $t(k)$  is the measurement time of sample  $k$  in seconds. The velocities along the other Cartesian axes are calculated similarly.

The velocity estimates along the horizontal axes are then used to construct estimates of net swimming speed in the horizontal plane,  $s_h$ , heading (direction of travel),  $\theta$ , and turn rate,  $\omega$  as follows.

$$s_h(k) = \sqrt{\dot{\varepsilon}(k)^2 + \dot{\eta}(k)^2} \quad (2)$$

$$\theta(k) = \arctan(\dot{\eta}(k) / \dot{\varepsilon}(k)) \quad (3)$$

$$\omega(k) = [\theta(k) - \theta(k-1)] / [t(k) - t(k-1)] \quad (4)$$

Point-wise differentiation has no tuning parameters and will be used as the baseline for our sensitivity study.

#### B. Kalman Algorithms

Kalman filters and Extended Kalman Filters (EKF) are two of the most commonly used algorithms for target tracking. They assume that the underlying target motion, referred to as the state, can be described as a dynamic system represented by a plant equation

$$\mathbf{x}(k) = \mathbf{f}(k, \mathbf{x}(k-1)) + \mathbf{g}(k, \mathbf{v}(k)) \quad (5)$$

which describes how the state evolves with time, and a measurement equation

$$\mathbf{z}(k) = \mathbf{h}(k, \mathbf{x}(k)) + \mathbf{w}(k) \quad (6)$$

which describes the measurement available from the state. The Kalman filter is used when the equations are linear, and the EKF is used when either one or both are non-linear. The vectors  $\mathbf{x}(k)$  and  $\mathbf{z}(k)$  are the state and measurement at time  $k$ .  $\mathbf{f}$  is the state transition function,  $\mathbf{g}$  is the noise gain function, and  $\mathbf{h}$  is the measurement function. The process noise and measurement noises  $\mathbf{v}$  and  $\mathbf{w}$  are assumed to be zero mean, mutually independent, Gaussian white sequences with known covariance matrices  $\mathbf{Q}(k)$  and  $\mathbf{R}(k)$ , respectively. Both filters seek to create improved estimates of the target state at each time sample by forming a Bayesian weighted sum of the predicted target state and the most recent target measurement. The Bayesian weighting is done using the covariances of the state prediction and the measurement.

Kalman smoothers are a modification of Kalman filters that use future measurements to improve past estimates. They are less commonly used in tracking because they introduce a delay when operating in real time. Two smoothers are used in this study, a fixed-interval smoother which uses data from an entire track to estimate the state at each time sample, and a fixed-lag smoother with a lag of 2, which uses only two future measurements to estimate the state at each time sample. Details for implementing Kalman filters and EKF can be found in [2]. Details for implementing Kalman smoothers can be found in [17].

The motion model used for the Kalman filter and smoothers in this study is the piecewise constant white acceleration model of Bar-Shalom and Li [2], which assumes that target motion is independent along each Cartesian axis and can be modeled as constant velocity motion perturbed by accelerations which are constant over each sampling period. The accelerations enter the model by way of the process noise  $\mathbf{v}$ . Each Cartesian axis is processed independently with its own identical Kalman filter. The state vector for the  $\mathcal{E}$ -axis,  $\mathbf{x} = [\mathcal{E}, \dot{\mathcal{E}}]^T$ , consists of the position and velocity of the target. (The notation  $[\ ]^T$  represents the transpose of a matrix or vector.) The state transition, noise gain, and measurement functions are

$$\mathbf{f}(k, \mathbf{x}(k-1)) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \mathbf{x}(k-1) \quad (7)$$

$$\mathbf{g}(k, \mathbf{v}(k)) = \begin{bmatrix} T^2/2 & T \end{bmatrix}^T \mathbf{v}(k) \quad (8)$$

$$h(k, \mathbf{x}(k)) = [1 \ 0] \mathbf{x}(k) \quad (9)$$

As this model only estimates velocities along each Cartesian axis, the net horizontal swimming speed, heading, and turn rate were calculated as for point-wise differentiation using the velocities estimated by the filter using (2).

The tuning parameter for this filter is the standard deviation of the process noise acceleration,  $\mathbf{Q}(k) = \mathbf{Q} = [\dot{\mathcal{E}}]$ . The process noise for the two horizontal axes were kept identical in the tuning process, and the vertical process noise was varied separately. In our sensitivity studies, they were all swept over a range of values from 1 to 100 cm/s<sup>2</sup> in 1 cm/s<sup>2</sup> steps.

The EKF in this study uses a constant velocity coordinated turn model in the horizontal plane. A coordinated turn describes motion along a circular path with a constant turn rate. The particular coordinated turn model used is the polar velocity coordinated turn of Gustafsson and Isaksson [7], as they showed analytically that this implementation of the coordinated turn model has lower linearization error than a model based on Cartesian coordinate velocities. The polar velocity model is actually a target-oriented model, using the state vector  $\mathbf{x} = [\mathcal{E} \ \eta \ \theta \ s_h \ \omega]^T$ , where the states are the target's location in Cartesian coordinates  $(\mathcal{E}, \eta)$ , and the target's current heading  $\theta$ , speed  $s_h$ , and turn rate  $\omega$  respectively.

The state transition, noise gain, and measurement functions are

$$\mathbf{f}(k, \mathbf{x}(k-1)) = \begin{bmatrix} \mathcal{E} + \frac{2s_h}{\omega} \sin\left(\frac{\omega T}{2}\right) \cos\left(\theta + \frac{\omega T}{2}\right) \\ \eta + \frac{2s_h}{\omega} \sin\left(\frac{\omega T}{2}\right) \sin\left(\theta + \frac{\omega T}{2}\right) \\ \theta + \omega T \\ s_h \\ \omega \end{bmatrix} \quad (10)$$

$$\mathbf{g}(k, \mathbf{v}(k)) = \mathbf{v}(k) \quad (11)$$

$$\mathbf{h}(k, \mathbf{x}(k)) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}(k) \quad (12)$$

The tuning parameters for the EKF are the standard deviations of the process noise acceleration  $\mathbf{v} = [0 \ 0 \ 0 \ s_h \ \omega]$ . In our sensitivity studies, the horizontal acceleration process noise was swept from 0 to 50 cm/s<sup>2</sup> in 1 cm/s<sup>2</sup> steps, and the angular acceleration process noise was swept from 10°/s<sup>2</sup> to 100°/s<sup>2</sup> in 10°/s<sup>2</sup> steps.

### C. Segmenting Track Identifier

The Segmenting Track Identifier (STI) [19] is a non-Bayesian curve segmentation and fitting algorithm, originally presented in [14] that recursively develops a suboptimal, segmented, least-squares fit of a continuously acquired track to a parametric motion model. The goal of the algorithm is to partition the track into segments  $S_n$ ,  $n = 1 \dots N$ , with the motion of each segment described by a single parameter vector  $\mathbf{x}_n$ , such that the total fitting cost for the entire track is minimized. The total fitting cost consists of two elements. The first is the sum of the measurement residuals - the squared error between the track's measurements and the estimated positions calculated by the motion model, and the second is the sum of the knot costs. The term "knot" refers to a point where two segments intersect, and the knot costs are generated by a "continuity" function whose purpose is to impose desired continuity and smoothness constraints on the estimated motion across the entire track.

A flowchart describing the overall operation of the STI algorithm is presented Fig 4. It works by dividing the measurements into overlapping segments,  $S_n$ ,  $n = 1 \dots N$ . Associated with each segment  $S_n$  are the following:  $Z_n$ , the set of measurements assigned to the segment;  $L_n$ , a count of the number of assigned measurements;  $\mathbf{x}_n$ , a parameter vector that describes the parameters at the start of the segment; and  $\psi_{n,min}$ , a record of the lowest root mean squared fitting error achieved for the segment. The algorithm is initialized by assigning the first measurement as the start of the first segment,  $S_1$ , and setting the total number of segments  $N = 1$ . STI then proceeds in a two stage fashion. The first stage is the fit and segmentation state. As each new measurement becomes available, it is added to the current segment  $S_N$ , increasing,  $L_N$ , the number of measurements in the segment by one. After each new measurement is acquired, the parameter vector  $\mathbf{x}_N$  is estimated such that it minimizes the least squares cost function  $\chi_N$ ,

$$\chi_N = \|\mathbf{f}(\mathbf{x}_N, Z_N, S_{N-1})\|_2^2 \quad (13)$$

where  $\mathbf{f}(\mathbf{x}_N, Z_N, S_{N-1})$  is the vector valued function that calculates both the knot costs between  $S_N$  and  $S_{N-1}$  and the measurement residuals for  $Z_N$  for a given  $\mathbf{x}_N$ . Note that for segment  $S_1$ , the knot costs are identically zero as there is no previous segment  $S_0$ , and that no fitting is performed until the number of points assigned to a segment is equal to  $L_{min}$ , the minimum number of measurements required to fit the motion model. At the time of the fitting, the RMSE,  $\psi_N$ , is also calculated,

$$\psi_N = \sqrt{\chi_N / \gamma_f} \quad (14)$$

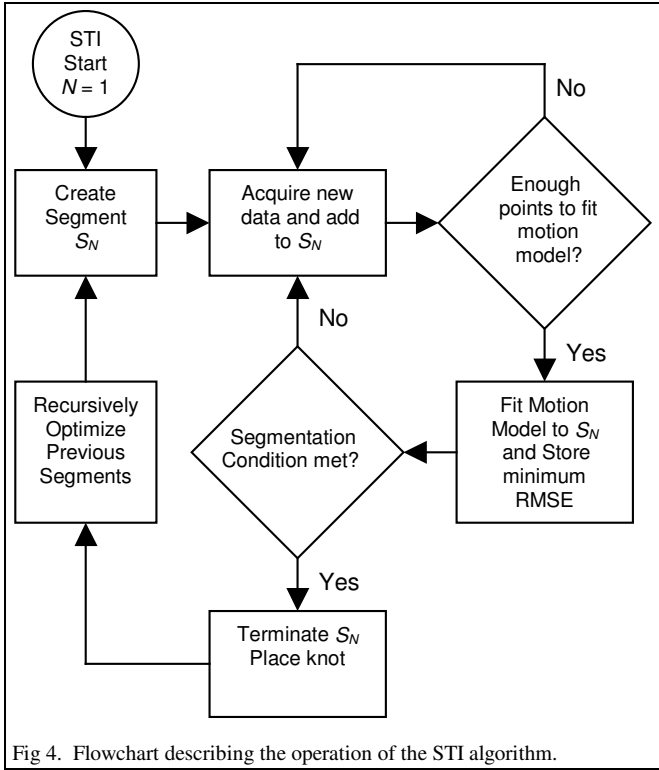


Fig 4. Flowchart describing the operation of the STI algorithm.

where  $\gamma_f$  is the length of the vector  $\mathbf{f}$ , which is equal to the dimension of the measurement vector times  $L_N$ , plus the dimension of the knot costs. Each time the segment is fit,  $\psi_{N,\min}$  is updated to be the lowest RMSE achieved for the segment (which is either the previous lowest value or the newest value if it is less than the previous low.)

This add and fit procedure is repeated for each new measurement until a segmentation condition is met. A segmentation condition occurs when any of the following are true:

$$\begin{aligned} \psi_N / \psi_{N,\min} &> \kappa_\psi \\ \psi_N &> \kappa_\sigma \sigma \\ \psi_{N,\tau} &> \kappa_\tau \sigma \end{aligned} \quad (15)$$

where  $\sigma$  is the measurement noise standard deviation, and  $\kappa_\psi$  and  $\kappa_\sigma$  are tuning parameters which determine the thresholds used for segmentation. The value  $\psi_{N,\tau}$  represents the root mean squared measurement residual for the last  $\tau$  measurements, and is helpful in detecting break conditions in long segments, where the poor fit from new measurements after a maneuver is obscured in  $\psi_N$  by the averaging across all the measurements of a long segment. The value  $\tau$  is also a tuning parameter of the algorithm.

The fit and segmentation stage ends once a segmentation condition has occurred. The segment  $S_N$  is terminated and the most recently added measurement, the one that caused the segmentation condition, is removed from the segment. A new segment,  $S_{N+1}$  is started, and the last measurement of  $S_N$  and the measurement which caused the segmentation condition are assigned as the first two measurements of  $S_{N+1}$ . The shared

measurement is the location of the knot between segments  $S_N$  and  $S_{N+1}$ . The algorithm then proceeds to the second stage, referred to as the recursive optimization stage.

The recursive optimization stage of the algorithm optimizes the fit of the four most recently completed segments. The optimization occurs for pairs of segments, starting with the fourth and third oldest segments. The data from the pair of segments being optimized is grouped together, and a search is performed to find the knot location that minimizes the total cost for both segments using (13). The segment pair with the lowest total cost then replaces the original segments. The optimization procedure is then repeated using the pair of segments consisting of the newly generated third oldest segment and the second oldest segment, and so on, until the most recently created segment has been reoptimized. After optimizing all segments, the algorithm returns to the fit and segmentation stage, adding new measurements to  $S_{N+1}$ .

#### 1) STI Motion Models and Knot Costs

Two motion models and associated knot costs were used with the STI for this study. One model for motion in the horizontal plane and the other model for vertical motion. The two filters were uncoupled from one another, allowing the horizontal and vertical data to be processed independently.

The motion model for the horizontal plane is a coordinated turn model, equivalent to that used for the EKF, with the parameter vector  $x_N = [\varepsilon_0 \quad \eta_0 \quad \theta_0 \quad s_h \quad \omega]^T$  consisting of the target's position and heading at the start of the segment, and the target's speed and turn rate (constant throughout the segment), respectively. The minimum number of measurements required to compute the fit is  $L_{\min} = 3$ . The measurement generating function  $\mathbf{m}(\mathbf{x}, k)$  used to derive the estimated position from the model is

$$\begin{aligned} \begin{bmatrix} \varepsilon(k) \\ \eta(k) \end{bmatrix} &= \begin{cases} \begin{bmatrix} \varepsilon_0 + \frac{2s_h}{\omega} \sin\left(\frac{\omega k}{2}\right) \cos\left(\theta_0 + \frac{\omega k}{2}\right) \\ \eta_0 + \frac{2s_h}{\omega} \sin\left(\frac{\omega k}{2}\right) \sin\left(\theta_0 + \frac{\omega k}{2}\right) \end{bmatrix}, & \omega \neq 0 \\ \begin{bmatrix} \varepsilon_0 + s_h k \cos(\theta_0) \\ \eta_0 + s_h k \sin(\theta_0) \end{bmatrix}, & \omega = 0 \end{cases} \end{aligned} \quad (16)$$

The continuity knot cost function  $\mathbf{c}(\mathbf{x}_N, S_{N-1})$  is

$$\begin{aligned} \mathbf{c}(\mathbf{x}_N, S_{N-1}) &= \begin{cases} \kappa_C (L_N + L_{N-1}) [\Delta_{\varepsilon,\eta} \quad \Delta_\theta]^T, & N > 1 \\ [0], & N = 1 \end{cases} \\ \Delta_{\varepsilon,\eta} &= \sqrt{[\mathbf{m}(\mathbf{x}_N, 0) - \mathbf{m}(\mathbf{x}_{N-1}, k_{N-1}(L_{N-1}))]^T \cdot [\mathbf{m}(\mathbf{x}_N, 0) - \mathbf{m}(\mathbf{x}_{N-1}, k_{N-1}(L_{N-1}))]} \\ \Delta_\theta &= \theta_N - (\theta_{N-1} + k_{N-1}(L_{N-1})\omega_{N-1}) \end{aligned} \quad (17)$$

where  $\kappa_C$  is the knot cost multiplier, a tuning factor that weights the importance of continuity in position and heading at the knots relative to the fit between the motion model and the measurements.  $\Delta_{\varepsilon,\eta}$  is the distance between the positions at the start of  $S_N$  and the end of  $S_{N-1}$ , and  $\Delta_\theta$  is the difference in heading, measured in radians, and ranging from 0 (same heading) to  $\pi$  (directly opposite heading). The  $(L_n + L_{n-1})$

factor scales the knot cost to the total length of the segments, preserving the relative cost of the knot for long segments.

The motional model for vertical motion is a constant velocity model, with the parameter vector  $\mathbf{x}_N = [\zeta_0 \quad \dot{\zeta}]^T$  consisting of the starting position of the segment and the velocity (constant throughout the segment), respectively. The minimum number of points required to compute the fit is  $L_{min} = 2$ . The measurement generating function  $m(\mathbf{x}, k)$  is

$$m(\mathbf{x}, k) = \zeta_0 + k \dot{\zeta} \quad (18)$$

The continuity knot cost function  $\mathbf{c}(\mathbf{x}_N, S_{N-1})$  is

$$\mathbf{c}(\mathbf{x}_N, S_{N-1}) = \begin{cases} \kappa_C (L_N + L_{N-1}) \Delta_\zeta, & N > 1 \\ 0, & N = 1 \end{cases} \quad (19)$$

$$\Delta_\zeta = m(\mathbf{x}_N, 0) - m(\mathbf{x}_{N-1}, k_{N-1}(L_{N-1}))$$

where  $\Delta_\zeta$  is the distance between the position at the start of segment  $S_n$  and the position at the end of  $S_{N-1}$ .

The cost function  $\mathbf{f}(\mathbf{x}_N, Z_N, S_{N-1})$  for both models is given by

$$\mathbf{f}(\mathbf{x}_N, Z_N, S_{N-1}) = \begin{bmatrix} \mathbf{c}(\mathbf{x}_N, S_{N-1}) \\ \mathbf{z}_N(1) - \mathbf{m}(\mathbf{x}_N, k_N(1)) \\ \dots \\ \mathbf{z}_N(L_N) - \mathbf{m}(\mathbf{x}_N, k_N(L_N)) \end{bmatrix} \quad (20)$$

where  $\mathbf{z}_N(a)$  is the  $a^{\text{th}}$  measurement of the set  $Z_n$ , and  $k_n(a)$  is the elapsed time of the same measurement relative to the first measurement in  $Z_n$ . (For example,  $k_n(1) = 0$ .)

## 2) STI Tuning

Unlike the Kalman algorithms, where tuning is done by varying the modeled process noise accelerations, the tuning of the STI algorithm is based on the measurement noise model, and not on the target behavior. As such, tuning the STI requires the correct choice of the measurement noise standard deviation,  $\sigma$ , used in the STI break condition computations of (15). The STI algorithm expects  $\sigma$  to be constant, but as the base sonar measurements are in spherical coordinates,  $\sigma$  is not constant in the Cartesian coordinates used in the STI motion models. Rather than modeling the changing measurement errors of each individual measurement, a constant representative value was chosen. The value used was that generated by a measurement which lies at the middle range value of the field of view, 337 cm, and at the middle of the angular limits,  $4^\circ$  in both bearing and elevation, on the basis that this represents approximately the midpoint of the Cartesian errors. The radial measurement error standard deviation,  $\sigma = \sqrt{\sigma_\epsilon^2 + \sigma_\eta^2} = 3.09$  cm was used for the horizontal filter, and the vertical measurement error standard deviation,  $\sigma = \sigma_\zeta = 3.26$  cm was used for the vertical filter.

In our sensitivity studies, the four STI tuning parameters were varied over the following ranges:  $\kappa_\sigma = [0.6 \ 2]$  in steps of 0.2;  $\kappa_\psi = 2, 3, 4$ ;  $\kappa_C = [0 \ 2]$  in steps of 0.2; and  $\tau = 1, 2, 3$  for the horizontal filter and  $\tau = 1, 2$  for the vertical filter.

## IV. RESULTS

The performance of the tracking algorithms and their

sensitivity to changes in their tuning parameters were evaluated using the previously discussed performance metrics. These metrics were plotted against all the combinations of each tracker's tuning parameters. Combinations of tuning parameters were generated using a series of nested *for-loops*. These graphs succinctly show both the peak performance of each algorithm, as well as its sensitivity to its tuning parameters. Please note that lower values represent better performance for RMSE and MAD statistics, while higher values represent better performance for the KS probabilities.

In terms of tuning parameter sensitivity response, the ideal insensitive response would produce a horizontal line across the graph for each metric. Undesirably high sensitivity produces either large total excursions or spikes in the graphs of the performance metrics.

Point-wise differentiation was used as the baseline against which the other algorithms were compared. Since it does not have tuning parameters, it is drawn as a bold horizontal bar on the position and speed RMSE and turn rate MAD graphs. The KS probabilities for point-wise differentiation are not plotted because they are so poor that they would obscure results within the generally accepted 0.01 and higher probability levels. Instead the desired minimum 0.01 probability level is indicated as a bold horizontal line.

All graphs are clipped to the same coordinates to facilitate comparison between algorithms; any portion of a graph that does not show a plotted value indicates that the error is off the chart, either too high for RMSE and MAD, or too low for KS probabilities. The worst-case performance for each algorithm is given in Table 1 for horizontal motion and Table 2 for vertical motion.

### A. Horizontal Motion Results

The horizontal trackers were evaluated using RMSE in horizontal speed (magnitude of total velocity in the horizontal plane), the two-sample KS probability for the horizontal speed distribution, turn rate MAD, the KS probability for turn rate magnitude distribution, and the root mean squared total radial position error in the horizontal plane. Fig. 5 presents the sensitivity plots for the Kalman filter and smoothers, Fig. 6 for the EKF, and Fig 7 for the STI.

#### 1) Kalman Filter / Smoother

The Kalman filters and smoothers perform much better than point-wise differentiation at estimating position, speed, and turn rate, and the Kalman smoothers achieve the lowest turn rate MAD of all the trackers tested. However, they are very sensitive to the process noise acceleration. The range of accelerations which produce the best speed estimates give poor position estimates; estimates so poor they have a larger RMSE error than the original measurements.

Smoothing improves performance with both smoothers outperforming the filter. The fixed-lag smoother performs better than the fixed-interval smoother, implying that it is important not to over-smoothing the track. While the smoothers are less sensitive than the filter, there is still only a small range of process noise levels that produce simultaneously good

Table I. Horizontal motion worst case results – Lower values are better for RMSE and MAD, higher values are better for KS Probabilities. Best results are in bold.

<i>Algorithm</i>	<i>Speed RMSE (cm/s)</i>	<i>Speed KS Prob.</i>	<i>Turn Rate MAD (°/s)</i>	<i>Turn Rate KS Prob.</i>	<i>Position RMSE (cm/s)</i>
<i>Point-wise Differentiation</i>	10.58	6.58e-8	63.29	2.93e-18	<b>3.04</b>
<i>Kalman Filter</i>	12.90	4.23e-13	52.49	4.59e-51	9.31
<i>Fixed-Lag Smoother</i>	9.50	1.90e-6	<b>29.47</b>	2.46e-30	5.92
<i>Fixed-Interval Smoother</i>	9.56	2.62e-11	28.16	2.92e-142	9.04
<i>EKF</i>	12.63	3.05e-8	41.78	6.67e-20	4.49
<i>STI</i>	<b>6.25</b>	<b>0.012</b>	32.24	<b>3.40e-15</b>	3.10

estimates of both speed and turn rate distribution, as shown by the graph of the KS metrics. Thus good performance requires selection of the process noise from a small window, with the fixed-lag smoother providing the largest window from approximately 30 to 50 cm/s<sup>2</sup>.

### 2) Coordinated turn EKF

The EKF using a coordinated turn model performs poorly both with respect to stability and overall performance. The speed RMSE is poor for all combinations of tuning parameters, often worse than that of point-wise differentiation. As with the Kalman filter-based trackers, the range of parameters that produce the best speed estimates also give poor position estimates. Turn rate distribution, as judged by the KS probability, is good when the filter is tuned properly.

### 3) STI

The STI algorithm performs well and displays low sensitivity, being the only algorithm that always out-performs point-wise differentiation in all speed and turn rate performance metrics. It achieves the lowest speed RMSE and is least sensitive to its tuning parameters for all but the turn rate KS test. Although the variance of the turn rate KS probability with tuning parameters is pronounced, it is not quite as bad as it seems from the turn rate KS plot in Fig. 7; to achieve reasonable performance one need only select either the knot continuity cost,  $\kappa_C$ , or the noise segmentation threshold,  $\kappa_\sigma$ , correctly. If either is chosen correctly the variance with the remaining tuning parameters is relatively small and performance is good. If the standard deviation of the measurement noise is known,  $\kappa_\sigma$  does not need to be tuned as  $\kappa_\sigma = 1$  (representing the known, correct value of the measurement noise) always gives good results. Values of  $\kappa_C$  at or slightly below 1 also always produced good results.

## B. Vertical Motion Results

The vertical trackers were evaluated using RMSE in vertical velocity, the two-sample KS probability for the vertical speed distribution, and the root mean squared position error in the vertical plane. Velocity was used in place of speed for the vertical RMSE tests as it is an easy substitution for one-

Table II. Vertical motion worst case results – Lower values are better for RMSE, and higher values are better for KS Probabilities. Best results are in bold.

<i>Algorithm</i>	<i>Vel. RMSE (cm/s)</i>	<i>Speed KS Prob.</i>	<i>Position RMSE (cm/s)</i>
<i>Point-wise Differentiation</i>	14.78	7.76e-29	3.26
<i>Kalman Filter</i>	10.36	1.86e-21	3.45
<i>Fixed-Lag Smoother</i>	<b>6.74</b>	<b>6.58e-8</b>	<b>2.75</b>
<i>Fixed-Interval Smoother</i>	6.77	5.08e-10	3.11
<i>STI</i>	8.17	1.31e-16	2.97

dimensional motion and is more demanding given that the vertical velocities were small and roughly centered around zero. Table I presents the sensitivity plots for the Kalman filter and smoothers, and Table II for the STI.

### 1) Kalman Filter / Smoother

The Kalman filter and smoothers perform well on the vertical motion tests, out-performing point-wise differentiation and the STI. The problem of increased position RMSE present in the horizontal motion tests is now essentially gone. While the vertical speed KS test still showed sensitivity to the choice of process noise, both smoothers and the filter are effective over a reasonable range of acceleration values given the biological constraints of fish motion in the vertical direction observed here. The smoothers again out-perform the filter in terms of position and velocity RMSE, but they do tend to over-smooth the speeds if the process noise is too small, especially the fixed-interval smoother.

### 2) STI

The STI algorithm does not perform well on the vertical motion tests, with generally higher velocity RMSE than the Kalman filters and smoothers for most selections of tuning parameters. Additionally it shows high sensitivity to the choice of tuning parameters, especially with respect to the speed KS metric. Selection of the knot cost,  $\kappa_C$ , at or slightly above 1 generally provides acceptable results, but outside of that range it is very dependent on the remaining tuning parameters.

## V. DISCUSSION

Although no algorithm performs flawlessly at tracking both the horizontal and vertical motion of fish, point-wise differentiation is clearly not the best choice. The STI performs best tracking the horizontal motion, doing well on all metrics, while Kalman filters and smoothers do best tracking the vertical motion. Understanding why each algorithm fails in each situation can lead to a means of choosing an appropriate algorithm for a given field study.

The horizontal fish motion in this study is characterized by many essentially unpredictable abrupt turns, while the vertical

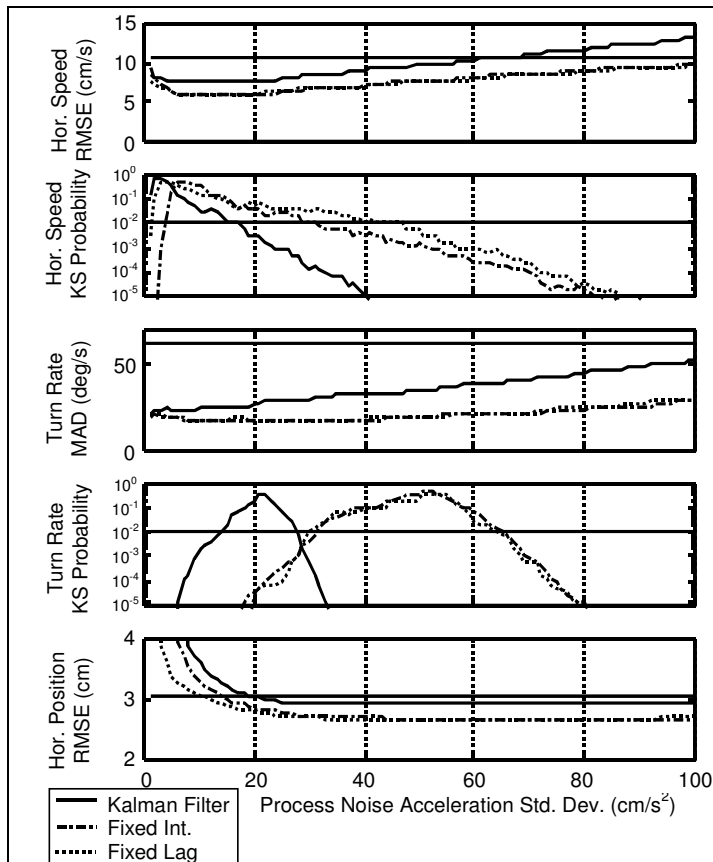


Fig. 5. Horizontal motion performance and tuning parameter sensitivity of Kalman filter and smoother. The algorithms' only tuning parameter is the process noise acceleration standard deviation. The horizontal bars represent the baseline performance for RMSE and MAD statistics (from point-wise differentiation) and KS probability levels ( $p = 0.01$ ). Lower values are better for RMSE and MAD, while higher values are better for KS probabilities.

motion is characterized by relatively predictable straight-line motion. Kalman filters are predictive model-based algorithms; they assume that the motion model is correct and that the state must evolve according to that model. The new state estimate is formed as a weighted sum of the new measurement and the prediction provided by the model's state transition matrix, with the weighting of these two elements essentially determined by the model's process and measurement noises. In effect the model guides the algorithm, and assuming the model is correct this guidance allows the filter to provide a better estimate of the state than using the noisy measurements alone. However, if the model is incorrect, the Kalman filters will tend to perform poorly. The vertical motion in this study is easy to predict and well modeled by the straight-line motion model of the Kalman algorithms, and thus the Kalman algorithms perform well for the vertical motion. In contrast, the abrupt and varied horizontal fish motion is difficult to model and predict, and is thus difficult to track using a Kalman algorithm. The algorithms must model the unknown accelerations as Gaussian distributed process noise. If the modeled process noise is set too small, measurements occurring after sudden large maneuvers will be ignored, and the state estimate will lag the actual behavior of the target. If the process noise is set too

large, most of the emphasis in the update will be given to the measurement and little actual filtering will be performed.

STI is a data driven algorithm that does not use its motion model in a predictive fashion; rather the model is used to constrain the plausible motion for each segment, with segmentation occurring when a single segment can no longer adequately describe target motion. A segment's description of target motion is deemed inadequate when the segment's fitting error is larger than expected given the measurement noise. STI achieves improved state estimates by performing curve fitting on the data from an entire segment, which is hypothesized to represent a single maneuver. No random process noise is required to model unknown maneuvers because maneuvers are accounted for by the segmenting rather than the fitting portion of the algorithm.

If the motion model is chosen correctly and the measurement noise is understood, STI will likely perform well when there are a relatively large number of measurements per segment or the measurement noise is "small". However, if the measurement noise is large, and there are few measurements per segment, STI will perform poorly. The low sampling rate of the sonar data in this study creates a situation where there are generally few data points per maneuver, making STI's performance dependent on the measurement noise level. The STI performs well on the horizontal motion tests, indicating the model is correct and the measurement noise is "small". However, the STI performs poorly on the vertical motion, so in this case either the model is incorrect or the measurement noise is too large. As the STI vertical motion model is similar to that used by the Kalman algorithms, and the Kalman algorithms perform well, the model is apparently correct and the problem must lie in the size of the measurement noise. A third explanation would be that the vertical motion doesn't have abrupt changes, and thus segmentation is not required, but does occur because of the higher measurement noise levels.

In order to quantify the effect of measurement noise on STI performance, we define the Noise to Motion Ratio (NMR) of a data set as the ratio of measurement noise standard deviation to the average distance traveled between measurements. Using the average speeds obtained from the video data and the sonar's sampling rate, this study has horizontal and vertical NMRs of 0.45 and 3.23 respectively. Because the measurement noise in the vertical direction is more than three-times greater than the mean distance between measurements, dominating the measurements, the STI algorithm has difficulty segmenting the track accurately.

An additional study was run on simulations of the vertical motion data to better understand STI's sensitivity to NMR. Noisy sonar measurements were simulated by adding zero-mean Gaussian noise of known standard deviation to the ground truth video data, with the noise standard deviation chosen to give NMR of 0.2 to 3 in steps of 0.2. STI's

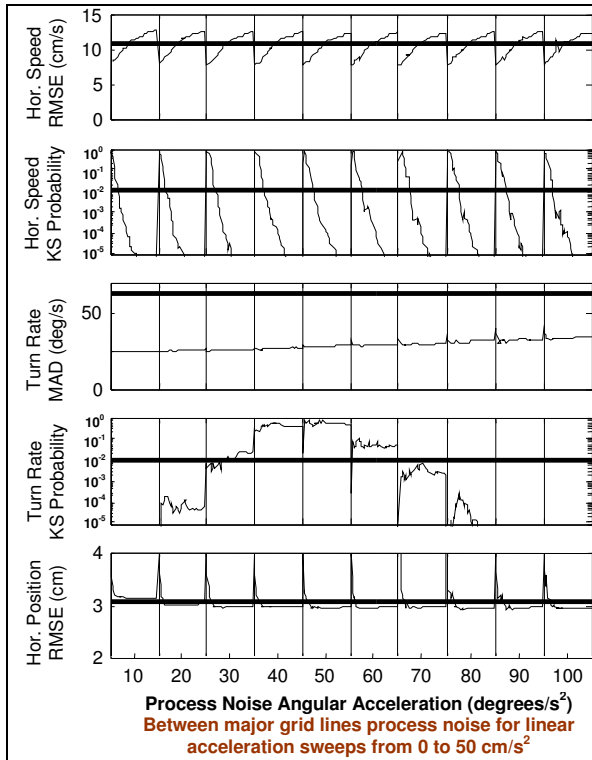


Fig. 6. Horizontal motion performance and tuning parameter sensitivity for EKF. All combinations of tuning parameters were iteratively produced by first varying angular acceleration process noise, and then the tangential acceleration process noise.

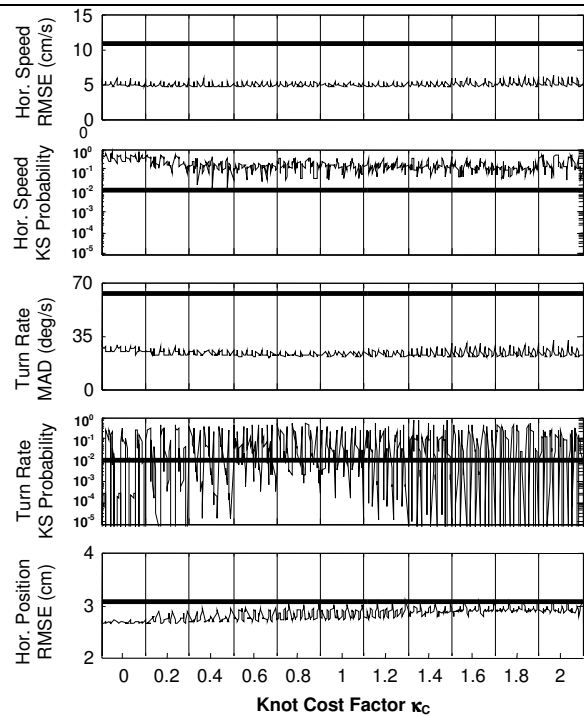


Fig. 7. Horizontal motion performance and tuning parameter sensitivity for STI. All combinations of tuning parameters were iteratively produced by first varying  $\kappa_c$ , then  $\tau$ ,  $\kappa_\psi$ , and  $\kappa_\sigma$ .

sensitivity to tail length  $\tau$  and RMSE ratio break level  $\kappa_\psi$  are graphed as a function of NMR in Fig. 10. The knot cost tuning parameter  $\kappa_c$  and the noise level tuning parameter  $\kappa_\sigma$  were both set to 1. The plots show that the STI performs well on all performance metrics until the NMR > 1.

With these results rules for choosing a Kalman algorithm or the STI can be provided. If the NMR < 1, one can be confident that STI characterizes motion well and the STI is a good choice. An estimate of NMR for an experimental setup can be calculated as the sensor measurement noise covariance divided by the product of the average speed given by point-wise differentiation and the sampling rate. Using this method for the data in this experiment, NMR = 0.44 was estimated for the horizontal case, clearly indicating that the STI should work well, and NMR = 1.16 for the vertical motion, indicating a situation where the STI performance would be degraded.

The applicability of the Kalman filter and smoothers can be ascertained by visual inspection of the experimental data. If the tracks are piecewise linear for many measurements with sudden turns occurring rarely in the data set, a Kalman filter or smoother will likely work well. In the absence of real time requirements, the superior performance of a fixed-lag smoother makes it preferred. The lag should be kept relatively small as increasing the lag increases the tendency to over-smooth.

## VI. CONCLUSION

Point-wise differentiation is not an effective means of estimating the motion of free-swimming fish, and alternative methods are required. The tracking algorithms used in this paper can provide quality estimates of fish motion when used appropriately. The overall recommendation is that if the ratio of measurement noise to the average distance traveled between measurements (termed here as the NMR) is small, the STI algorithm is an excellent choice, as it works well and requires little tuning. If the NMR is not small and the motion is reasonably approximated by a straight line, as might be the case for actively migrating riverine fish, a fixed-lag Kalman smoother is the best choice. While the STI algorithm performs best when the NMR is small, it should be noted that in this experiment the STI algorithm always provided better estimates of swimming speed and turn rates than point-wise differentiation regardless of the choice of tuning parameters, even in the vertical motion case where the NMR was large.

The use of tracking algorithms to estimate fish motion is also indicated because it lends itself to the automatic tracking of multiple fish. Several data association algorithms exist that are designed to work with tracking algorithms to either simultaneously track multiple fish or to track a single fish in measurement clutter. Bar-Shalom and Li [3] provide several examples of such algorithms for use with Kalman filters, and a method of using the STI in these frameworks can be found in

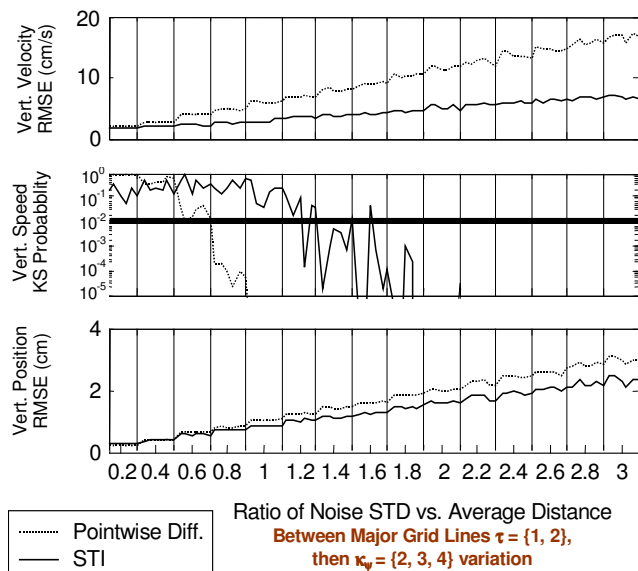


Fig. 8. The performance of STI and with respect to Noise to Motion Ratio (NMR) is shown for simulated vertical motion data. Point-wise differentiation performance using the same simulated data is also plotted as a baseline. The STI algorithm clearly becomes ineffective as measured by the KS probability metric when  $NMR > 1$ .

[18, 19].

The STI algorithm has one additional possible advantage not previously mentioned. Its segmented output provides a natural breakdown of fish motion into individual maneuvers. This provides a description not only of swimming speeds and turn rates, but also maneuver length information. This additional information might be helpful in better understanding fish behavior. For example, frequent short maneuvers might represent a situation of higher activity and corresponding higher energy expenditure.

On a final note, if the motion under study is complicated enough to be essentially unpredictable by a simple model, and the measurement errors are large, it seems unlikely that accurate information can be generated by any algorithm. A better sensor is likely required in these situations.

#### ACKNOWLEDGMENT

This material is based in part upon work supported under a National Science Foundation Graduate Fellowship. We would also like to acknowledge the invaluable support of Dr. Jules S. Jaffe of Scripps Institution of Oceanography. Dr. Jaffe as role of thesis advisor provided support and lab resources to obtain the experimental data analyzed herein.

#### REFERENCES

- [1] F. Arrhenius, B. J. A. M. Benneheij, L. G. Rudstam, and D. Boisclair, "Can stationary bottom split-beam hydroacoustics be used to measure fish swimming speed in situ?," *Fisheries Research (Amsterdam)*, vol. 45, pp. 31-41, 2000.
- [2] Y. Bar-Shalom and X.-R. Li, *Estimation and Tracking: Principles Techniques, and Software*. Storrs, CT: YBS, 1998.
- [3] Y. Bar-Shalom and X.-R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*. Storrs, CT: YBS, 1995.

- [4] D. Boisclair, "An evaluation of the stereocinematographic method to estimate fish swimming speed," *Canadian Journal of Fisheries and Aquatic Sciences*, pp. 523-531, 1992.
- [5] D. Boisclair, "Relationship between feeding and activity rates for actively foraging juvenile brook trout (*Salvelinus fontinalis*)," *Canadian Journal of Fisheries and Aquatic Sciences*, vol. 49, pp. 2566-2573, 1992.
- [6] D. Boisclair and M. Tang, "Empirical analysis of the influence of swimming pattern on the net energetic cost of swimming in fishes," *Journal of Fish Biology*, vol. 42, pp. 169-183, 1993.
- [7] F. Gustafsson and A. J. Isaksson, "Best choice of coordinate system for tracking coordinated turns," presented at Proc. of the 35th IEEE Conference on Decision and Control, Kobe, Japan, 1996.
- [8] D. G. Harper and R. W. Blake, "A Critical Analysis of the Use of High-Speed Film To Determine Maximum Accelerations of Fish," *Journal of Experimental Biology*, pp. 465-472, 1989.
- [9] N. F. Hughes and L. H. Kelly, "New techniques for 3-D video tracking of fish swimming movements in still or flowing water," *Canadian Journal of Fisheries and Aquatic Sciences*, pp. 2473-2483, 1996.
- [10] J. S. Jaffe, "Target localization for a three-dimensional multibeam sonar imaging system," *Journal of the Acoustical Society of America*, vol. 105, pp. 3168-3175, 1999.
- [11] J. S. Jaffe, E. Reuss, D. McGehee, and G. Chandran, "FTV: A sonar for tracking macrozooplankton in three dimensions," *Deep-Sea Research, Part A: Oceanographic Research Papers*, vol. 42, pp. 1495-1512, 1995.
- [12] R. L. Johnson and R. A. Moursund, "Evaluation of juvenile salmon behavior at Bonneville Dam, Columbia River, using a multibeam technique," *Aquatic Living Resources*, vol. 13, pp. 313-318, 2000.
- [13] M. M. Krohn and D. Boisclair, "Use of a stereo-video system to estimate the energy expenditure of free-swimming fish," *Canadian Journal of Fisheries and Aquatic Sciences*, vol. 51, pp. 1119-1127, 1994.
- [14] S. P. Linder, M. D. Ryan, and R. J. Quintin, "Concise Track Characterization of Maneuvering Targets," presented at AIAA Conference on Guidance, Navigation, and Control, Montreal, CA, 2001.
- [15] L. Mo, X. Song, Y. Zhou, K. Sun Zhong, and Y. Bar-Shalom, "Unbiased converted measurements for tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, pp. 1023-1027, 1998.
- [16] T. J. Mulligan and D. G. Chen, "Comment on 'Can stationary bottom split-beam hydroacoustics be used to measure fish swimming speed in situ?' by Arrhenius et al," *Fisheries Research (Amsterdam)*, vol. 49, pp. 93-96, 2000.
- [17] S. Roweis and Z. Ghahramani, "A unifying review of linear gaussian models," *Neural Computation*, vol. 11, pp. 305-345, 1999.
- [18] C. Schell, "Advanced Tracking Algorithms for the Study of Fine Scale Fish Behavior," in *Electrical and Computer Engineering*. La Jolla: University of California, San Diego, 2003.
- [19] C. Schell, S. P. Linder, and J. R. Zeider, "Tracking highly maneuverable targets with unknown behavior," *Proceedings of the IEEE*, vol. 92, pp. 558-574, 2004.
- [20] M. Tang and D. Boisclair, "Relationship between respiration rate of juvenile brook trout (*Salvelinus fontinalis*), water temperature, and swimming characteristics," *Canadian Journal of Fisheries and Aquatic Sciences*, vol. 52, pp. 2138-2145, 1995.
- [21] M. Tang, D. Boisclair, C. Menard, and J. A. Downing, "Influence of body weight, swimming characteristics, and water temperature on the cost of swimming in brook trout (*Salvelinus fontinalis*)," *Canadian Journal of Fisheries and Aquatic Sciences*, vol. 57, pp. 1482-1488, 2000.
- [22] P. W. Webb, "Composition and Mechanics of Routine Swimming of Rainbow Trout *Oncorhynchus-Mykiss*," *Canadian Journal of Fisheries and Aquatic Sciences*, vol. 48, pp. 583-590, 1991.



**Chad Schell** (Member, IEEE) received the B.S. degree in electrical engineering from the University of New Mexico, Albuquerque, in 1996, and the M.S. and Ph.D. degrees in electrical engineering (applied ocean science) from the University of California, San Diego, La Jolla, in 2000 and 2003 respectively. From 1991 to 1996, he worked at Sandia National Laboratories in the Intelligent Systems and Robotics

and the Microsensors divisions. He currently conducts research on geolocation systems at Rincon Research Corporation in Tucson, AZ, and operates his own consumer electronics business. Dr. Schell received a National Science Foundation fellowship.



**Stephen Paul Linder** received the B.S. degree in mechanical engineering from the Massachusetts Institute of Technology, Cambridge, in 1982 and the M.S. degree in computer systems engineering and the Ph.D. degree in electrical and computer systems engineering from Northeastern University, Boston, MA, in 1996 and 1998, respectively. He was with the Applied Research Laboratory, Pennsylvania State University, University Park, working on target tracking and sensor data fusion, and a Faculty Member in computer science at the State University of New York, Plattsburgh. He is currently in the Computer Science Department, Dartmouth College, Hanover, NH, and is working on projects that include the track prediction of bouncing ball using limited data rate cameras, a rock climbing robot and dynamic health assessment of first responders. He also teaches classes in mountaineering or sea kayaking when not sitting in front of the computer. See <http://alum.mit.edu/www/spl> for more details.