

# A NON-BAYESIAN SEGMENTING TRACKER FOR HIGHLY MANEUVERING TARGETS

**Stephen Paul Linder\***  
spl@alum.mit.edu

**Chad Schell†**  
chad@schells.com

## Abstract

The Segmenting Track Identifier (STI) is introduced as a new methodology to tracking highly maneuvering targets. This non-Bayesian approach dynamically partitions a target track into a sequence of track segments, making hard estimates of when the target's maneuvering mode transitions occur, and then estimates the parameters of the target model for each segment. STI is compared to two VS IMM algorithms through simulations, where it is shown to have a three fold performance advantage in median absolute turn rate estimation errors, as well as better position estimation for very highly maneuvering targets. STI performance is also shown to increase substantially if small delays in output can be tolerated.

**Keywords:** target tracking, hybrid systems, curve fitting, IMM trackers, track characterization

## I. INTRODUCTION

The motion of a maneuvering target is often modeled as a hybrid system whose continuously varying state evolves according to an underlying model that performs discrete jumps between different operating modes. Tracking of such targets is often accomplished using a Multiple Model (MM) method that models the hybrid nature of the target dynamics by using a bank of (extended) Kalman filters running in parallel to estimate target motion. Each mode of the target is modeled with a separate filter, and the jump process between modes is modeled as a Markov process with known mode transition probabilities. The final estimate of target motion consists of a Bayesian weighted sum of

---

\* Currently at SUNY Plattsburgh, Computer Science, 101 Broad Street., Plattsburgh, NY 12901

† Currently at University of California at San Diego and Scripps Institution of Oceanography, San Diego, CA

the mode matched estimates proportional to the likelihood of each mode (Bar-Shalom and Li 1998.) Effective computationally efficient MM methods, such as the Interacting Multiple Model (IMM) method, are well established for slowly or moderately maneuvering targets. However, when the target is highly maneuvering the required bank of filters can become large, reducing the effectiveness of the tracker because the proper mode estimate is obscured by the unlikely mode estimates in the weighted sum. Researchers have recently developed adaptive MM algorithms to overcome this problem, including Variable Structure (VS) IMM filters that continuously adapt a small model set to the dynamics of the target.

This paper presents an alternative approach to tracking highly maneuvering targets, the Segmenting Track Identifier (STI), which does not rely on the Bayesian mixing of a bank of filters. Instead, the STI segments the track, making *hard* estimates of when mode transitions occur, and then determines the optimal model parameters, in the least square sense, of the target motion for each segment. Accurate segmentation of the track allows more accurate estimation of the target model, and subsequently improved estimates of the target state. This method is similar to methodologies used in the pattern recognition community to parameterize complex curves (West and Rosin 1991; Lim, Xin and Hong 1995; Wenyin 1998; Sheu and Hu 1999; Pittman and Murthy 2000). When a small delay in estimating target motion is acceptable, estimation accuracy can be improved by using the STI tracker as fixed lag smoother. As the STI is inherently a smoothing algorithm, no modifications are required to obtain these additional smoothed estimates. Unlike smoothing methodologies proposed for IMM trackers (Chen and Tugnait 2000) the additional computational cost of smoothing is constant time for all magnitudes of lag intervals.

Section II of this paper presents background on MM algorithms and the STI algorithm and how they differ from a hybrid systems point of view. Section III presents a detailed description of the STI algorithm and a brief description of two VS IMM algorithms chosen as top performers from a recent survey article (Semerdjiev, Mihaylova and Li 2000). Section IV presents simulation results comparing the performance of the STI, and the two VS IMM algorithms, and a constant velocity Kalman smoother on a track scenario used in the same survey article and on a second scenario using a more highly maneuvering target.

## II. BACKGROUND

The motion of maneuvering targets can be characterized by a continuously varying base state which evolves according to an underlying modal state that switches modes in discrete jumps. This is referred to as a hybrid system because it has both continuous and discrete elements. How these systems are represented by various algorithms is the subject of this section.

### Multiple Model Algorithms

In the Kalman filter-based MM approach the target is assumed to follow the jump linear dynamic system equations

$$\mathbf{x}(k) = \mathbf{F}_k \mathbf{x}(k-1) + \mathbf{G}_k \mathbf{v}_k$$

$$\mathbf{z}(k) = \mathbf{H}_k \mathbf{x}(k) + w_k$$

where  $\mathbf{x}(k)$  is the continuous valued base state at time  $k$  and  $\mathbf{z}(k)$  is the available measurement of this state. The modal state at time  $k$  is given by  $M_k = \{\mathbf{F}_k, \mathbf{G}_k, \mathbf{H}_k, \mathbf{v}_k, w_k\}$ , where  $\mathbf{F}_k$ ,  $\mathbf{G}_k$  and  $\mathbf{H}_k$  are the state transition, input, and measurement matrices, and  $\mathbf{v}_k$  and  $w_k$  are the process and measurement noise processes associated with the model in effect at time  $k$ . For a typical Kalman filter-based MM filter,  $\mathbf{v}_k$  and  $w_k$  are orthogonal, zero-mean, white Gaussian processes with known covariances  $\mathbf{Q}_k$  and  $\mathbf{R}_k$  respectively.

In the traditional fixed structure MM approach,  $M_k$  is assumed to be constrained to one of a fixed finite set of  $r$  possible models such that  $M_k \in \{M_j\}_{j=1}^r$ . It is also assumed that the transitions in  $M_k$  occur as discrete jumps governed by a first-order homogeneous Markov process with known mode transition probabilities. The actual modal state estimate in MM algorithms is represented by the mode probabilities of the models  $M_j$ ,  $\mu_j(k) = P\{M_j | Z\}$ , where  $Z$  is the set of measurements up to time  $k$ . The base state estimate  $x(k)$  is formed by combining the mode-conditioned base state estimates available from each possible mode sequence using Bayes theorem. The optimal implementation of a MM tracker maintains all possible mode histories, whose numbers increase exponentially with time. Therefore, suboptimal methods are usually used to bound the computational complexity of the algorithm by limiting the number of mode histories, with the most popular of these methods being the highly efficient IMM algorithm.

These fixed structure MM methods, including the IMM algorithm, work well when a target's maneuvers can be characterized by a small set of operating modes. However a highly maneuvering target's modal state might cover a large continuous range of parameter values that cannot be adequately expressed by a compact finite set of models. In these situations the fixed structure MM algorithms can perform poorly. To address this problem, newer variants of IMM trackers have been created which use an adaptive rather than a fixed set of models. Two examples of these methods are the AGIMM (Jilkov, Angelova and Semerdjiev 1999) and VS-AIMM (Semerdjiev, Mihaylova and Li 2000) algorithms, both of which replace the fixed, discrete set of models with a small set of models whose parameters are continuously adapted to the maneuvers of the target.

### **Curve Segmentation and Fitting**

An alternative approach to MM-based methodologies is the use of piecewise curve segmentation and fitting to represent a target track. Rather than using a Bayesian framework to describe the modal state of a target, it is possible to make a hard estimate of when the modal state transitions occur. A piecewise functional description of a noisy data set can then be created by partitioning the data into segments, with each segment fit to a simple parametric function, such as a straight line, parabola, or circular arc (Pittman and Murthy 2000; Chen, Zhang, Ou and Feng 2003). Yoshimito (Yoshimoto 1999) characterizes this segmentation as a nonlinear optimization problem that (1) optimizes the fit to the original data, (2) optimizes the segmentation and (3) minimizes the number of segments. The resulting segments are joined together at knots such that they meet certain end point conditions. A common technique of this type is the use of splines to piecewise represent a curve, where the end point requirements are chosen such that the piecewise function be continuous and smooth at the knots. Smoothness is typically represented by continuity in the derivatives of the fitting function up to a chosen order.

The segmentation of curves is the topic of multiple pattern recognition papers [(West and Rosin 1991); Saga, 1995 #3; Sheu, 1999 #7; Lim, 1995 #13]. Although these algorithms are designed to work with relatively noise free data and a large number of data points, similar concepts can be employed in an algorithm that can work with noisy data or a small number of points per curve (although not typically both.) One such algorithm is presented in this paper, the Segmenting Track

Identifier (STI), which achieves these goals by using recursive segmentation and enforcing continuity conditions at the knots.

In the STI approach the target is assumed to follow the jump-dynamic system of equations

$$\mathbf{x}(k) = f[\mathbf{x}(k-1)]$$

$$\mathbf{z}(k) = h[\mathbf{x}(k)] + w_k$$

where  $\mathbf{x}(k)$  is the base state at time  $k$  and  $\mathbf{z}(k)$  is the available measurement of this state. The modal state at time  $k$  is given by  $\mathbf{M} = \{f, h, w\}$  where  $f$ , and  $h$  are the state transition and measurement functions, and  $w$  is the associated measurement noise process associated with the model. STI does not assume a linear system, nor does it include a process noise as part of its model. It is assumed that  $w$  is a zero-mean random process with known covariance matrix  $\mathbf{R}$ , but it is not required that  $w$  be Gaussian. (Although correspondingly there is also no explicit statement that the least square fitting criterion used is optimal.)

Although the systems of equations used in the MM and STI algorithms are similar, their treatment of the modal state and the relationship between the modal and base states is very different. The MM algorithms represent the modal state as the probabilities that a given mode is the true mode at a given time, while the STI algorithm gives an explicit estimate of the modal state in effect at each time  $k$ , as well as the temporal sequence of these explicit modal transitions. However, the difference between the two approaches is more fundamental in that the MM algorithms and their underlying Kalman filters are predictive model-based algorithms while the STI algorithm is a data-driven, curve-fitting algorithm. MM algorithms assume that the models and their probabilities currently in effect are correct and that the base state must evolve according to those models. Each mode-conditioned base state estimate is formed as a weighted sum of the new measurement and the prediction provided by the model's transition matrix, with the weighting of these two elements 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 base state than the noisy measurements alone. If the models are incorrect, or the modal transitions are too rapid to be readily modeled in a predictive manner, the MM algorithms, including the adaptive ones, will tend to perform poorly.

In contrast, the STI algorithm does not use the current modal state to guide the filter, but rather it uses the data to improve the determination of the modal state. The actual “model” provided to the STI algorithm is the parameter space on which the STI’s curve fitting algorithm is free to operate. STI achieves improved estimates by using data from an entire segment to estimate the current modal state, performing parameter estimation over a portion of the track where the modal state is hypothesized to be constant. If the segmentation and the model parameter space is correct, and either there are a “large” number of measurements per segment or the noise levels are low, STI will provide a good estimate of the base and modal states. However, if the noise levels are high and there are few measurements per segment, STI will perform poorly even if the model parameter space is chosen correctly, because STI will follow the noise.

### III. THE ALGORITHMS

This section presents a description of the STI algorithm and two variable structure IMM algorithms: the AGIMM and the VS-AIMM.

#### STI Algorithm

The **Segmenting Track Identifier** (STI) algorithm is a real-time algorithm that recursively develops a suboptimal, segmented least squares fit of a continuously acquired track. The track is segmented into a series of  $N$  overlapping segments,  $S_1$  to  $S_N$  joined by knots  $\mathbf{X}_1$  to  $\mathbf{X}_{N-1}$ . A state vector  $\mathbf{X}_n$  is associated with each segment  $S_n$  where  $\mathbf{X}_n$  represents the parameters at the beginning the segment under a given functional model  $M$ .  $M$  is the functional model for all segments and represents the parameter space over which the curve fitting is done.

After STI fits the initial segment to the target track the algorithm iterates between two stages. The first stage produces an initial fit with the incoming data and determines when a new segment is necessary, while the second stage reoptimizes the fit and segmentation of the previous segments. Figure 1 gives a high level flow chart of the algorithm, and the following sections give a description.

## Initialization

A track is initialized by estimating the parameters of the first segment,  $S_1$ , after the number of measurements equals the minimum segment length  $L_{min}$ .  $L_{min}$  is a tuning parameter whose minimum value is determined by the minimum number of data points required to perform a fit to  $M$ . For example, a minimum of three measurements is required by a circular arc model, but a larger number of measurements can be used to increase the robustness of the fit.

As each new measurement is acquired it is added to  $S_1$ , and the fit is recalculated until a segment break condition is reached. Two break conditions are used for this paper. The first break condition occurs when the most recent  $L_h$  measurements no longer have a good fit to the model, defined as the point when the ratio of the mean residual of those points to the mean residual of all prior points in the segment exceeds  $K_\Delta$ . The second break condition occurs when the RMS residual for the segment exceeds by the factor  $K_{RMS}$  the standard deviation of the measurement noise as defined by  $M$ . The tuning parameters  $K_\Delta$  and  $K_{RMS}$  are adjusted to produce a parsimonious segmentation of the track, while still allowing enough sensitivity to detect jump transitions.

Once a break condition is detected,  $S_1$  is terminated at the measurement prior to the break. A new segment,  $S_2$  is initiated using the last measurement of  $S_1$  and the new measurement which caused the break condition. The shared measurement is the location of the knot  $X_j$ . The algorithm then proceeds to the *Fit and Segmentation Stage*.

## Fit and Segmentation

The *Fit and Segmentation Stage* of the algorithm proceeds as with the *Initialization Stage*, except the fitness function is augmented with the continuity cost between the current and previous segment. The cost function for continuity varies with application, but for this paper consists of both position and heading continuity costs. When the segment break condition is met, rather than creating a new segment the algorithm proceeds to the *Recursive Optimization Stage*.

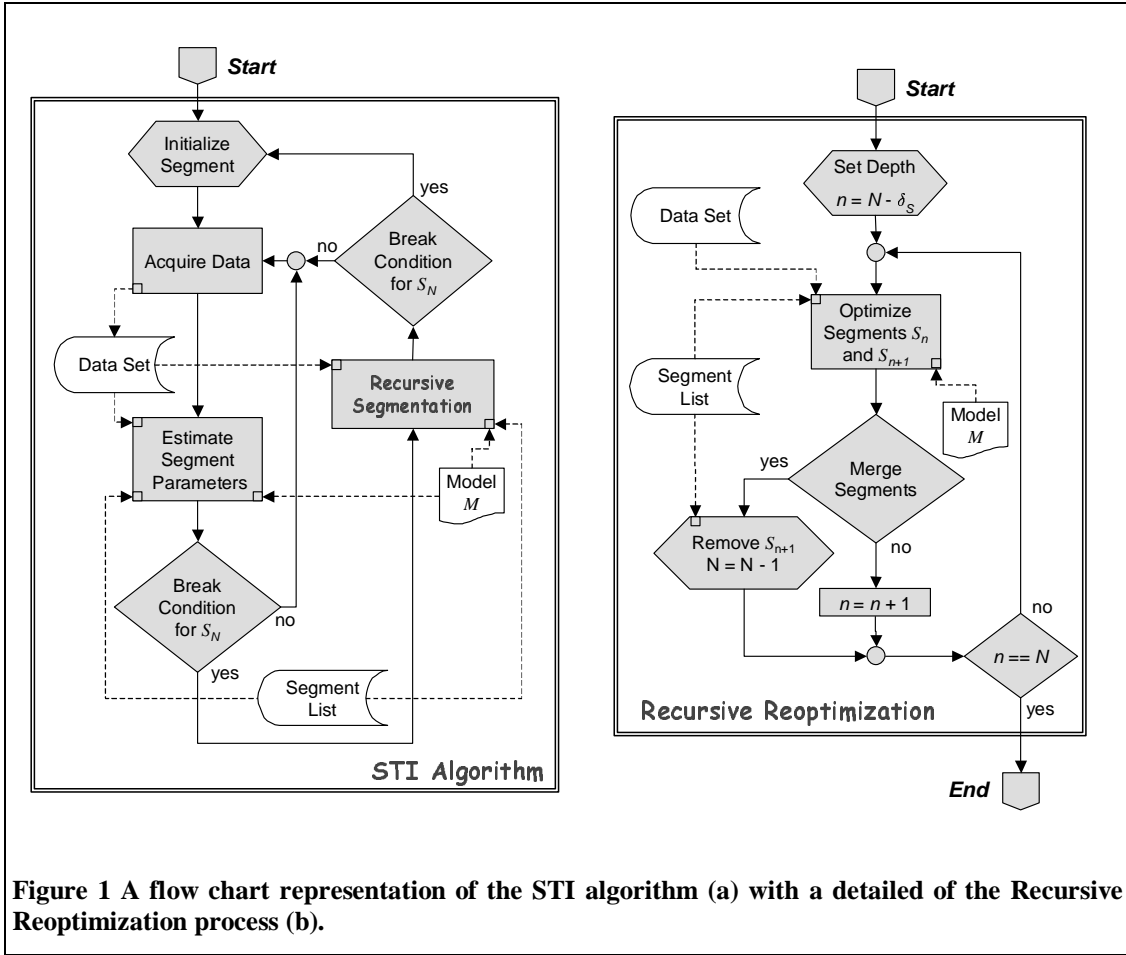


Figure 1 A flow chart representation of the STI algorithm (a) with a detailed of the Recursive Reoptimization process (b).

## Recursive Optimization

The *Recursive Optimization Stage* begins when a break condition has been met and there are multiple segments. Before measurements are added to a new segment, the last  $\delta_S$  segments are reoptimized. Optimization is performed recursively, beginning with the pair of segments  $S_{\delta_S}$  and  $S_{\delta_S+1}$ . STI searches for the knot location between the two segments that minimizes the total cost function for both segments. The total cost function is the sum of the costs of the fit of  $M$  for each segment and the continuity cost between  $S_{\delta_S}$  and  $S_{\delta_S+1}$  and if  $S_{\delta_S-1}$  exists the continuity cost between  $S_{\delta_S}$  and  $S_{\delta_S-1}$ . In general this optimization problem is a mixed integer, nonlinear programming problem. Since in our implementation the knot location is associated with the measurements of the two segments, the minimum cost can be found by a simple search and fit of all possible knot locations.

Next, STI tests if the two segments should be merged. The measurement from both segments are combined and a new segmented fitted. If the cost of the combined segment is less than  $K_m$  time the combined cost of the original two segments, the segments are merged. The algorithm then continues by reoptimizing segments  $S_{\delta_S+1}$  and  $S_{\delta_S+2}$ ,  $S_{\delta_S+2}$  and  $S_{\delta_S+3}$ , etc. until the last segment is reached. If after optimization a break condition still exists in the final segment, a new segment is begun. Otherwise the new measurement that created the break condition is added to the now reoptimized final segment. The algorithm then returns to the *Fit and Segmentation Stage*.

### Segment Model and Cost Functions

In the *Fit and Segmentation Stage* our implementation estimates the parameters for a segment  $S_n$  using nonlinear least-squares minimization of the form

$$\min_{\mathbf{X}_n \in \mathcal{M}} \frac{1}{2} \|F(\mathbf{X}_n, S_n, \mathbf{X}_{n-1})\|_2^2$$

where the cost function  $F$  is a stacked vector containing both the continuity cost for knot  $\mathbf{X}_{n-1}$  and measurement residuals for segment  $S_n$ . Our model  $M$  for this paper specifies that the elements of  $\mathbf{X}_n$  are the parameters of a Coordinated Turn (CT) model. The five CT parameters  $\{x_0, y_0, \phi_0, v, \omega\}$  are the initial  $x$  and  $y$  position and heading of the segment, the speed and turn rate of the target respectively. The estimated position  $\hat{\mathbf{x}}_k$  for the  $k^{\text{th}}$  data point associated with the segment is then given by

$$\begin{aligned} \hat{x}_k &= x_0 + \frac{2v}{\omega} \sin\left(\frac{k\omega}{2}\right) \cos\left(\phi_0 + \frac{k\omega}{2}\right) \\ \hat{y}_k &= y_0 + \frac{2v}{\omega} \sin\left(\frac{k\omega}{2}\right) \sin\left(\phi_0 + \frac{k\omega}{2}\right) \end{aligned}$$

When  $\omega = 0$  the equations reduce to a constant velocity mode of the form

$$\begin{aligned} \hat{x} &= x_0 + vk_s \cos(\phi_0) \\ \hat{y} &= y_0 + vk_s \sin(\phi_0) \end{aligned}$$

Our model  $M$  specifies that the cost of  $\mathbf{X}_{N-1}$  include continuity in position and direction

$$K_X L_N \left[ \|\mathbf{x}_0^N - \mathbf{x}_F^{N-1}\|^2 + \|\phi_0^N - \phi_F^{N-1}\|^2 \right]$$

where  $L_N$  is the number of measurements in  $S_n$ ,  $x_0^N$  is the initial target position in  $S_n$  and  $x_F^{N-1}$  is the final target position in  $S_{n-1}$ ;  $\phi_F^{N-1}$  is the final heading in  $S_{n-1}$  and  $\phi_0^N$  is the initial target heading in  $S_n$  expressed in radians.  $K_x$  is a tuning parameter that adjusts the importance of a smooth fit versus a good fit to the measurements themselves.

During the *Recursive Optimization Stage* our model is augmented to include the cost of the two segments along with the cost of the two knots that precede each segment. The minimization problem then becomes

$$\min_{X_n, X_{n+1} \in M} \frac{1}{2} \|F(X_n, X_{n+1}, S_n, X_{n-1}, S_{n+1}, X_n)\|_2^2$$

where the search space is extended to include all valid locations of the knot  $X_n$ . Because of the added continuity condition and the extra data from two segments the fit from this model is often better than from the model for a single segment.

## Discussion

As shown by the previous derivation, STI is inherently a smoothing algorithm, it continuously reoptimizes the previous track segmentation when additional data is acquired to improve the estimation accuracy for the current segment. These improvements can be seen even for small lags. When lags are larger a concise smoothed piecewise parametric representation of the track history is obtained.

The STI approach differs from the segmenting algorithms developed by the pattern recognition community in that STI is a real-time algorithm that continuously estimates the target track as new measurement data is acquired. In order to achieve a computationally tractable algorithm, STI makes the reasonable assumption that only the most recent segments have an influence on the current track estimate<sup>‡</sup>, and STI additionally constrains computational complexity by recursively optimizing knot placements between pairs of segments, rather than simultaneously searching for multiple optimal

---

<sup>‡</sup> This assumption also implies a new break condition may be the manifestation of a recent erroneous segmentation. Therefore, the STI algorithm always recursively optimizes the most recent segments before initiating a new segment.

knot placements between multiple segments. Even with these constraints, the run time performance of the STI algorithm is poor when compared to the variable structure IMM algorithms.

The STI implementation for this paper is written in Matlab® and uses the *lsqnonlin* function included in Matlab®’s *Optimization Toolbox* to perform the non-linear least squares fit to the combined motion model and knot continuity cost function. *lsqnonlin* implements a “subspace trust region method based on the interior-reflective Newton method”. STI ran 1000 times slower than the AGIMM tracker when benchmarked on a track with 9 segments and 400 data points. Average run times for 20 trials were: AGIMM, 0.63 sec; VS-AIMM, 0.29 sec; and STI 469 sec. As with most nonlinear optimization techniques, the runtime is nondeterministic; STI run times ranged from 366 to 523 seconds.

## AGIMM

The Adaptive Grid IMM (AGIMM) tracker initially formulated by (Jilkov, Angelova and Semerdjiev 1999) uses three constant speed Coordinated Turn (CT) model-based filters. The CT models are *left*, *center* and *right*-turn models with turn rates centered around the probabilistically weighted sum of the three models:  $\omega_{k+1}^C = \mu_k^L \omega_k^L + \mu_k^C \omega_k^C + \mu_k^R \omega_k^R$  where  $\mu_k^L$ ,  $\mu_k^C$  and  $\mu_k^R$  are the model probabilities at time  $k$  for the left, center and right models, respectively. Additionally, the AGIMM algorithm increases the separation (e.g. grid size) between the center turn rate  $\omega_{k+1}^C$  and the left and right turn rate  $\omega_k^L$  and  $\omega_k^R$  when the mode probabilities strongly indicate that the target is turning to the left or the right, respectively. Conversely, if the model probabilities for the left and right models are too low the separation is decreased.<sup>§</sup>

A later paper by (Semerdjiev, Mihaylova and Li 2000) modified the AGIMM tracker by substituting a Constant Velocity (CV) model for the center CT model using the turn rates for the two CT models centered around  $\omega_{k+1}^C = \mu_k^L \omega_k^L + \mu_k^R \omega_k^R$ . Our simulations show that this configuration increased estimation accuracy.

---

<sup>§</sup> Our simulation results show that these grid adjustments are rarely performed.

## VS-AIMM

The Variable Structure Augmented IMM (VS-AIMM) tracker also uses three parallel Kalman filters: a single CV filter and two CT filters. In the VS-AIMM implementation the target turn rate is indirectly estimated by the two CT model-based Extended Kalman Filters. Using a hybrid structure to help insure tracker stability, the estimated turn rate is maintained independently of the filters, where the filters only estimate the change in the turn rate at every iteration of the filter. Using a  $\Delta\hat{\omega}(t)$  from the EKF and a set of rules, the turn rate is updated at every iteration of the filter while keeping a bound on the turn rate and separation between the turn rates used by each model.

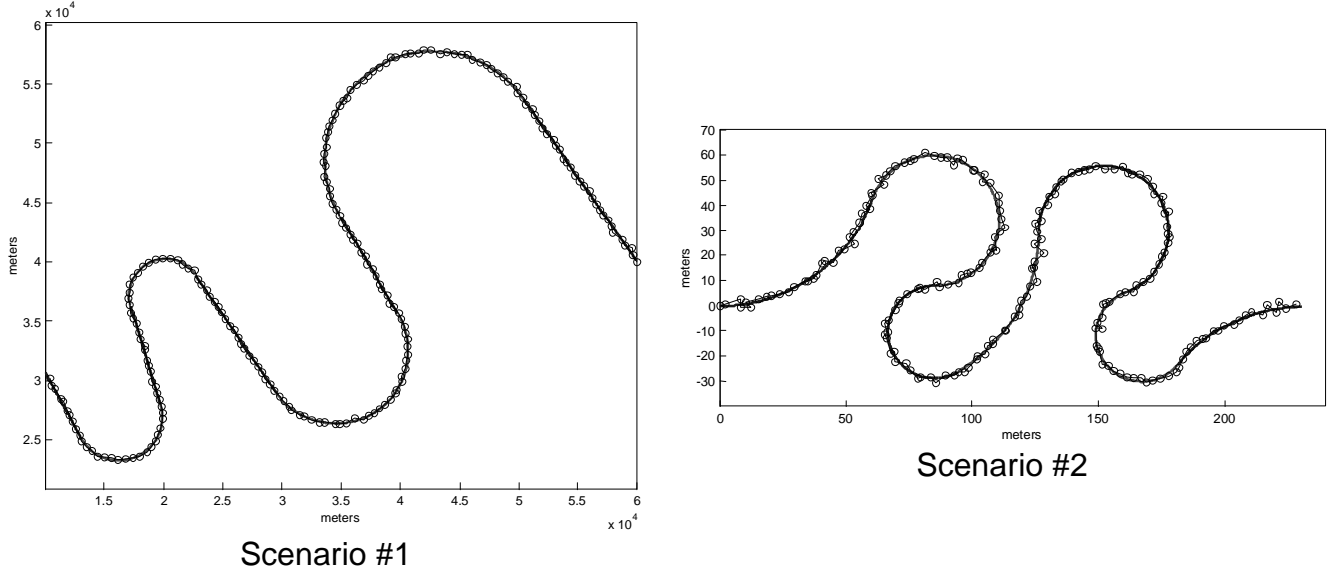
## IV. SIMULATION RESULTS

The performance of the STI was evaluated against the VS-AIMM, AGIMM and a fixed-interval Constant Velocity Kalman Smoother (Roweis and Ghahramani 1999) using two track geometries shown in Figure 2. **Track Scenario #1** is taken from (Jilkov, Angelova and Semerdjiev 1999; Semerdjiev, Mihaylova and Li 2000) while **Track Scenario #2** is a scenario for a highly maneuvering, nimble target making linked coordinated turns with large (up to  $60^\circ$ ) step changes in turn rates. We considered a single Cartesian sensor corrupted with zero-mean Gaussian noise with standard deviation  $\sigma = \sigma_x = \sigma_y$  and examined two sampling rates for each track scenario. Tuning parameters for all trackers were kept constant for both scenarios. Full knowledge of  $\sigma = \sigma_x = \sigma_y$  was assumed.

These simulations were done using Matlab® and the Optimization Toolbox. Code for the VS-AIMM tracker was obtained from Lyudmila Mihaylova and code for the fixed interval Kalman Smoother was obtained from A-V.I. Rosti at <http://svr-www.eng.cam.ac.uk/~avir2/>. The AGIMM tracker was implemented by the authors. The code for the complete simulation can be obtained by email from [spl@alum.mit.edu](mailto:spl@alum.mit.edu).

### Tracker Parameters

The VS-AIMM and AGIMM tuning parameters were taken directly from (Semerdjiev, Mihaylova and Li 2000). Only the maximum turn rate was increased to  $\omega_{\max} = 80^\circ$  to accommodate the turn



**Figure 2** Track Scenario #1 is for a target that performs four 180° turns. Superimposed on the track are 400 measurements with  $\sigma = 85$ . Track Scenario #2 is for a nimble target. Superimposed on the track are 200 measurements with  $\sigma = 1.0$

rates in Scenario 2<sup>\*\*</sup>. The Kalman filter and the fixed-interval Kalman Smoother uses a piecewise constant white acceleration model. The smoothing interval includes all measurements. The process noise standard deviation for both the filter and smoother is taken as the RMS acceleration of Scenario 1,  $\sigma_v = 9.47$ . The following table summarizes the tuning parameters for the STI algorithm.

Parameter	Simulation Value	Description
$L_{min}$	5	minimum segment length
$L_h$	2	the length of the segment head
$K_{\Delta}$	2	maximum residual ratio between tail and head
$K_{RMS}$	1.2	maximum ratio of RMS(v) to $\sigma$
$K_x$	0.25	the continuity cost gain
$K_{\delta}$	2	the recursive depth of the knot optimization
$K_m$	1.1	merge ratio
$\phi_{max}$	$\pi/16$	maximum discontinuity in heading at knot

Please refer to the source code for complete configuration of the trackers and the parameters.

---

<sup>\*\*</sup> This adjustment did not effect the performance for Scenario 1.

## Scenario 1

Track Scenario #1, shown in Figure 2, is for a target that performs four successive 180° turns with turn rates of 1.87, -2.8, 5.6, and -4.68°/sec during the time intervals [56,150], [182, 245], [285, 314], and [343,379] seconds respectively. The target has initial position of [60,000 40,000], an initial velocity of [246, -172], and maintains a speed of 300 meters/sec for 400 seconds. Measurement noise standard deviation is  $\sigma_x = \sigma_y = 85$ . Two measurement scenarios were evaluated, with  $N = 200$  and  $N = 400$ .

## Scenario 2

Track Scenario #2, shown in Figure 2, is for a highly maneuverable target that performs a series of linked turns of 10, -25, 35, 10, -25, and 35°/sec for duration of 7, 10, 6, 6, 10, 6 and 5 seconds respectively. While this high g scenario is not obtainable by a manned vehicle, it can be realized by the small unmanned drones that are currently being developed, and by birds, fish, etc.

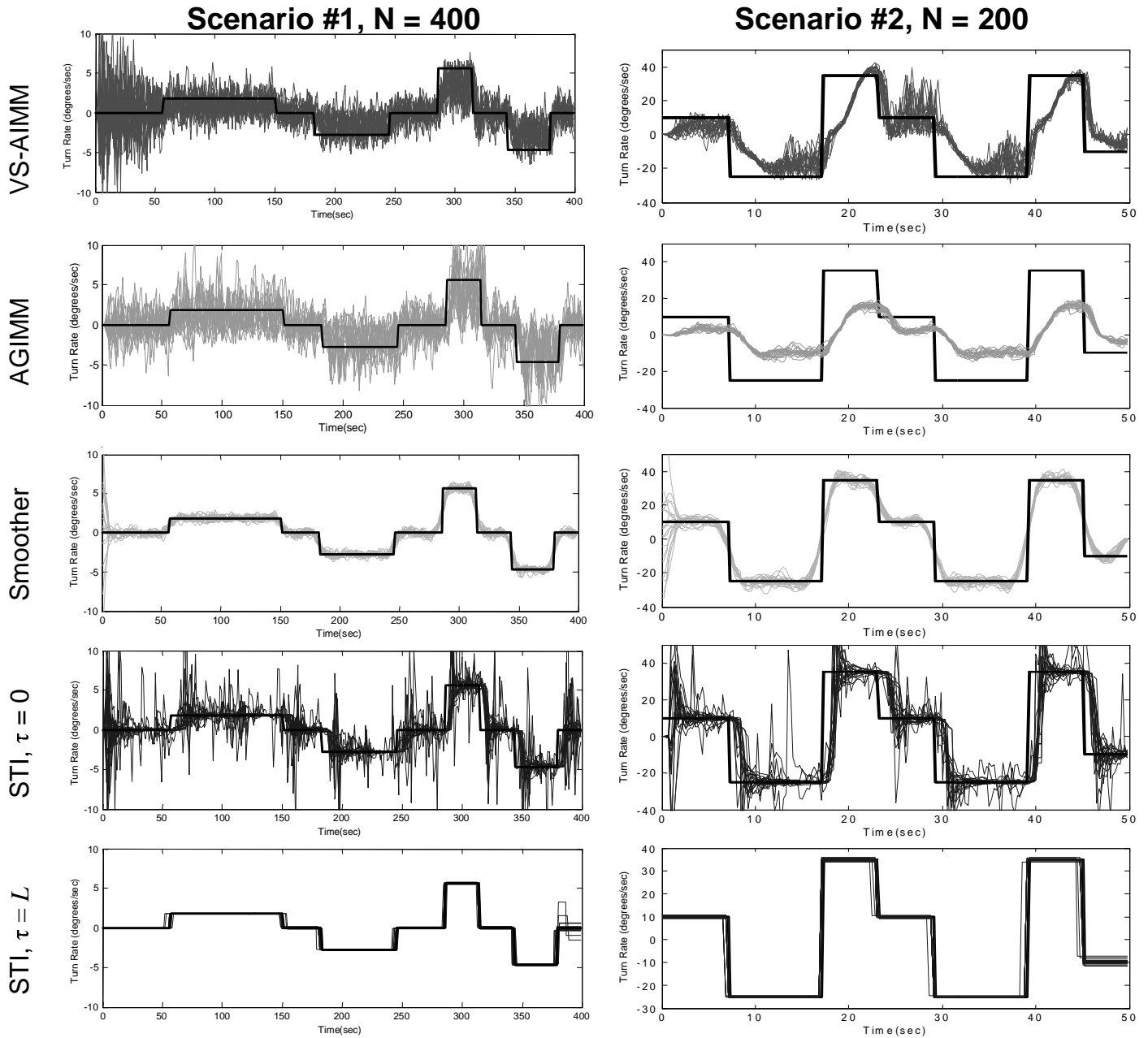
The target has an initial heading of 0°, initial position of [0 0], and speed of 11.252 meters/sec. Two measurement scenarios were evaluated, with  $N = 100$  and  $N = 200$  for  $\sigma_x = \sigma_y = 1$ .

## Statistics

The RMS estimation errors given in Table 1 show the VS-AIMM tracker gives better position esti-

**Table 1. Average RMS error in estimating position and turn rate (with standard deviation) for 100 trials.**

		Scenario #1				Scenario #2			
		N = 200		N = 400		N = 100		N = 200	
		Position	Turn Rate	Position	Turn Rate	Position	Turn Rate	Position	Turn Rate
CV Kalman filter		114 (4.93)	1.88 (0.09)	117 (5.8)	2.05 (0.20)	1.73 (0.09)	15 (0.66)	1.98 (0.09)	16.6 (1.07)
	VS-AIMM	85.5 (4.34)	2.06 (0.13)	73.4 (3.07)	2.05 (0.09)	3.56 (0.17)	16.9 (0.65)	2.18 (0.08)	16.3 (0.39)
AG IMM		89.6 (4.3)	2.0 (0.12)	76.6 (2.98)	2.14 (0.19)	1.31 (0.08)	18.7 (0.30)	1.21 (0.06)	18.7 (0.16)
STI	( $\tau = 0$ )	97.1 (9.27)	1.69 (0.09)	80.2 (5.15)	1.79 (0.14)	1.25 (0.13)	14.1 (1.44)	0.98 (0.10)	17.0 (0.93)
STI	( $\tau = 1$ )	64.4 (5.03)	1.45 (0.07)	60.3 (3.74)	1.69 (0.15)	0.8 (0.07)	7.07 (1.02)	0.69 (0.05)	15.0 (1.05)
Kalman Smoother		55.2 (3.92)	0.714 (0.06)	42.8 (3.52)	0.77 (0.28)	0.88 (0.069)	7.42 (0.38)	0.78 (0.05)	8.24 (1.74)
STI	( $\tau=L$ )	39.7 (6.87)	0.647 (0.09)	31.8 (5.78)	0.564 (0.10)	0.484 (0.09)	11.2 (0.48)	0.36 (0.07)	7.77 (0.68)



mates than both the AGIMM and the STI tracker for Scenario #1, while the STI tracker gives better position estimates for Scenario #2. The STI gives better turn rates estimates in the RMS sense for three of the four sets of trials, with the VS-AIMM outperforming the STI tracker for Scenario #2,  $N = 200$ .

When turn rate estimates are compared graphically in Figure 3 by superimposing the turn rate estimates of 20 trials, the difference in error characteristics between the algorithms is seen. The IMM trackers are too slow to respond to the change in turn rates, almost always both lagging and underestimating the true turn rates. STI also lags the true turn rate, although not as badly, and tends to have large individual error spikes but otherwise the turn rate estimates are centered around the true turn rate. These spikes are caused by improper segmentation, where a break condition is created by a noisy measurement and the new segment estimates a false turn rate from the noisy measurements. These effects are largely absent if smoothing is allowed as these segments are typically quickly recombined with the previous segment. The elimination of the turn rate errors by smoothing is visible in the smoothed STI output shown in Figure 3.

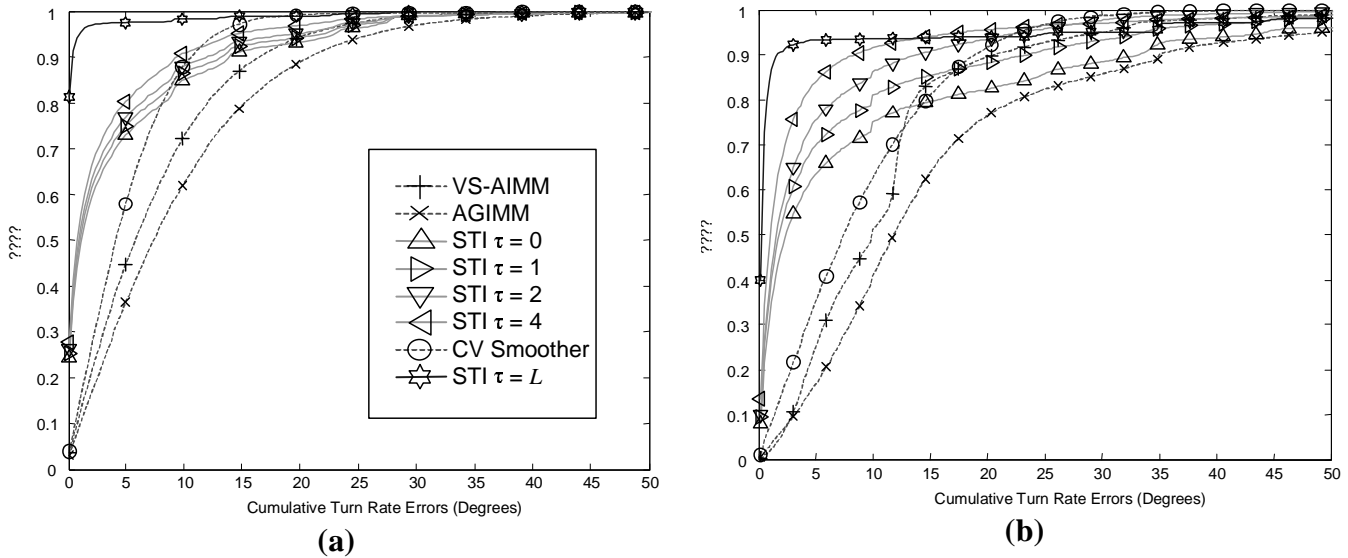
The large estimation errors caused by the lag in turn rate estimates, and the occasional spike inflates the RMS turn rate of the STI algorithm, while the RMS statistic tends to overstate the performance of the VS IMM trackers. A graph of the Cumulative Distribution Function (CDF) of the absolute turn rate error given in Figure 4 shows that the STI tracker gives consistently smaller errors in estimating turn rate than the other algorithms. This can also be quantified using a robust error statistic that does not give large weights to outliers, such as median absolute deviation (MAD). The turn rate MAD for each algorithm is given in Table 2. The MAD statistics for STI are at most one third of the best performing IMM tracker, and even outperform the Kalman smoother. Using this statistic, the consistent bias of the IMM algorithms becomes apparent.

## IV. CONCLUSION

**Table 2. Median absolute deviation in turn rate estimates for 100 trials.**

	Scenario 1		Scenario 2	
	N = 200	N = 400	N = 100	N = 200
<b>VS-AIMM</b>	1.041	0.203	10.000	8.376
<b>AG IMM</b>	0.952	1.509	11.884	23.577
<b>STI (<math>\tau = 0</math>)</b>	0.229	0.268	2.412	1.916
<b>STI (<math>\tau = 1</math>)</b>	0.193	0.243	1.817	1.649
<b>STI (<math>\tau = 2</math>)</b>	0.166	0.223	1.575	1.445
<b>STI (<math>\tau = 4</math>)</b>	0.152	0.198	1.055	1.235
<b>Smoother</b>	1.098	0.868	7.497	6.166
<b>STI (<math>\tau = L</math>)</b>	0.022	0.018	0.259	0.200

Our simulations show that STI can develop an accurate piecewise parametric representation of the track by dynamically partitioning it into segments with each track segment corresponding to a different modal state of the target. STI achieves results on par with the two VS IMM trackers, and is especially effective for situations with large changes in



**Figure 4. Cumulative distribution of absolute turn rate error for (a) Scenario #1,  $N=400$  and (b) Scenario #2,  $N=100$ .**

modal state, where the slow response of VS IMM trackers limits their effectiveness.

The run time performance of STI limits its current applicability to real-time tracking scenarios with fast update rates or a large number of targets. However, STI is currently applicable to tracking scenarios that do not require real-time performance, such as biological applications where tracking is often performed on a previously recorded data set. In these applications the low mass and speed of the targets allows them to maneuver with much higher turn rates than is possible for a rapidly moving aircraft. STI's piecewise parameterization of the target track may also be helpful in classifying these target's behavior.

## V. ACKNOWLEDGEMENTS

We would like to acknowledge the help of former student Matthew D. Ryan in developing the initial code for this project. This material is based on work previously supported by Dr. Teresa McMullen through the Office of Naval Research under Contract No. N00039-D-0042, Delivery Order No. D.O. 278.

## VI. BIBLIOGRAPHY

Chen, Bing and Jitendra K. Tugnait (2000). "Interacting Multiple Model Fixed-Lag Smoothing Algorithm For Markovian Switching Systems." *IEEE Transaction on Aerospace and Electronic Systems* **36**(2): 243-250.

Chen, Ke-Zhang, Xi-Wen Zhang, Zong-Ying Ou and Xin-An Feng (2003). "Recognition of digital curves scanned from paper drawings using genetic algorithms." *Pattern Recognition* **36**: 123-130.

Jilkov, V. P., D.S. Angelova and TZ. A. Semerdjiev (1999). "Design and Comparison of Mode-Set Adaptive IMM Algorithms for Maneuvering Target Tracking." *IEEE Transaction on Aerospace and Electronic Systems* **35**(1): 343-350.

Lim, Kah Bin, Kai Xin and Geok Soon Hong (1995). "Detection and estimation of circular arc segments." *Pattern Recognition Letters* **16**(6): 627-636.

Pittman, Jennifer and C. A. Murthy (2000). "Fitting Optimal Piecewise Linear Functions Using Genetic Algorithms." *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(7): 701-718.

Roweis, Sam and Zoubin Ghahramani (1999). "A Unifying Review of Linear Gaussian Models." *Neural Computation* **11**(2): 305-345.

Semerdjiev, Emil, Ludmila Mihaylova and X. Rong Li (2000). *Variable- and Fixed-Structure Augmented IMM Algorithms Using Coordinated Turn Model*. International Conference on Information Fusion (Fusion' 2000), Paris, France.

Sheu, Hsin-Teng and Wu-Chih Hu (1999). "Multiprimitive segmentation of planar curves - a two-level breakpoint classification and tuning approach." *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(8): 791-797.

Wenyin, L.; Dori, D. (1998). "Incremental arc segmentation algorithm and its evaluation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(4): 424 - 431.

West, G.A.W. and P.L. Rosin (1991). "Techniques for segmenting image curves into meaningful descriptions." *Pattern Recognition* **24**: 643-652.

Yoshimoto, F.; Moriyama, M.; Harada, T. (1999). *Automatic knot placement by a genetic algorithm for data fitting with a spline*. International Conference on Shape Modeling International '99.