COMPARING NUMERICAL METHODS FOR NONLINEAR TRACKING APPLICATIONS[§]

Stanton Musick[†], John Greenewald[l], Keith Kastella[‡], Chris Kreucher[‡]

[†]Sensors Directorate, Air Force Research Laboratory, Wright-Patterson AFB, Ohio, USA
Stanton.Musick@wpafb.af.mil

[1]Veridian Engineering, 5200 Springfield Pike, Dayton, Ohio, USA
[‡]Veridian Systems, PO Box 134008, Ann Arbor, Michigan, USA
{John.Greenewald,Keith.Kastella,Chris.Kreucher}@Veridan.com

ABSTRACT

Performance can be enhanced in dim target tracking by using predetection methods that directly accrue pixelized, unthresholded data in a nonlinear filter (NLF). A significant impediment to using NLF methods for real-time applications is finding numerical techniques that are fast and accurate enough to solve the associated NLF equations. Recent progress in NLF research has produced promising new numerical solution methods. This effort implemented a particle filter (PF) and a finite difference filter (based on a partial differential equation solver known as alternating direction implicit, ADI), and compared them under variations in signal-to-noise ratio (SNR). When each method was given equal computational resources, we found that RMS errors were consistently lower for PF than ADI, the maximum difference being about 60%. More significantly, PF appears to be much more robust against track loss at low SNR levels.

1. INTRODUCTION

Development of effective numerical methods for nonlinear tracking applications is a significant hurdle for data fusion systems. One of the most stressing applications of nonlinear filtering (NLF) is pre-detection tracking: tracking algorithms that use pixelized, unthresholded data as the filter input. Pre-detection tracking avoids the information-losing step of thresholding the data to produce partial target state measurements such as range, range-rate and bearing. If these nonlinear problems can be solved with sufficient numerical accuracy and speed, it will allow targets to be tracked at lower signal-to-clutter+noise ratios (SCNR), enabling improved performance for detecting and tracking targets under foliage, surface vessels in the littoral zone and cruise missiles.

While it is well known that the Kalman filter is the Bayes' optimal estimator for the standard linear Gaussian problem, much less is known theoretically about predetection tracking – even when the target motion itself is linear.

However, there has been much recent progress in the area of nonlinear filtering that is applicable to problems of this sort. There are now a number of viable numerical methods to solve nonlinear filtering problems. Candidates include (but are not limited to):

1) particle methods [1, 3, 8],
2) finite difference methods [4,5],
3) spectral methods [6],
4) probabilistic data association methods, and
5) multiple hypothesis, multiple frame methods.

While not fully mature, these methods have been developed to the point where they can now be considered for transition to deployable systems. Their relative performance for realistic problems needs to be better understood. The actual gain in estimation performance obtained by processing predetected data has not been characterized. How this performance gain varies with SCNR and the impact of target maneuvers remains to be investigated. The nature of the numerical error incurred by each method and its impact on various parts of the target envelope remains to be systematically studied.

To promote the development of engineering expertise in this area, the Air Force Research Lab (AFRL) is sponsoring a study project to evaluate the effectiveness and relative numerical performance of nonlinear filtering algorithms. To support this effort, which is being called a challenge problem, AFRL is developing a set of synthetic, ground-truthed data sets and associated metrics for testing in the research community. We propose two approaches to generating the target motion. As a baseline, we propose to generate random target motions using a simple 2nd order model $(x, \dot{x}, y, \dot{y})$. This has some advantages from an analysis point of view in that it is simple and enables focused study on numerical NLF issues. In a subsequent approach, we propose scenarios involving multiple targets maneuvering at unspecified times with variable normal accelerations.

The primary measures of performance (MOPs) are target RMS error as a function of SNR. As part of this effort, a separate MOPs module has been developed. This code is available in the form of commented MATLAB code at https://www.tenet.vdl.afrl.af.mil.

The remainder of this paper is organized as follows. Section 2 briefly summarizes the basics of nonlinear filtering in a Bayesian formulation and defines the simple motion model we have used for estimation. Section 2.1 presents the sensor model, and Section 2.2 presents the definition of effective signal-to-noise ratio that we use. Section 2.3 briefly presents the Sampling, Importance, Resampling (SIR) particle method filter [1,3,5] developed for these tests and Section 2.4 presents the Alternating Direction Implicit (ADI) finite difference scheme [4,5,10,11] used here. Section 2.5 presents the definitions of the performance metrics used in this evaluation. Section 3 presents results and Section 4 presents conclusions and suggests a few directions for further work.

2. BAYESIAN NONLINEAR FILTERING

To track a target from a sequence of measurements $\mathbf{y}_k$ made at discrete times t_k, let the target dynamics be described by an Ito stochastic differential equation [2] for the time-dependent target state $\mathbf{x}_t$

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)dt + \mathbf{G}(\mathbf{x}_t, t)d\boldsymbol{b}_t, \quad t \geq t \qquad (1)$$

where $\mathbf{x}_t$ and $\mathbf{f}$ are n-vectors, $\mathbf{G}$ is an $n \times r$ matrix function, and $\{\boldsymbol{b}_t, t \geq t_0\}$ is an r-vector Brownian motion process with $E\{d\boldsymbol{b}_t d\boldsymbol{b}_t^{\mathrm{T}}\} = \mathbf{Q}(t)dt$. The set of observations through time t_k is denoted

$$\mathbf{Y}_{t_k} = \{y_k, y_{k-1}, \cdots, y_1\} = \{y_k\} \cup \mathbf{Y}_{t_{k-1}} \qquad (2)$$

Between observations the evolution of the conditional density is determined by the target dynamics as characterized by the Ito equation. The time evolution of the joint density between measurements is the solution to the Fokker-Planck equation (FPE)

$$\frac{\partial p}{\partial t}(\mathbf{x}_t \mid \mathbf{Y}_{t_k}) = L(p), \quad t_k \leq t < t_{k+1}, \qquad (3)$$

where

$$L(p) \equiv -\sum_{i=1}^{n} \frac{\partial(\mathbf{f}_i p)}{\partial \mathbf{x}_i} + \frac{1}{2}\sum_{i,j=1}^{n} \frac{\partial^2 \left(p\left(\mathbf{G}\mathbf{Q}\mathbf{G}^{\mathrm{T}}\right)_{ij}\right)}{\partial \mathbf{x}_i \partial \mathbf{x}_j} \qquad (4)$$

with initial condition given by $p(\mathbf{x}_{t_k} \mid \mathbf{Y}_{t_k})$.

Then given a new observation y_k, the updated conditional density $p(\mathbf{x}_{t_k} \mid \mathbf{Y}_{t_k})$ is obtained from the predicted density $p(\mathbf{x}_{t_k} \mid \mathbf{Y}_{t_{k-1}})$ using Bayes' formula.

In this version of the challenge problem, we focus on the effect of measurement nonlinearity due to low SNR, and use a *linear* motion model, the so-called "constant velocity" model with $\mathbf{x} = (x, \dot{x}, y, \dot{y})^{\mathrm{T}}$, $\mathbf{f}(\mathbf{x}) = (\dot{x}, 0, \dot{y}, 0)^{\mathrm{T}}$,

$$\mathbf{G} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \text{ and } \mathbf{Q} = \begin{pmatrix} q & 0 \\ 0 & q \end{pmatrix}. \qquad (5)$$

The resulting FPE is

$$\frac{\partial p}{\partial t} = -\dot{x}\frac{\partial p}{\partial x} - \dot{y}\frac{\partial p}{\partial y} + \frac{q}{2}\frac{\partial^2 p}{\partial \dot{x}^2} + \frac{q}{2}\frac{\partial^2 p}{\partial \dot{y}^2} \qquad (6)$$

2.1 Sensor Model

An example of how to incorporate the likelihood function into NLF is presented by pixelized data such as point-target image-tracking applications. Envelope-detected radar data takes a similar form. In this case an image contains M pixels labeled $i = 1, \ldots, M$. The measurement consists of the pixel output vector $\mathbf{y}_k = [y_{k,1}, \ldots, y_{k,M}]^{\mathrm{T}}$ where the $y_{k,i}$ can be positive, real or complex, depending on the details of the sensor and signal processing. For point targets, pixel outputs are conditionally independent and

$$p(\mathbf{y}_k \mid \mathbf{x}_k) = \prod_{i=1}^{M} p(y_{k,i} \mid \mathbf{x}_k) \qquad (7)$$

When there is no target in a pixel, its output statistics are determined by the background rate $p_0(y_{k,i})$. Depending on the nature of the imager, this may be modeled by Rayleigh, Poisson, or more complicated distributions. Further, it may vary with pixel index and time, depending on the clutter statistics. When the target projects into a pixel i, its output statistics will be given by $p_1(y_{k,i})$, which, again, may be pixel and time dependent and will depend on the detailed nature of the sensor and target.

Using Bayes' formula directly to update the density is cumbersome because it involves M factors in the product for each discretized value of the target state vector $\mathbf{x}_{t_k}$. This can be simplified so that only the measurement likelihood ratio

$$l(y_{k,i_{\mathbf{x}_k}}) = p_1(y_{k,i_{\mathbf{x}_k}})/p_0(y_{k,i_{\mathbf{x}_k}}) \qquad (8)$$

needs to be evaluated for each discretized value of the target state vector $\mathbf{x}_k$, a significant saving in computation.

To illustrate, for a Rayleigh target with SNR parameter I, the probability distribution for intensity $y_{i_{\mathbf{x}_k}}$ of the target-containing pixel is

$$p_1(y_{i_{\mathbf{x}_k}}) = \frac{y_{i_{\mathbf{x}_k}}}{1+I}\exp\left(-y_{i_{\mathbf{x}_k}}{}^2/2(1+I)\right) \qquad (9)$$

The distribution for the background pixels is given by the same expression with $I = 0$. A little algebra shows that up to an irrelevant constant that can be dropped, the likelihood is

$$l(y_{i_{\mathbf{x}_k}}) = \exp\left(\frac{I y_{i_{\mathbf{x}_k}}{}^2}{2(1+I)}\right) \qquad (10)$$

2.2 Effective SNR

While signal-to-noise ratio (SNR) is often used to gauge likely performance for filter processes, it is only rigorously related to performance in the case of Gaussian signals. The Kullback-Leibler discrimination, a more general quantity that is related to detection and estimation performance, is defined by

$$L(q_0;q_1) \equiv \int q_0(y)\ln\left(q_0(y)/q_1(y)\right)dy \qquad (11)$$

Its symmetrized relative, the divergence, is defined by

$$\bar{L}(q_0;q_1) \equiv L(q_0;q_1) + L(q_1;q_0) \qquad (12)$$

The divergence is a convenient measure of effective SNR. For a Gaussian signal given by

$$p_1(y) = \frac{1}{\sqrt{2pI}}\exp\left(-(y-I)^2/2I\right) \qquad (13)$$

$$p_0(y) = \frac{1}{\sqrt{2pI}}\exp\left(-y^2/2I\right) \qquad (14)$$

it is straightforward to find that

$$\bar{L}^{Gauss}(p_0;p_1) = I \qquad (15)$$

Similarly, for Rayleigh distributed measurements,

$$\bar{L}^{Ray}(p_0;p_1) = \frac{I^2}{1+I} \qquad (16)$$

Note that in the large I limit, the Rayleigh and Gaussian divergences are the same, while they differ significantly for small values of I.

2.3 Particle Method

Particle methods are a collection of Monte Carlo techniques in which the probability density is represented by a collection of N independent and identically distributed random samples, $\left\{\mathbf{x}_k^{(i)}; i = 1,\ldots,N\right\}$. The samples $\mathbf{x}_k^{(i)}$, referred to as particles, are distributed according to $\mathbf{x}_k^{(i)} \sim p(\mathbf{x}_k \mid Y_k)$. Then the conditional likelihood is approximated by

$$p(\mathbf{x}_k \mid Y_k) \approx \frac{1}{N}\sum_{i=1}^{N} d\left(\mathbf{x}_k - \mathbf{x}_k^{(i)}\right) \qquad (17)$$

where $d(\mathbf{x})$ is the Dirac d-function in target state space.

For this effort we implemented the particle filter using the technique called Sampling, Importance, Resampling (SIR), where we mechanized prediction and measurement update using the approximating density above. For prediction at time k, we propagate each particle $\mathbf{x}_{k-1}^{(i)}$ by producing a single draw from $p\left(\mathbf{x}_{k+1} \mid \mathbf{x}_k^{(i)}\right)$. The predicted particles are denoted $\mathbf{x}_k^{(i)}$ and the predicted density is approximated by

$$p(\mathbf{x}_k \mid Y_{k-1}) \approx \frac{1}{N}\sum_{i=1}^{N} d\left(\mathbf{x}_k - \mathbf{x}_k^{(i)}\right) \qquad (18)$$

At measurement update, a new set of particles is generated from the predicted set by resampling with likelihood $p\left(y_k \mid \mathbf{x}_k^{(i)}\right)$. Thus particles with high likelihood are sampled many times while particles with low likelihood are unlikely to appear in the resampled set.

2.4 ADI Finite Difference Method

For the finite difference filter, we used the Alternating Direction Implicit (ADI) scheme to solve the Fokker-Planck Equation. Although space does not permit a review of the ADI algorithms, here are some factors that must be considered in its implementation: resolution in space and time of the grid on which the solution will be computed; accuracy of the approximate solution formulas that will be used; methods for inverting tridiagonal matrix equations; initial conditions within and on the boundaries of the solution grid; methods for moving the computation sub-grid to keep it centered over the target which moves in a larger space.

See [7] for implementation details on both the particle and finite difference filters of the challenge problem.

2.5 Tracking Metrics

The RMS values of position and velocity error were chosen as metrics to illustrate performance in this paper. Error is defined as the difference between estimate and truth, $\mathbf{e}_t = \hat{\mathbf{x}}_t - \mathbf{x}_t$, where the estimate $\hat{\mathbf{x}}_t$ is computed as the mean of the estimated density $p(\mathbf{x}_t \mid \mathbf{Y}_t)$. Although RMS accuracy alone does not provide a thorough characterization of performance in a general tracking problem, it would seem to be appropriate in this special case where there is just one target (the usual multi-track metrics degenerate or disappear) and track state initialization is nearly perfect (e.g. convergence times would be artificially small).

As noted previously, it is expensive and unnecessary to compute the joint density over the entire range of motion. Instead, a computational gate is established on which to produce a solution, this gate being a small fixed-size subset of the motion region. The gate must be translated using estimates of target motion as inputs to a translation control algorithm. If the filter estimate drifts so far from the truth that the target exits the gate, *lost lock* occurs. With target observations lost, the filter diverges quickly and seldom recovers. The gate translation control problem was challenging anytime filter estimates were inaccurate, e.g. at startup or at the lowest SNRs. Lost lock events were logged and their frequency computed.

3. RESULTS

Accuracy as measured by RMS error improves for the ADI method with finer grid resolution, and improves for the PF method with increasing numbers of particles. However, as grid resolution becomes finer, the computational burden for ADI grows polynomially, but as particle count increases, the computational burden for PF grows only linearly. Given such complexities at this fundamental level, it is easier and fairer to compare the two filtering methods on the basis of equal computational burden, which translates to "equal flops" in MATLAB. The results that follow examine performance at a single constant value of 630 Kflops per study. This flop level translates to 5121 particles for PF, and a 10x10x10x10 grid of 10,000 cells for ADI.

3.1 Experiment methodology

The estimation problem is to track a single dim target moving in a 2D space using intensity images of the track area for measurements. True target motion is generated using a nearly constant velocity (NCV) model of target dynamics, a model based on the assumption that acceleration is a white noise process, $a(t) = w(t)$. The white noise in the NCV model imposes randomness in the motion so that a differ-

ent true trajectory is produced on every Monte Carlo run. Target motion is represented in each filter by the same NCV model. This decision to match filter to truth avoids most mis-modeling issues. For the results that follow, even the noise strength of the filter was matched to the truth.

The simulated sensor images the entire target motion region to produce a scene of 256 x 256 pixels. The intensity in each pixel is governed by a Rayleigh distribution with noise power one. In the pixel holding the target, the intensity is adjusted for the SNR of the study. Identical simulated sensor images are input to each filter as measurements, but only the portion of the scene in the instantaneous gate contributes to the joint density estimate.

The initial state of each filter is chosen to approximately match the truth. The initial density of each filter is uniform in each of the four dimensions $(x, \dot{x}, y, \dot{y})$, and extends over the space in the initial gate. For the results that follow, the gate was fixed at 10x10 pixels in (x, y) space.

In the case of a lost lock event, accuracy degrades precipitously and the run is effectively ruined. When this occurs, data from that run is removed from the study ensemble, and a new run is made to replace the spoiled one.

Experimental results are based on studies of 50 Monte Carlo runs each. Studies were conducted for ADI and PF separately, at 2 dB intervals in the range 4-20 dB effective SNR, Eq. (16). Altogether, 18 studies (9 each for PF and ADI) contributed to the results reported next

3.2 Discussion

Figures 1 and 2 show RMS performance in position and velocity for the two methods. These figures were built from the 18 Monte Carlo studies of 50 runs each. Figure 1 shows reasonably good estimation of position for both PF and ADI. Position RMS error falls as SNR increases for both methods, as expected, with PF being roughly twice as accurate as ADI, on average.

Figure 2 shows improving velocity estimation for PF as SNR rises. For ADI, however, the RMS curve indicates poor estimation performance for all SNRs. Other tests were run to assess this situation, and they all confirmed that ADI was not estimating velocity. The resolution-induced limit on estimation accuracy for ADI is approximately 0.06 m/sec (2 m/sec spanned by 10 square grid cells) so the "flat" performance at roughly 0.45 m/sec is not resolution induced. We are not certain why ADI does not estimate velocity but we suspect a problem with its code. Unfortunately, this issue surfaced recently and we failed to find a correction before this paper was due.

Over all studies PF was more robust to noise and maneuvers than ADI, e.g. PF lost lock just 4 times per study for SNR under 12dB, versus roughly 20 times for ADI.

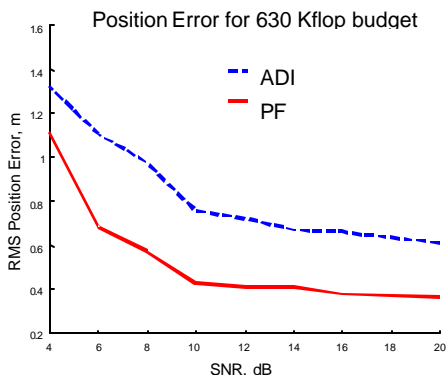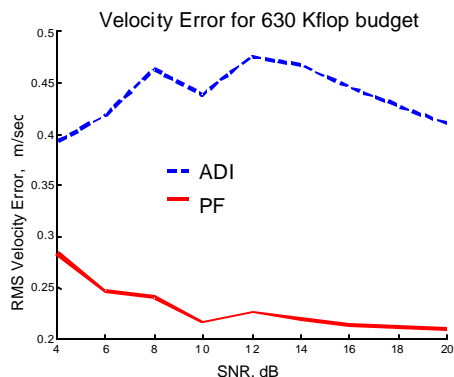Figure 1 – RMS position error, both methods

Figure 2 – RMS velocity error, both methods

4. SUMMARY

This paper introduces a challenge problem in nonlinear filtering that is made available to the research community in the form of MATLAB code posted at a public web site. This problem consists of modules for scenario generation and performance evaluation, as well as modules offering baseline solutions for two methods, a particle filter and an alternating direction implicit version of a finite difference scheme. These baseline solutions demonstrate in concrete terms how nonlinear methods can be applied to image intensity data. We are hopeful that their availability will encourage researchers with innovative ideas to apply their methods to the same problem data that was used here.

This paper also provides technical descriptions of the two solution techniques, and illustrates results from each.

This effort has shown that both particle and finite difference methods can solve the dim target tracking problem, and that particle methods produce greater accuracy for equal computational resources.

REFERENCES

[1] A. Doucet, "On sequential simulation-based methods for Bayesian filtering", Signal Processing Group, Department of Engineering, 1998, No. CUED/F-INFENG/ TR.310, University of Cambridge.

[2] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, Academic Press, New York, 1970.

[3] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "A novel approach to nonlinear/non-Gaussian Bayesian state estimation", IEEE Proceedings on Radar and Signal Processing, vol. 140, 1993, pp. 107-113.

[4] K. Kastella and A. Zatazelo, "Nonlinear filtering for low elevation targets in multipath," *Proceedings of SPIE, Signal and Data Processing of Small Targets 1998,* 14-16 April, 1998, Orlando, Florida, Vol. 3373, pp. 452-459.

[5] K. Kastella, "Finite Difference Methods for Tracking and Automatic Target Recognition", *Multitarget Multisensor Tracking: Advance Applications* Vol. 3, Y. Bar-Sholom, D. Blair (Eds.), Artech, 2000.

[6] S. Lototsky, R. Mikuleviclus, B. L Rozovskii, "Nonlinear filtering revisited: a spectral approach", *SIAM J. Contr. Optim.*, Vol. 35, No.2, March 1997.

[7] S. Musick, J. Greenewald, C. Kreucher, Keith Kastella, "Comparison of particle method and finite difference nonlinear filters for low SNR target tracking", Proceedings Fusion 2001, Montreal, Canada, Aug 2001.

[8] D. B. Rubin, Using the SIR algorithm to simulate posterior distributions, Bayesian Statistics 3, Oxford Universty Press, J. M. Bernardo, M. H. DeGroot, D. V. Lindley and A. F. M. Smith, eds., 1988, pp. 395-402.

[9] H.W. Sorenson, "On the development of practical nonlinear filters," *Information Sciences*, Vol. 7, 1974, pp. 253-270.

[10] J. C. Strikwerda, Finite Difference Schemes and Partial Differential Equations, Chapman & Hall, New York, 1989

[11] N. N. Yanenko, *The Method of Fractional Steps*, Springer-Verlag, 1971.