

# SOFT RESAMPLING FOR IMPROVED INFORMATION RETENTION IN PARTICLE FILTERING

Praveen B. Choppala, Paul D. Teal, Marcus R. Frean

School of Engineering and Computer Science  
Victoria University of Wellington  
Email: {praveen, pault, marcus}@ecs.vuw.ac.nz

## ABSTRACT

The reweighing process of particle filtering can result in very large variance of particle weights; a problem known as degeneracy. The usual solution to this is an intermediary resampling step in which particles with lower weights are replaced by copies of those with large weights. This resampling inevitably results in loss of the information contained in those particles of low weights. Most of the existing stochastic and deterministic resampling schemes cause further loss of information because all the resampled particles have equal weights. These current techniques are what we would call, “hard resamplers,” and they impede the accumulation of information over many successive observations, which affects the detection of very covert targets. This paper presents two variants of a soft and deterministic resampling procedure that retains information contained in the weights over multiple observations. We demonstrate their effectiveness using a track-before-detect application.

**Index Terms**— Particle filter, soft resampling, track-before-detect

## 1. INTRODUCTION

Many scientific problems require sequential estimation (or tracking) of the state of a dynamic system (or target). The objective of the estimator is to build a posterior probability density function (pdf) of the state of the target using the noisy sensor data (observations). Although many techniques have been developed over the years, Bayesian estimation [1, 2] provides a rigorous platform to accomplish this. One prominent Bayesian tracking algorithm in extensive use is the particle filter (PF) [3], which operates on the principle of approximating the posterior state density by a set of weighted samples, which in this context are known as particles [4].

Particle filtering [5] can be regarded as a technique that uses two fundamental approaches in succession. The first, *sequential importance sampling* (SIS) specifies the process of drawing new particles at each time step based on a state equation [5–7] and updating the weights according to the likelihood of the state they represent. By itself, SIS results in a large variance of the weights and this inefficiency is known as degeneracy. This is overcome using a second stage, *resampling*, that represents the posterior by a new set of particles [8, 9]. Consequently, the PF can be termed the *sequential importance resampling* (SIR) filter. The interest of this paper is in resampling, the objective of which is to replicate the particles with higher weights. The particles with lower weights are replaced by those with higher weights so that the final particle cloud represents the posterior more accurately. This replacement would reweight the replaced particles so that the variance among the weights is reduced

[8, 10].

Among the stochastic resamplers, multinomial [3] and residual [11] resampling were the first to be developed. Apart from being computationally expensive, these are based on duplicating a particle after comparing the cumulative weight with a random number  $\mathcal{U}(0, 1]$  (i.e, uniformly distributed between the maximum and minimum values that the cumulative normalized weights can take). If the random number is larger than the weight of a *potential particle*, even a particle with large weight could be discarded. A particle can be considered to be potential when it is starting to gain weight either because the target is making a turn towards it or it is in the process of declaring a covert target. Stratified [12] and systematic [5] resampling overcome this problem by pre-partitioning the space  $(0, 1]$  into  $I$  disjoint sets whose values increase linearly ( $I$  is the total number of particles). If the  $i$ th particle has a weight lower than the  $i$ th random number, it would be discarded. Hence these methods suffer from information loss due to random number comparison. All the above techniques reset all the weights equally. This could be interpreted to mean that each resampled particle is equally valid in representing the posterior density associated with the state of the target, which in practice is rarely true and unwanted in scenarios where a target makes a sharp maneuver or suddenly becomes covert.

This problem is overcome using deterministic resampling schemes which are threshold-based. One of its variants is optimal resampling [13] that proposes to keep the weights of the dominant (large weight) particles unchanged, while the others are resampled and reweighted using the stratified distribution. However, this technique does not reduce the variance among the weights. Partial resampling [9] proposes to keep the set of moderately weighted particles unchanged and resample the set of dominant and negligible particles. However, the reweighting is now based on the size of the sets but not on the individual weights. This causes loss of potential information contained in the negligible particles.

When  $I$  is large, the posterior density represented by all these techniques is an accurate representation of the true density. However it is usually the case that  $I$  is limited. Our proposed soft resampling procedures provide a method for enhancing the posterior representation with fewer particles.

The detection of very covert targets using a Bayes filter [14] relies on the accumulation of information over many consecutive observations. Most resampling procedures impede this accumulation. Hence the motive behind the current work is to devise a resampling scheme which retains this information. In scenarios where particles are searching for new (very covert) targets, the

current resamplers create fairly tight clusters so that there are not enough particles to span the entire observation space by the time a target appears. Hence the conventional track-before-detect (TBD) PF [4, 15] proposes to model the birth and death of particles so that when the target is absent, most particles are dead and take a maximum weight of 1. Hence the resampler does not produce many replacements and results in a loose cloud. The TBD PF efficiency therefore depends on the total number of particles. The greater the number of particles, the looser is the particle cloud and the faster the target lock, but also the higher is the computation. This paper proposes two variants of a novel soft resampling technique that is deterministic, reweights the particles based on their actual weights and the weights of the retained particles.

The rest of the paper is organized as follows. Section 2 sets the notation and gives a brief overview of the PF. We describe the two variants of the proposed soft resampling technique in section 3, give the evaluation details in section 4, relate to prior work in section 5 and conclude in section 6.

## 2. PARTICLE FILTERING

In this section, we fix the notation and mathematically describe how the particles are drawn by importance sampling and are sequentially updated for the next time step after resampling.

The posterior pdf of the state of the target  $x_k$  at time  $k$  can be given by  $p(x_k|z_{1:k})$ , where  $z_{1:k}$  denote the noisy sensor data up to time step  $k$ . This posterior is represented by a set of particles  $\{x_k^i\}_{i=1}^I$  and their corresponding weights  $\{w_k^i\}_{i=1}^I$ .  $i$  is the particle index and  $I$  is the total number of particles. At time  $k = 0$  the particles are drawn from a suitable distribution, for example a uniform or a Gaussian distribution. Their weights are also initialized. In the succeeding time steps  $k \geq 1$ , the new set of particles should be ideally drawn from the actual posterior  $p(x_k|z_{1:k})$ . Since this is difficult, particles are drawn from a convenient importance distribution  $q(x_k|z_{1:k})$ , which is generally the Markov transition density  $p(x_k|x_{k-1})$ . This can be represented as

$$x_k^i \sim q(x_k^i|z_{1:k}) \sim p(x_k^i|x_{k-1}^i) \quad (1)$$

The weights of the particles are updated as

$$w_k^i = \frac{p(x_k|z_{1:k})}{q(x_k|z_{1:k})} = w_{k-1}^i \frac{p(z_k|x_k^i) p(x_k^i|x_{k-1}^i)}{q(x_k^i|z_{1:k})} \quad (2)$$

where  $p(z_k|x_k)$  is the likelihood. Then the weights are normalized and the posterior can be approximated as

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^I w_k^i \delta(x_k - x_k^i) \quad (3)$$

As the algorithm progresses, the discrepancy between the weights increases. This problem is called degeneracy [5]. One measure of the efficiency in this context is the effective sample size [9], defined as

$$I_{\text{eff}} = \frac{1}{\sum_{i=1}^I (w_k^i)^2} \quad (4)$$

such that  $I_{\text{eff}} \ll I$ , and low  $I_{\text{eff}}$  exhibits high degeneracy. The solution to degeneracy is to resample the particles with replacement. Whenever  $I_{\text{eff}}$  falls below a certain threshold [8], the particles with negligible weights should be replaced by those with higher weights. The idea is to eliminate particles that have small weights and replace them at the locations described by those that have larger weights [9, 12].

## 3. SOFT RESAMPLING

In this section, we present our technique of soft resampling. The set of weights  $\{w_k^i\}_{i=1}^I$  at time  $k$  are obtained after importance sampling. These weights are first normalized as

$$w_k^i = \frac{w_k^i}{\sum_{i=1}^I w_k^i} \quad (5)$$

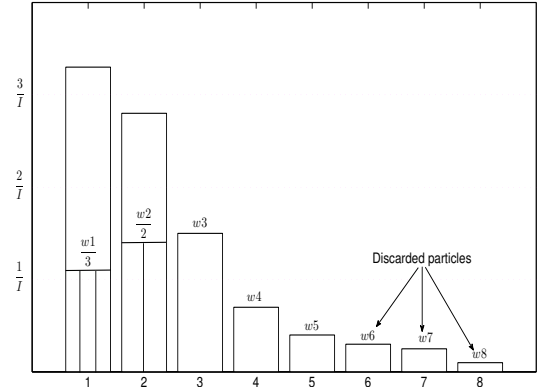
so that they sum to 1, and then sorted in descending order. Each particle  $x_k^i$  is replicated if its weight is larger than  $\frac{2}{I}$ , with the number of replications given by

$$C_{x_k^i} = \max\{1, \lfloor I \cdot w_k^i \rfloor\} \quad (6)$$

Each  $x_k^i$  that is now available  $C_{x_k^i}$  times will be reweighted as

$$a_k^i = \frac{w_k^i}{C_{x_k^i}} \quad (7)$$

It is now evident that, every particle whose weight is less than  $\frac{2}{I}$  will appear only once with its weight unchanged, while every particle whose weight is greater than  $\frac{2}{I}$  will appear as many times as the integer part in the numerator of its weight with its weight equally divided amongst them. This procedure can be visualized in Fig. 1



**Fig. 1:** Among the 8 particles, the first particle has a weight of  $\frac{3.3}{I}$  and hence gets 3 copies each weighted at  $\frac{w_1}{3} = \frac{1.1}{I}$ . The second particle has a weight of  $\frac{2.8}{I}$  and hence gets 2 copies each weighted at  $\frac{w_2}{2} = \frac{1.4}{I}$ . The other particles' weight is left unchanged since they are all less than  $\frac{2}{I}$ . The last 3 particles are discarded since we need only 8 particles

Since only  $I$  reweighted particles are enough, the algorithm is terminated once  $I$  particles are created, and the others are discarded. This implies that we have *deterministically replicated the particles and reweighted them based on their individual weights*. The weight of the discarded particles can be reallocated in one of the two following ways.

**Soft resampling - normalization:** The weights can then be normalized as

$$b_k^i = \frac{a_k^i}{\sum_{i=1}^I a_k^i} \quad (8)$$

Normalizing would adjust the weights so that they sum to 1. When these weights are reused in the subsequent time step, they accurately describe the previous particles' influence on the new particle set.

**Soft resampling - redistributing the discarded weights:** The weights of the discarded particles can be assigned to the retained particles in the following procedure. We first evaluate the total discarded weight as

$$a_{\text{spare}} = 1 - \sum_{i=1}^I a_k^i \quad (9)$$

This weight is shared among the lower weight particles. To do this, we find the particle index  $p$  such that

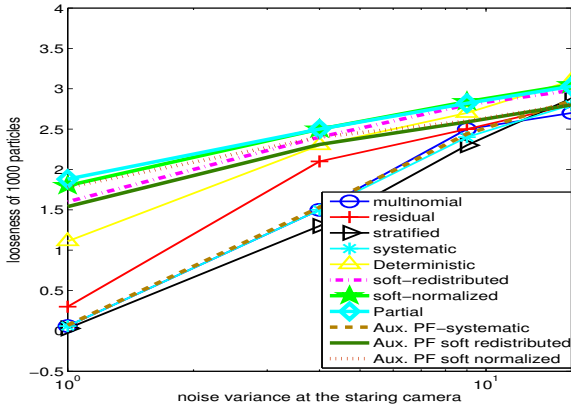
$$a_{\text{spare}} + a_k^I + a_k^{I-1} + \dots + a_k^{I-p} \leq a_k^{I-p-1} \quad (10)$$

and each of the  $I - p$  particles starting from the  $I$ th particle are reweighted as

$$\{b_k^i\}_{i=I-p}^I = \frac{a_{\text{spare}} + \sum_{i=I-p}^I a_k^i}{p} \quad (11)$$

#### 4. EVALUATION

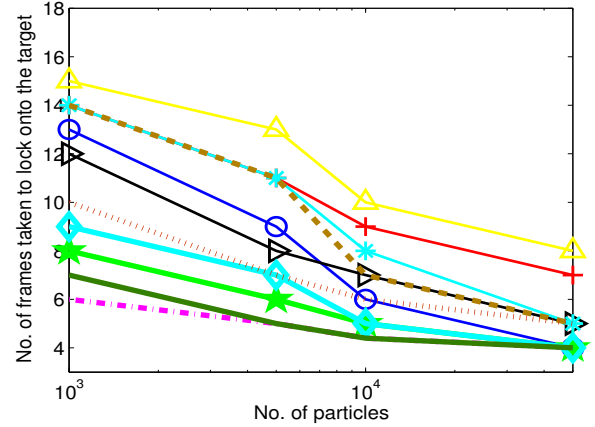
To demonstrate the validity of our techniques, we first test the softness of our methods. For this we used the PF based TBD numerical example described in chapter 11 of [4], which is also found in [14]. The observations now are  $10 \times 10$  frames which are 2-D snapshots from a staring camera and all the results are averaged over 50 iterations. Fig. 2 shows the looseness of 1000 particles measured according to  $C_{\text{tight}} = \frac{\sum \|x_k^i - \sum_{i=1}^I \frac{x_k^i}{I}\|^2}{I}$  versus the noise variance at the camera with no target appearing for 30 frames. It can be observed that our techniques leave a looser cluster by the 30th frame.



**Fig. 2:** Tightness of 1000 particles versus noise variance. Particles keep searching for 30 frames with no target. The implementations are the PF with multinomial [3], residual [11], stratified [12], systematic [5] (implemented from [8]), deterministic (with the threshold chosen as  $\frac{1}{T}$ ) [13] and partial [9] (with the thresholds chosen as  $\frac{2}{T}$  and  $\frac{9}{T}$ ) resampling, auxiliary PF [6] with systematic resampling and our proposed techniques on the PF and the auxiliary PF. The legend of this figure applies to Fig. 3, Fig. 4, Fig. 5 and Fig. 7 also.

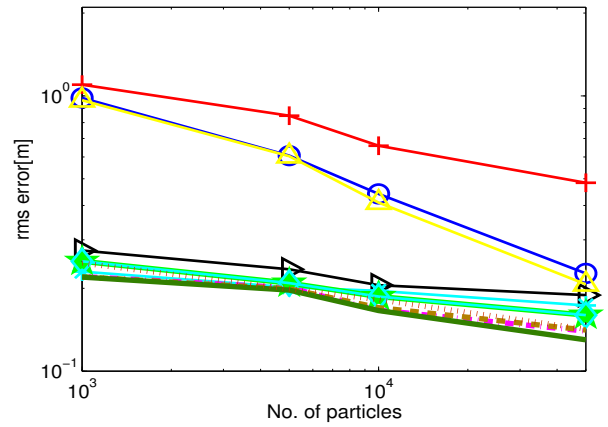
Since there is a loose cluster of particles to span the entire observation space, our techniques lock onto a new target much faster. To

demonstrate this, we use the same example, now with a noise standard deviation of  $\Sigma = 2$ . The target appears in the 8th frame and maneuvers itself for 20 frames. Fig. 3 shows the number of frames taken to lock onto the target versus the number of particles, and it can be observed that our techniques lock faster with fewer particles.



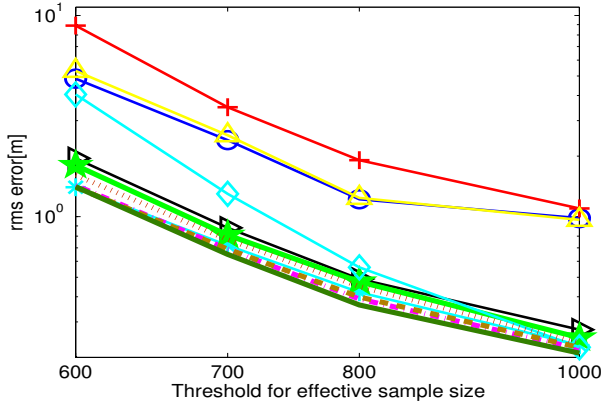
**Fig. 3:** No. of frames taken from the appearance of a target to lock onto a target (i.e for the rms error to decrease to below 0.1) versus no. of particles. The target appears in the 8th frame and maneuvers for 20 frames.

We now show that redistributing the weight of the discarded particles improves the accuracy of the scheme. Considering the same scheme where the target appears in the 8th frame and maneuvers for 20 frames, Fig. 4 shows the rms error versus the number of particles. It can be observed that the error is lower for the proposed methods.



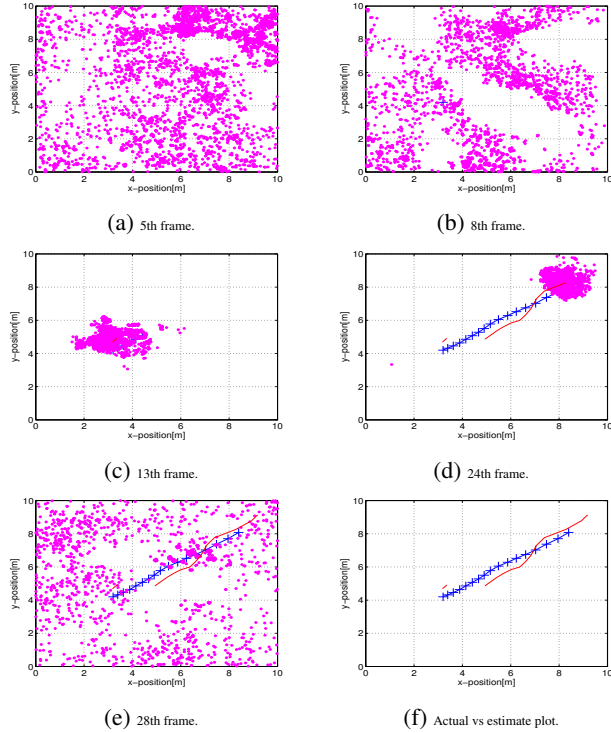
**Fig. 4:** rms error versus number of particles. Noise variance at the camera is 4. The target appears in the 8th frame and maneuvers for 20 frames.

It is a common practice to resample when the effective sample size  $I_{\text{eff}}$  drops below a threshold rather than at every time step [3, 5]. Fig. 5 shows rms error versus the  $I_{\text{eff}}$  threshold and it can be observed that our techniques perform accurately with a lower threshold, thereby allowing resampling less frequently, which in turn reduces the computational load.



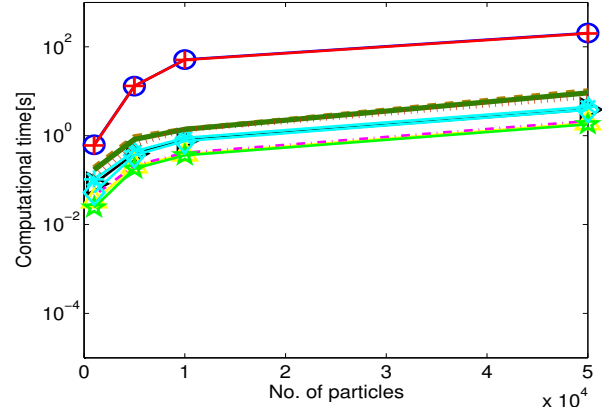
**Fig. 5:** rms error vs  $I_{\text{eff}}$  threshold for 1000 particles. Noise variance at the camera is 4. Target appears in the 8th frame and maneuvers for 20 frames.

We now demonstrate the complete TBD procedure and show that our resampler can be used directly in a classical PF, bypassing the need to model the birth and death of the targets. All the particles are alive all the time and tracks are initiated and terminated by observing the cluster looseness chosen to be 0.4. Fig. 6 shows the evolution of 5000 particles and the filter output. Our filter takes a little longer than the conventional TBD PF to declare absence of the target after the target has disappeared.



**Fig. 6:** Particle evolution of the PF using soft resampling. A target appears in the 8th frame and persists until the 24th frame. Noise variance at the camera 16. The blue line is the ground truth, magenta dots are the particles and the solid red line is the filter estimate.

We finally test the simplicity of our approaches. Fig. 7 gives the computational time to complete the above simulation versus the number of particles. Our techniques are more simple since we do not need to model the birth and death of the particles.



**Fig. 7:** Computational time vs number of particles plot. Observations are 2-D snapshots from a  $10 \times 10$  frame with a noise variance of 9. Results are averaged over 50 iterations of 30 frames each.

## 5. RELATION TO PRIOR WORK

Stochastic resampling techniques [3, 5, 11, 12] are extensively used in PFs to reduce particle degeneracy. However, they discard particles that may be gradually accumulating weight causing loss of potential information. These techniques also reset the weights equally [8, 16], with consequent loss of some of the information that could have been propagated to the next time step. Although deterministic resampling schemes overcome these limitations by allowing the information regarding the weights to survive the resampling process, the optimal resampling scheme of [13] does not reduce the variance among the weights and partial resampling [9] reweights the particles based on the number of dominant and negligible particles rather than on their individual weights. All these techniques are hard resamplers that are not feasible for direct use in TBD scenarios and necessitate modelling of the birth and death of particles [4, 14]. In this paper, we proposed two variants of a novel soft resampling method that achieves the simultaneous goals of maximising the retention of information over successive time steps, while minimizing the variance of the particle weights. The techniques can be used in a classical PF as a TBD PF making the process much simpler.

## 6. CONCLUSION

In this paper, we proposed two approaches for soft resampling that can be used in a PF. Through simulations, we first demonstrated that our algorithms retain a looser cluster, which then aids in a faster lock when a target appears. We also demonstrated that our techniques exhibit higher tracking efficiency because of redistributing the weights of the discarded particles. We then demonstrated that our technique can be used in a classical PF for TBD applications and established that the computational load is reduced since we do not explicitly model particle birth and death.

## 7. REFERENCES

- [1] R.P.S. Mahler, *Statistical Multisource-Multitarget Information Fusion*, vol. 685, Artech House Publishers, 2007.
- [2] W.M. Bolstad, *Introduction to Bayesian statistics*, WILEY-INTERSCIENCE, 2004.
- [3] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel approach to non-linear/non-Gaussian Bayesian state estimation," in *Proceedings IEE Radar and Signal Processing*, 1993, vol. 140, pp. 107–113.
- [4] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House Publishers, 2004.
- [5] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.
- [6] M.K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American Statistical Association*, pp. 590–599, 1999.
- [7] M. Orton and W. Fitzgerald, "A Bayesian approach to tracking multiple targets using sensor arrays and particle filters," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 216–223, 2002.
- [8] R. Douc and O. Cappé, "Comparison of resampling schemes for particle filtering," in *Proceedings ISPA 4th International Symposium on Image and Signal Processing and Analysis*, 2005, pp. 64–69.
- [9] M. Bolić and S. and Hong P.M. Djurić, "Resampling algorithms for particle filters: A computational complexity perspective," *EURASIP Journal of Applied Signal Processing*, vol. 2004, pp. 2267–2277, 2004.
- [10] D. Crisan, P. Del Moral, and T. Lyons, "Discrete filtering using branching and interacting particle systems," *Journal of Markov Processes and Related Fields*, vol. 5, no. 3, pp. 293–318, 1999.
- [11] J. Liu and R. Chen, "Sequential Monte Carlo Methods for Dynamic Systems," *Journal of the American Statistical Association*, vol. 93, pp. 1032–1044, 1998.
- [12] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics*, pp. 1–25, 1996.
- [13] P. Fearnhead and P. Clifford, "On-line inference for hidden Markov models via particle filters," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 65, pp. 887–899, 2003.
- [14] D.J. Salmond and H. Birch, "A particle filter for track-before-detect," in *Proceedings IEEE American Control Conference*, 2001, vol. 5, pp. 3755–3760.
- [15] M. Rollason and D. Salmond, "A particle filter for track-before-detect of a target with unknown amplitude," in *Proceedings IEE Target Tracking: Algorithms and Applications*, 2001, vol. 1, pp. 14/1–14/4.
- [16] Jeroen D. Hol, Thomas B. Schon, and Fredrik Gustafsson, "On Resampling Algorithms for Particle Filters," in *Proceedings Nonlinear Statistical Signal Processing Workshop*, 2006.