

# Sequential State-Space Filters for Speech Enhancement

S.Patil, S. Srinivasan, S. Prasad, R. Irwin, G. Lazarou and J. Picone

Center for Advanced Vehicular Systems

Mississippi State University

{patil, srinivas, prasad, rirwin, glaz, picone}@cavs.msstate.edu

## Abstract

*Particle filters have recently been proposed as a new form of state-space filtering for speech enhancement applications. Despite theoretical foundations that suggest superior performance, robust performance in practical applications has been elusive. This paper presents a comparative analysis of three popular recursive filtering algorithms that share a state-space formulation: Kalman filters, Unscented Kalman filters, and Particle filters. We present a general formulation of these state-space models and then introduce applications of these to time series prediction based on autoregressive models. Results indicate that for signals produced from linear systems, as expected, Particle filters and unscented Kalman filters do not perform significantly better than a Kalman filter. For typical speech signals, the traditional Kalman filter provides more robust performance at the same level of computational and representational complexity. At extremely low signal-to-noise ratios, unscented Kalman filter gave the best results.*

## 1. Introduction

Filtering is a necessary operation for almost all signal processing applications. The main attribute of a filtering algorithm is to remove noise from corrupted observations while retaining useful information. All signals received at the signal processing unit are corrupted by noise. Thus, an algorithm must estimate the noise component to improve the signal-to-noise ratio (SNR). For a signal with time-varying statistics, the algorithm must discover the underlying statistics and then adapt itself to the changing statistics, while at the same time continue to remove unwanted components of the signal such as noise. For problems such as speech processing, this is often an ill-posed problem because differentiating speech from noise requires higher-level linguistic knowledge.

A common approach to handling a time-varying signal is to segment the signal into smaller, stationary sections. Conventional signal processing algorithms that assume stationarity can then be applied to these segments. This is a common piecewise linear approach to signal processing that has been so successful to date on applications such as speech processing.

In this paper, we explore the properties of three such algorithms: (1) Kalman filter (KF): a recursive linear filter designed to minimize the mean-squared estimation error; (2) unscented Kalman filter (UKF): a Kalman filter designed specifically for nonlinear systems with Gaussian statistics; and (3) Particle filter (PF): a filter designed for nonlinear systems that allows arbitrary distributions for the underlying states and noise estimates.

This paper describes implementations of these algorithms and compares them on a speech enhancement task. Section 2 describes autoregressive modeling of noisy speech in a state-space framework. Section 3 describes recursive filters from an adaptive filtering point of view and introduces the three algorithms described in this paper. Section 4 presents our experimental results. We conclude the paper with comments on the results and an overview of future research directions.

## 2. AR Based Speech Model

A speech signal [1][2] is typically modeled as an autoregressive (AR) process of order  $p$ :

$$s(n) = \sum_{i=1}^p a_i(n) s(n-i) + u(n) \quad (1)$$

where  $s(n)$  is the  $n^{\text{th}}$  sample of the speech signal,  $a_i$  is the  $i^{\text{th}}$  AR coefficient and  $u(n)$  is the driving (process) noise. This formulation has been the backbone of various approaches to speech analysis for over 40 years.

For the filtering algorithms discussed in the paper, Equation (1) is implemented within a state-space framework. The state-space model is represented using two equations – a state evolution equation and an observation evolution equation [3][4]:

$$\begin{aligned} X_k &= F(X_{k-1}, W_{k-1}) \\ Y_k &= H(X_k, V_k) \end{aligned} \quad (2)$$

where  $X$  is the state vector;  $F$  is the state evolution function that relates the previous state vector and  $W$  (process noise) to the present state vector.  $Y$  represents the observations,  $V$  represents the measurement noise and  $H$  is the measurement function that relates the state vector to the observations. State-space modeling has been

a popular tool in control systems design, tracking problems etc.

The general state-space formulation of Equation (2) can be embedded in an AR-model framework as illustrated in the set of equations (3)

$$\begin{aligned} X_k &= F X_{k-1} + W_{k-1} \\ Y_k &= H X_k + V_k \end{aligned} \quad (3)$$

where

$$F = \begin{bmatrix} 0, & 1, & 0, & \dots, & 0 \\ 0, & 0, & 1, & \dots, & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ -a_p, & -a_{p-1}, & \dots, & -a_1 \end{bmatrix},$$

$$H = [0, 0, \dots, 1]$$

where  $X$  is the  $p \times 1$  state vector (filtered speech),  $F$  is the  $p \times p$  transition matrix,  $W$  is a  $p \times 1$  process noise vector,  $Y$  represents the observations (e.g., corrupted speech signal),  $H$  is a  $1 \times p$  observation matrix, and  $V$  is the observation noise.

### 3. Recursive Filtering

Recursive filtering, also known as sequential filtering is a class of algorithms that provide tools for filtering, prediction and smoothing of an input time series. This family of algorithms derives its name from the recursive updates of the old estimates of the states using the information contained in new data samples. The recursive estimation equation is given by Equation (4)

$$\hat{X}_k = \hat{X}_k^- + K_k (Y_k - H \hat{X}_k^-) \quad (4)$$

where all variables with a cap (^) refer to filtered (a posteriori) state estimates while variables with a superscript (-) refer to predicted estimates (not containing the information contained in the most recent observation sample).  $Y_k - H \hat{X}_k^-$  is the extra information provided by the  $k^{th}$  observation sample and is referred to as an innovation process.  $K$  is the weight for the innovation process in the update equation.

Various implementations (e.g., KF, UKF and PF) exist for performing recursive filtering. KF [4] was developed for spacecraft navigational systems and later employed for control system applications. UKF [5] was also developed for control system applications and has been applied to navigation controls for unmanned vehicles. PF [6] has been widely used for image processing (e.g. object tracking) and wireless communication applications. Sections 3.1 and 3.2 describe implementations for these three filters in detail.

### 3.1. Kalman Filter

Kalman filtering is a popular tool in the research community for estimating non-stationary processes when it is possible to model the system dynamics by linear behavior and Gaussian statistics. Kalman filters have been applied previously to applications where the corrupting process is additive white Gaussian noise. KF can also be used as a non-stationary process estimator, enabling us to recover a signal corrupted by a time-varying channel.

Kalman filters are linear recursive filters where a Gaussian random (state) vector is propagated through a linear state space model. The objective is to find a 'filtered' estimate of the state vector,  $X$ , at time  $k$ , represented as a linear combination of the measurements up to time  $k$  such that the following quadratic cost function is minimized:

$$E[(X_k - \hat{X}_k)^T M (X_k - \hat{X}_k)] \quad (5)$$

where  $M$  is a symmetric nonnegative definite weighting matrix.

The Kalman filtering framework can be summarized by the set of equations in (6):

$$\begin{aligned} \hat{X}_k^- &= F_k \hat{X}_{k-1} \\ P_k^- &= F_k P_{k-1} F_k^T + Q \\ K_k &= P_k^- H^T (H P_k^- H^T + R)^{-1} \\ \hat{X}_k &= \hat{X}_k^- + K_k (Y_k - H \hat{X}_k^-) \\ P_k &= (I - K_k H) P_k^- \end{aligned} \quad (6)$$

Here  $Q$  and  $R$  represent the covariance matrices of the process and measurement noise respectively. Conceptually, such a filtering proceeds in two steps – prediction and estimation. In the prediction step, the previous state vector is propagated through the state-space model to generate the prediction at any time. This prediction is then refined into an estimate by incorporating the weighted innovations term. The weight  $K_k$ , for the innovation term,  $Y_k - H \hat{X}_k^-$  is referred to as the Kalman Gain.

### 3.2. Unscented Kalman Filter and Particle Filter

Though Kalman filters have been successfully employed in a variety of applications, they fail to model nonlinear systems. UKF and PF extend the concept of sequential filtering to nonlinear settings. UKF overcomes the drawbacks of KF by estimating the statistics of the state vector propagated through the nonlinear state space model.

The key concept in UKF is the unscented transformation. Accurate statistics of the propagated

vector are estimated from a deterministically chosen set of sample points [3]. These carefully chosen points are called sigma points and are computed as described in Equation (7):

$$\begin{aligned}
& \text{Unscented Transformation} \\
& \chi_0 = \bar{X} \\
& \text{for } i = 1, \dots, L \\
& \quad \chi_i = \bar{X} + \left( \sqrt{(L + \lambda) P_X} \right)_i \\
& \text{for } i = L + 1, \dots, 2L \\
& \quad \chi_i = \bar{X} - \left( \sqrt{(L + \lambda) P_X} \right)_{i-L} \\
& W_0^{(m)} = \lambda / (L + \lambda) \\
& W_0^{(c)} = \lambda / (L + \lambda) + (1 - \alpha^2 + \beta) \\
& \text{for } i = 1, \dots, 2L \\
& W_i^{(c)} = W_i^{(m)} = 1 / \{2(L + \lambda)\}
\end{aligned} \tag{7}$$

where,  $X$  is a vector of length  $L$  for which the sigma points are computed, and  $W^m$  and  $W^c$  are weights associated with the sigma points (for estimating the mean vectors and covariance matrices.)

For an  $L$ -dimensional vector, we must choose  $2L + 1$  sigma points about the mean of the distribution. It has been shown that the choice of sigma points as depicted in the set of equations (7) provides an accurate estimate of the first two moments of the data. The values for  $\alpha$ ,  $\beta$  and  $\lambda$  are chosen such that the resulting estimate of the covariance matrix results in a positive definite matrix. In this paper, we have chosen  $\lambda = 3 - L$ .  $\alpha$  (specifying the spread of the sigma points about the mean) is usually set to a small positive number, and  $\beta = 2$  is optimal for Gaussian densities.

A Particle filter (PF) is another sequential state-space algorithm that is being increasingly applied to systems with nonlinear functional relationships in their state model. The main advantage of PF over UKF is that it does not constrain the statistics of the noise and the states to be Gaussian [1]-[1]. The PF algorithm is based on sequential Monte Carlo technique where probability distributions are represented in terms of randomly picked samples (particles) and these particles are then recursively updated using Bayesian estimation procedures.

The accuracy in representation of the probability distribution increases with the number of particles. Since direct generation of particles for arbitrary distributions is difficult, particles are generated for another distribution function (called the importance density function) and then appropriately scaled. The scaling factors are called importance weights. The importance function is chosen as one which facilitates easy drawing of samples. The

equations governing the prediction and update steps are given by equation (8):

$$\begin{aligned}
p(x_k | y_{1:k}) &= \frac{p(x_k | y_k) p(x_k | y_{1:k-1})}{p(y_k | y_{1:k-1})} \\
p(x_k | y_{1:k-1}) &= \int p(x_k | x_{k-1}) p(x_k | y_{1:k-1}) dx_{k-1}
\end{aligned} \tag{8}$$

Depending on the method used for generating and updating particles, there are several variants of PF. In this paper we use the Sampling Importance Resampling (SIR) procedure [6] for generating particles with an accept-reject mechanism to perform resampling.

## 4. Experimental Setup and Results

The algorithms were tested on two synthetic signals and a speech signal selected from the TIDIGTIS corpus. The first synthetic signal considered was generated by an AR process with defined parameters. The second synthetic series was a sinusoidal waveform. For all three test cases, the input SNR was varied from -10dB to +20dB. To build the state evolution matrix, the input signals were divided into frames of 20 ms duration. Each frame was modeled using a  $10^{th}$  order AR filter and filter coefficients were estimated using the Yule-Walker method [1]. For the first analysis frame, the state vector was initialized to the first  $p$  samples of the observation sequence. State vectors for subsequent frames were initialized to the last  $p$  filtered values of the previous frame, to ensure continuity between frames. For testing with PF, the number of particles used in each run was 100. The best result was reported for each input SNR case. The performance metric used for comparative analysis is the output Mean-Squared Error (MSE).

Table 1 illustrates the results obtained from the experiments described above. As can be seen from the table, for all the three test signals, MSE decreases with increasing SNR, irrespective of the algorithm used. The performance is poorest for sine waveform input, and this behavior can be explained by the fact that an AR model is inappropriate to predict samples from a sine waveform. Also for most SNR values, all the three algorithms perform better on the AR signals than on speech signals. The reason behind this trend is that an AR filter is only an approximate model for speech signals. Comparison of MSE performance of the different algorithms for each type of input signal shows that KF outperforms the other two algorithms at medium and high SNRs while UKF gives the best performance at very low SNRs.

**Table 1: Comparative results for all test conditions**

I/P SNR	AR modeled			Sine			Speech		
	KF	PF	UKF	KF	PF	UKF	KF	PF	UKF
-10.0	0.02930	0.01828	0.0064	0.05140	0.56070	0.13690	0.019437	0.018595	0.007900
-5.0	0.01280	0.00124	0.0052	0.01507	0.54080	0.08320	0.006589	0.006048	0.005100
0.0	0.00398	0.00517	0.0038	0.00537	0.19110	0.03660	0.002268	0.001954	0.002340
5.0	0.00157	0.00206	0.0026	0.00173	0.08828	0.02440	0.000737	0.000658	0.001100
10.0	0.00060	0.00076	0.0020	0.00064	0.03638	0.02070	0.000250	0.000252	0.000800
15.0	0.00017	0.00032	0.0017	0.00018	0.01293	0.01950	0.000076	0.000112	0.000070
20.0	0.00006	0.00055	0.0016	0.00006	0.00506	0.02060	0.000024	0.000067	0.000770

## 5. Conclusions and Future Work

KF, UKF and PF have been adopted as popular tools for sequential filtering of signals in harsh environments. In this paper, we have demonstrated the use of KF, UKF, and PF algorithms in speech enhancement applications assuming a linear AR model for speech. Results indicate that KF outperforms PF and UKF in clean signal conditions while UKF performs best for speech at very low SNRs. A Java applet [7] developed at the Intelligent Electronic Systems (IES) program at Mississippi State University that demonstrates these algorithms is available at [http://www.cavs.msstate.edu/hse/ies/projects/speech/software/demonstrations/applets/util/pattern\\_recognition/current/](http://www.cavs.msstate.edu/hse/ies/projects/speech/software/demonstrations/applets/util/pattern_recognition/current/).

Our future work in recursive filtering would include more extensive experimentation in a speech recognition application in which these filters are applied to both the signal and feature streams. It is hoped that we can improve the robustness of the recognition system to noise by using sequential filters in the front-end. In addition, we are also exploring variants of KF such as square root KF, central difference KF, and variants of PF such as Markov Chain Monte Carlo (MCMC) and unscented PF. One restriction of such state-space based filtering paradigms is that they require knowledge of the state-space model of the underlying system. In practice, this structure can be learned using a neural network.

## 6. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. IIS-0414450. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## 7. References

- [1] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, Englewood Cliffs, NJ, 1993.
- [2] M. Gabrea, "Robust Adaptive Kalman Filtering-Based Speech Enhancement Algorithm," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, pp-1301-304, May 2004.
- [3] E. Wan and R. Merwe, "The unscented Kalman Filter for Nonlinear Estimation," *Proceedings of the Adaptive Systems for Signal Processing, Communications and Control Symposium*, pp. 153-158, November 2005.
- [4] G. Welch and G. Bishop, "Introduction to the Kalman Filter," Technical Report TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, April 2004.
- [5] R. Merve, N. de Freitas, A. Doucet, and E. Wan, "The Unscented Particle Filter," Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department, Cambridge University, U.K, August 2000.
- [6] P. Djuric, J. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. Bugallo, and J. Miguez, "Particle Filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19-38, September 2003.
- [7] N. Arulampalam, S. Maskell, N. Gordan, and T. Clapp, "Tutorial On Particle Filters For Online Nonlinear/ Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174-188, February 2002.