MTR 05W0000004

MITRE TECHNICAL REPORT

A Tutorial on Bayesian Estimation and Tracking Techniques Applicable to Nonlinear and Non-Gaussian Processes

January 2005

A.J. Haug

Sponsor: MITRE MSR
Dept. No.: W400
Contract No.: W15P7T-04-D199
Project No.: 01MSR0115RT

MITRE

Corporate Headquarters
McLean, Virginia

MITRE Department Approval: _____

Dr. Frank Driscoll

MITRE Project Approval: _____

Dr. Garry Jacyna

Abstract

Nonlinear filtering is the process of estimating and tracking the state of a nonlinear stochastic system from non-Gaussian noisy observation data. In this technical memorandum, we present an overview of techniques for nonlinear filtering for a wide variety of conditions on the nonlinearities and on the noise. We begin with the development of a general Bayesian approach to filtering which is applicable to all linear or nonlinear stochastic systems. We show how Bayesian filtering requires integration over probability density functions that cannot be accomplished in closed form for the general nonlinear, non-Gaussian multivariate system, so approximations are required. Next, we address the special case where both the dynamic and observation models are nonlinear but the noises are additive and Gaussian. The extended Kalman filter (EKF) has been the standard technique usually applied here. But, for severe nonlinearities, the EKF can be very unstable and performs poorly. We show how to use the analytical expression for Gaussian densities to generate integral expressions for the mean and covariance matrices needed for the Kalman filter which include the nonlinearities directly inside the integrals. Several numerical techniques are presented that give approximate solutions for these integrals, including Gauss-Hermite quadrature, unscented filter, and Monte Carlo approximations. We then show how these numerically generated integral solutions can be used in a Kalman filter so as to avoid the direct evaluation of the Jacobian matrix associated with the extended Kalman filter. For all filters, step-by-step block diagrams are used to illustrate the recursive implementation of each filter. To solve the fully nonlinear case, when the noise may be non-additive or non-Gaussian, we present several versions of particle filters that use importance sampling. Particle filters can be subdivided into two categories: those that re-use particles and require resampling to prevent divergence, and those that do not re-use particles and therefore require no resampling. For the first category, we show how the use of importance sampling, combined with particle re-use at each iteration, leads to the sequential importance sampling (SIS) particle filter and its special case, the bootstrap particle filter. The requirement for resampling is outlined and an efficient resampling scheme is presented. For the second class, we discuss a generic importance sampling particle filter and then add specific implementations, including the Gaussian particle filter and combination particle filters that bring together the Gaussian particle filter, and either the Gauss-Hermite, unscented, or Monte Carlo Kalman filters developed above to specify a Gaussian importance density. When either the dynamic or observation models are linear, we show how the Rao-Blackwell simplifications can be applied to any of the filters presented to reduce computational costs. We then present results for two nonlinear tracking examples, one with additive Gaussian noise and one with non-Gaussian embedded noise. For each example, we apply the appropriate nonlinear filters and compare performance results.

Acknowledgement

The author would like to thank Drs. Roy Bethel, Chuck Burmaster, Carol Christou Garry Jacyna and for their review and many helpful comments and suggestions that have contributed to the clarity of this report. Special thanks to Roy Bethel for his help with Appendix A and to Garry Jacyna for his extensive work on the likelihood function development for DIFAR sensors found in Appendix B.

1. Introduction

Nonlinear filtering problems abound in many diverse fields including economics, biostatistics, and numerous statistical signal and array processing engineering problems such as time series analysis, communications, radar and sonar target tracking, and satellite navigation. The filtering problem consists of recursively estimating, based on a set of noisy observations, at least the first two moments of the state vector governed by a dynamic nonlinear non-Gaussian state space model (DSS). A discrete time DSS consists of a stochastic propagation (prediction or dynamic) equation which links the current state vector to the prior state vector and a stochastic observation equation that links the observation data to the current state vector. In a Bayesian formulation, the DSS specifies the conditional density of the state given the previous state and that of the observation given the current state. When the dynamic and observation equations are linear and the associated noises are Gaussian, the optimal recursive filtering solution is the Kalman filter [1]. The most widely used filter for nonlinear systems with Gaussian additive noise is the well known extended Kalman filter (EKF) which requires the computation of the Jacobian matrix of the state vector [2]. However, if the nonlinearities are significant, or the noise is non-Gaussian, the EKF gives poor performance (see [3] and [4], and the references contained therein.) Other early approaches to the study of nonlinear filtering can be found in [2] and [5].

Recently, several new approaches to recursive nonlinear filtering have appeared in the literature. These include grid-based methods [3], Monte Carlo methods, Gauss quadrature methods [6]-[8] and the related unscented filter [4], and particle filter methods [3], [7], [9]-[13]. Most of these filtering methods have their basis in computationally intensive numerical integration techniques that have been around for a long time but have become popular again due to the exponential increase in computer power over the last decade.

In this paper, we will review some of the recently developed filtering techniques applicable to a wide variety of nonlinear stochastic systems in the presence of both additive Gaussian and non-Gaussian noise. We begin in Section 2 with the development of a general Bayesian approach to filtering, which is applicable to both linear and nonlinear stochastic systems, and requires the evaluation of integrals over probability and probability-like density functions. The integrals inherent in such a development cannot be solved in closed form for the general multi-variate case, so integration approximations are required.

In Section 3, the noise for both the dynamic and observation equations is assumed to be additive and Gaussian, which leads to efficient numerical integration approximations. It is shown in Appendix A that the Kalman filter is applicable for cases where both the dynamic and measurement noise are additive and Gaussian, without any assumptions on the linearity of the dynamic and measurement equations. We show how to use analytical expressions for Gaussian densities to generate integral expressions for the mean and covariance matrices needed for the Kalman filter, which include the nonlinearities directly inside the integrals. The most widely used numerical approximations used to evaluate these integrals include Gauss-Hermite quadrature, the unscented filter, and Monte Carlo integration. In all three approximations, the integrals are replaced by discrete finite sums,

leading to a nonlinear approximation to the kalman filter which avoids the direct evaluation of the Jacobian matrix associated with the extended Kalman filter. The three numerical integration techniques, combined with a Kalman filter, result in three numerical nonlinear filters: the Gauss-Hermite Kalman filter (GHKF), the unscented Kalman filter (UKF) and the Monte Carlo Kalman filter (MCKF).

Section 4 returns to the general case and shows how it can be reformulated using recursive particle filter concepts to offer an approximate solution to nonlinear/non-Gaussian filtering problems. To solve the fully nonlinear case, when the noise may be non-additive and/or non-Gaussian, we present several versions of particle filters that use importance sampling. Particle filters can be subdivided into two categories: those that re-use particles and require resampling to prevent divergence, and those that do not re-use particles and therefore require no resampling. For the particle filters that require resampling, we show how the use of importance sampling, combined with particle re-use at each iteration, leads to the sequential importance sampling particle filter (SIS PF) and its special case, the bootstrap particle filter (BPF). The requirement for resampling is outlined and an efficient resampling scheme is presented. For particle filters requiring no resampling, we discuss a generic importance sampling particle filter and then add specific implementations, including the Gaussian particle filter and combination particle filters that bring together the Gaussian particle filter, and either the Gauss-Hermite, unscented, or Monte Carlo Kalman filters developed above to specify a Gaussian importance density from which samples are drawn. When either the dynamic or observation models are linear, we show how the Rao-Blackwell simplifications can be applied to any of the filters presented to reduce computational costs [14]. A roadmap of the nonlinear filters presented in Sections 2 through 4 is shown in Fig. 1.

In Section 5 we present an example in which the noise is assumed additive and Gaussian. In the past, the problem of tracking the geographic position of a target based on noisy passive array sensor data mounted on a maneuvering observer has been solved by breaking the problem into two complementary parts: tracking the *relative* bearing using noisy *narrowband* array sensor data [15], [16] and tracking the geographic position of a target from noisy bearings-only measurements [10], [17], [18]. In this example, we formulate a new approach to single target tracking in which we use the sensor outputs of a passive ring array mounted on a maneuvering platform as our observations, and recursively estimate the position and velocity of a constant-velocity target in a fixed geographic coordinate system. First, the sensor observation model is extended from narrowband to broadband. Then, the complex sensor data are used in a Kalman filter that estimates the geo-track updates directly, without first updating relative target bearing. This solution is made possible by utilizing an observation model that includes the highly nonlinear geographic-to-array coordinate transformation and a second complex-to-real transformation. For this example we compare the performance results of the Gauss-Hermite quadrature, the unscented, and the Monte Carlo Kalman filters developed in Section 3.

A second example is presented in Section 6 in which a constant-velocity vehicle is tracked through a field of DIFAR (Directional Frequency Analysis and Recording) sensors. For this problem, the observation noise is non-Gaussian and embedded in the nonlinear

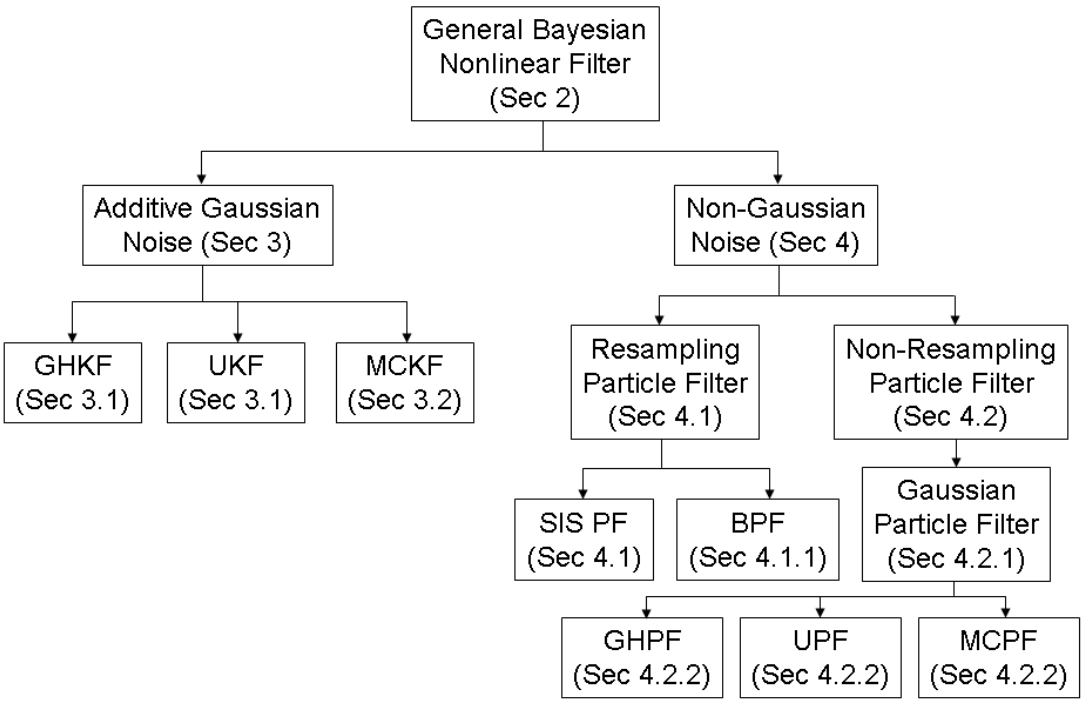

Figure 1: Roadmap to Techniques developed in Sections 2 Through 4.

observation equation, so it is an ideal application of a particle filter. All of the particle filters presented in Section 4 are applied to this problem and their results are compared. All particle filter applications require an analytical expression for the likelihood function, so Appendix B presents the development of the likelihood function for a DIFAR sensor for target signals with bandwidth-time products much greater than one.

Our summary and conclusions are found in Section 7. In what follows, we treat bold small $\mathbf{x}$ and large $\mathbf{Q}$ letters as vectors and matices, respectively, with $[\cdot]^{\mathrm{H}}$ representing the complex conjugate transpose of a vector or matrix, $[\cdot]^{\intercal}$ representing just the transpose and $\langle\cdot\rangle$ or $\mathcal{E}(\cdot)$ used as the expectation operator. It should be noted that this tutorial assumes that the reader is well versed in the use of Kalman and extended Kalman filters.

2. General Bayesian Filter

A nonlinear stochastic system can be defined by a stochastic discrete-time state space transition (dynamic) equation

$$\mathbf{x}_n = \mathbf{f}_n(\mathbf{x}_{n-1}, \mathbf{w}_{n-1}), \tag{1}$$

and the stochastic observation (measurement) process

$$\mathbf{y}_n = \mathbf{h}_n(\mathbf{x}_n, \mathbf{v}_n), \tag{2}$$

where at time t_n, $\mathbf{x}_n$ is the (usually hidden or not observable) system state vector, $\mathbf{w}_n$ is the dynamic noise vector, $\mathbf{y}_n$ is the real (in comparison to complex) observation vector and $\mathbf{v}_n$ is the observation noise vector. The deterministic functions $\mathbf{f}_n$ and $\mathbf{h}_n$ link the prior state to the current state and the current state to the observation vector, respectively. For complex observation vectors, we can always make them real by doubling the observation vector dimension using the in-phase and quadrature parts (see Appendix A.)

In a Bayesian context, the problem is to quantify the posterior density $p(\mathbf{x}_n|\mathbf{y}_{1:n})$, where the observations are specified by $\mathbf{y}_{1:n} \triangleq \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$. The above nonlinear non-Gaussian state-space model, Eq. 1, specifies the predictive conditional transition density, $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_{1:n-1})$, of the current state given the *previous* state and all *previous* observations. Also, the observation process equation, Eq. 2, specifies the likelihood function of the current observation given the current state, $p(\mathbf{y}_n|\mathbf{x}_n)$. The prior probability, $p(\mathbf{x}_n|\mathbf{y}_{1:n-1})$, is defined by Bayes' rule as

$$p(\mathbf{x}_n|\mathbf{y}_{1:n-1}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_{1:n-1})\, p(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1})\, d\mathbf{x}_{n-1}. \tag{3}$$

Here, the previous posterior density is identified as $p(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1})$.

The correction step generates the posterior probability density function from

$$p(\mathbf{x}_n|\mathbf{y}_{1:n}) = cp(\mathbf{y}_n|\mathbf{x}_n)\, p(\mathbf{x}_n|\mathbf{y}_{1:n-1}), \tag{4}$$

where c is a normalization constant.

The filtering problem is to estimate, in a recursive manner, the first two moments of $\mathbf{x}_n$ given $\mathbf{y}_{1:n}$. For a general distribution, $p(\mathbf{x})$, this consists of the recursive estimation of the expected value of any function of $\mathbf{x}$, say $\langle g(\mathbf{x}) \rangle_{p(\mathbf{x})}$, using Eq's. 3 and 4 and requires calculation of integrals of the form

$$\langle g(\mathbf{x}) \rangle_{p(\mathbf{x})} = \int g(\mathbf{x}) \, p(\mathbf{x}) \, d\mathbf{x}. \tag{5}$$

But for a general multivariate distribution these integrals cannot be evaluated in closed form, so some form of integration approximation must be made. This memorandum is primarily concerned with a variety of numerical approximations for solving integrals of the form given by Eq. 5.

3. The Gaussian Approximation

Consider the case where the noise is additive and Gaussian, so that Eq's. 1 and 2 can be written as

$$\mathbf{x}_n = \mathbf{f}_n(\mathbf{x}_{n-1}) + \mathbf{w}_{n-1}, \tag{6}$$

and

$$\mathbf{y}_n = \mathbf{h}_n(\mathbf{x}_n) + \mathbf{v}_n, \tag{7}$$

where $\mathbf{w}_n$ and $\mathbf{v}_n$ are *modeled* as independent Gaussian random variables with mean $\mathbf{0}$ and covariances $\mathbf{Q}_n$ and $\mathbf{R}_n$, respectively. The initial state $\mathbf{x}_0$ is also modeled as a stochastic variable, which is independent of the noise, with mean $\widehat{\mathbf{x}}_0$ and covariance $\mathbf{P}_0^{\mathbf{xx}}$.

Now, assuming that deterministic functions $\mathbf{f}$ and $\mathbf{h}$, as well as the covariance matrices $\mathbf{Q}$ and $\mathbf{R}$, are not dependent on time, from Eq. 6 we can identify the predictive conditional density as

$$p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_{1:n-1}) = \mathcal{N}(\mathbf{x}_n; \mathbf{f}(\mathbf{x}_{n-1}), \mathbf{Q}), \tag{8}$$

where the general form of the multivariate Gaussian distribution $\mathcal{N}(\mathbf{t}; \mathbf{s}, \mathbf{\Sigma})$ is defined by

$$\mathcal{N}(\mathbf{t}; \mathbf{s}, \mathbf{\Sigma}) \triangleq \frac{1}{\sqrt{(2\pi)^n \|\mathbf{\Sigma}\|}} \exp\left\{ -\frac{1}{2} [\mathbf{t} - \mathbf{s}]^{\intercal} (\mathbf{\Sigma})^{-1} [\mathbf{t} - \mathbf{s}] \right\} \tag{9}$$

We can now write Eq. 3 as

$$p(\mathbf{x}_n|\mathbf{y}_{1:n-1}) = \int \mathcal{N}(\mathbf{x}_n; \mathbf{f}(\mathbf{x}_{n-1}), \mathbf{Q}) \, p(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}) \, d\mathbf{x}_{n-1}. \tag{10}$$

Much of the Gaussian integral formulation shown below is a recasting of the material found in Ito, et. al. [6]. For the Gaussian distribution $\mathcal{N}(\mathbf{t}; \mathbf{f}(\mathbf{s}), \mathbf{\Sigma})$, we can write the expected value of $\mathbf{t}$ as

$$\langle \mathbf{t} \rangle \triangleq \int \mathbf{t} \mathcal{N}(\mathbf{t}; \mathbf{f}(\mathbf{s}), \mathbf{\Sigma}) \, d\mathbf{t} = \mathbf{f}(\mathbf{s}). \tag{11}$$

Using Eq. 10, it immediately follows that

$$\begin{aligned}
\langle \mathbf{x}_n | \mathbf{y}_{1:n-1} \rangle &\triangleq \mathcal{E}\left\{ \mathbf{x}_n | \mathbf{y}_{1:n-1} \right\} \\
&= \int \mathbf{x}_n p\left(\mathbf{x}_n | \mathbf{y}_{1:n-1} \right) d\mathbf{x}_n \\
&= \int \mathbf{x}_n \left[\int \mathcal{N}\left(\mathbf{x}_n ; \mathbf{f}\left(\mathbf{x}_{n-1} \right), \mathbf{Q} \right) p\left(\mathbf{x}_{n-1} | \mathbf{y}_{1:n-1} \right) d\mathbf{x}_{n-1} \right] d\mathbf{x}_n \\
&= \int \left[\int \mathbf{x}_n \mathcal{N}\left(\mathbf{x}_n ; \mathbf{f}\left(\mathbf{x}_{n-1} \right), \mathbf{Q} \right) d\mathbf{x}_n \right] p\left(\mathbf{x}_{n-1} | \mathbf{y}_{1:n-1} \right) d\mathbf{x}_{n-1} \\
&= \int \mathbf{f}\left(\mathbf{x}_{n-1} \right) p\left(\mathbf{x}_{n-1} | \mathbf{y}_{1:n-1} \right) d\mathbf{x}_{n-1}, \qquad (12)
\end{aligned}$$

where Eq. 11 was used to evaluate the inner integral above.

Now, assume that

$$p\left(\mathbf{x}_{n-1} | \mathbf{y}_{1:n-1} \right) = \mathcal{N}\left(\mathbf{x}_{n-1} ; \widehat{\mathbf{x}}_{n-1|n-1}, \mathbf{P}^{\mathbf{xx}}_{n-1|n-1} \right), \qquad (13)$$

where $\widehat{\mathbf{x}}_{n-1|n-1}$ and $\mathbf{P}^{\mathbf{xx}}_{n-1|n-1}$ are estimates of the mean and covariance of $\mathbf{x}_{n-1}$, given $\mathbf{y}_{1:n-1}$, respectively. Estimates of the mean and covariance of $\mathbf{x}_n$, given $\mathbf{y}_{1:n-1}$, $\widehat{\mathbf{x}}_{n|n-1}$ and $\mathbf{P}^{\mathbf{xx}}_{n|n-1}$, respectively, can now be obtained from Eq. 12 as follows

$$\widehat{\mathbf{x}}_{n|n-1} = \int \mathbf{f}\left(\mathbf{x}_{n-1} \right) \mathcal{N}\left(\mathbf{x}_{n-1} ; \widehat{\mathbf{x}}_{n-1|n-1}, \mathbf{P}^{\mathbf{xx}}_{n-1|n-1} \right) d\mathbf{x}_{n-1}, \qquad (14)$$

and

$$\begin{aligned}
\mathbf{P}^{\mathbf{xx}}_{n|n-1} = \mathbf{Q} &+ \int \mathbf{f}\left(\mathbf{x}_{n-1} \right) \mathbf{f}^{\intercal}\left(\mathbf{x}_{n-1} \right) \mathcal{N}\left(\mathbf{x}_{n-1} ; \widehat{\mathbf{x}}_{n-1|n-1}, \mathbf{P}^{\mathbf{xx}}_{n-1|n-1} \right) d\mathbf{x}_{n-1} \\
&- \widehat{\mathbf{x}}^{\intercal}_{n|n-1} \widehat{\mathbf{x}}_{n|n-1}. \qquad (15)
\end{aligned}$$

The expected value of $\mathbf{y}_n$, given $\mathbf{x}_n$ and $\mathbf{y}_{1:n-1}$, can be obtained from

$$\begin{aligned}
\langle \mathbf{y}_n | \mathbf{x}_n, \mathbf{y}_{1:n-1} \rangle &\triangleq \mathcal{E}\left\{ \mathbf{y}_n | \mathbf{x}_n, \mathbf{y}_{1:n-1} \right\} \\
&= \int \mathbf{y}_n p\left(\mathbf{x}_n | \mathbf{y}_{1:n-1} \right) d\mathbf{x}_n. \qquad (16)
\end{aligned}$$

Now, if we use a Gaussian approximation of $p\left(\mathbf{x}_n | \mathbf{y}_{1:n-1} \right)$ given by

$$p\left(\mathbf{x}_n | \mathbf{y}_{1:n-1} \right) = \mathcal{N}\left(\mathbf{x}_n ; \widehat{\mathbf{x}}_{n|n-1}, \mathbf{P}^{\mathbf{xx}}_{n|n-1} \right), \qquad (17)$$

we can obtain an estimate, $\widehat{\mathbf{y}}_{n|n-1}$, of $\langle \mathbf{y}_n | \mathbf{x}_n, \mathbf{y}_{1:n-1} \rangle$ from

$$\begin{aligned}
\widehat{\mathbf{y}}_{n|n-1} &= \int \mathbf{y}_n \mathcal{N}\left(\mathbf{x}_n ; \widehat{\mathbf{x}}_{n|n-1}, \mathbf{P}^{\mathbf{xx}}_{n|n-1} \right) d\mathbf{x}_n \\
&= \int \mathbf{h}\left(\mathbf{x}_n \right) \mathcal{N}\left(\mathbf{x}_n ; \widehat{\mathbf{x}}_{n|n-1}, \mathbf{P}^{\mathbf{xx}}_{n|n-1} \right) d\mathbf{x}_n. \qquad (18)
\end{aligned}$$

If we let $e^{y}_{n|n-1} \triangleq \mathbf{h}(\mathbf{x}_n) - \widehat{\mathbf{y}}_{n|n-1}$, we can also estimate the covariance of $\mathbf{y}_n$, given $\mathbf{x}_n, \mathbf{y}_{1:n-1}$, from

$$\begin{aligned}\mathbf{P}^{\mathbf{yy}}_{n|n-1} &= \left\langle \left[e^{y}_{n|n-1}\right]\left[e^{y}_{n|n-1}\right]^{\mathsf{T}} \right\rangle \\ &= \mathbf{R} + \int \mathbf{h}(\mathbf{x}_n)\,\mathbf{h}(\mathbf{x}_n)\,\mathcal{N}\left(\mathbf{x}_n; \widehat{\mathbf{x}}_{n|n-1}, \mathbf{P}^{\mathbf{xx}}_{n|n-1}\right) d\mathbf{x}_n \\ &\quad - \widehat{\mathbf{y}}^{\mathsf{T}}_{n|n-1}\widehat{\mathbf{y}}^{\mathsf{T}}_{n|n-1}. \end{aligned} \tag{19}$$

In addition, we can use the same technique to estimate the cross-covariance matrix $\mathbf{P}^{\mathbf{xy}}_{n|n-1}$ from

$$\begin{aligned}\mathbf{P}^{\mathbf{xy}}_{n|n-1} &= \left\langle \left[\mathbf{x}_n - \widehat{\mathbf{x}}_{n|n-1}\right]\left[e^{y}_{n|n-1}\right]^{\mathsf{T}} \right\rangle \\ &= \int \mathbf{x}_n \mathbf{h}^{\mathsf{T}}(\mathbf{x}_n)\,\mathcal{N}\left(\mathbf{x}_n; \widehat{\mathbf{x}}_{n|n-1}, \mathbf{P}^{\mathbf{xx}}_{n|n-1}\right) d\mathbf{x}_n \\ &\quad - \widehat{\mathbf{x}}_{n|n-1}\widehat{\mathbf{y}}^{\mathsf{T}}_{n|n-1}. \end{aligned} \tag{20}$$

In Appendix A, we show that the Kalman filter is applicable to any DSS where both the dynamic and observation models have additive Gaussian noise, regardless of the non-linearities in the models. Therefore, we can use the Kalman filter to construct a Gaussian approximation of the posterior density $p\left(\mathbf{x}_{n|n}\right)$ with mean and covariance given by

$$\widehat{\mathbf{x}}_{n|n} = \widehat{\mathbf{x}}_{n|n-1} + \mathbf{K}_n\left[\mathbf{y}_n - \widehat{\mathbf{y}}_{n|n-1}\right], \tag{21}$$

and

$$\mathbf{P}^{\mathbf{xx}}_{n|n} = \mathbf{P}^{\mathbf{xx}}_{n|n-1} - \mathbf{K}_n \mathbf{P}^{\mathbf{yy}}_{n|n-1}\mathbf{K}^{\mathsf{T}}_n, \tag{22}$$

where the Kalman gain $\mathbf{K}_n$ is given by

$$\mathbf{K}_n = \mathbf{P}^{\mathbf{xy}}_{n|n-1}\left[\mathbf{P}^{\mathbf{yy}}_{n|n-1}\right]^{-1}. \tag{23}$$

Note that the only approximation to this point in the development is that the noise be modeled as additive and Gaussian. So the above formulation generates $\widehat{\mathbf{x}}_{n|n}$ and $\mathbf{P}^{\mathbf{xx}}_{n|n}$ without any approximations. In order to implement this filter, however, we must develop approximation methods to evaluate the integrals in Eq's. 14, 15 and 18-20, which are of the form

$$I = \int \mathbf{g}(\mathbf{x})\,\mathcal{N}(\mathbf{x}; \widehat{\mathbf{x}}, \mathbf{P}^{\mathbf{xx}})\,d\mathbf{x}, \tag{24}$$

where $\mathcal{N}(\mathbf{x}; \widehat{\mathbf{x}}, \mathbf{P}^{\mathbf{xx}})$ is a multivariate Gaussian distribution with mean $\widehat{\mathbf{x}}$ and covariance $\mathbf{P}^{\mathbf{xx}}$.

In the subsections below, we will present three approximations to the integral given in Eq. 24. The first is a Gauss-Hermite quadrature approximation to the integral which also results in a weighted sum of support points of the integral, where both the weights and support points are predetermined and related to the first and second moments of

the probability density function (PDF). The second approximation is given by the unscented transform, which is a modification of a Gauss-Hermite quadrature approximation. The last is a Monte Carlo approximation in which random samples (support points) $\{\mathbf{x}^i, i = 1, 2, \ldots, N_s\}$ are generated from $\mathcal{N}(\mathbf{x}; \widehat{\mathbf{x}}, \mathbf{P}^{\mathbf{xx}})$ and the integral is evaluated as the sample mean. All of these approximations result in the propagation of the PDF support points through the nonlinearity $\mathbf{g}(\mathbf{x})$ and the resulting outputs summed after multiplication with the appropriate weights.

3.1. Numerical Integration Using Gauss-Hermite Quadrature or The Unscented Transform

Following the work first presented in [6], we can write Eq. 24 explicitly as

$$I = \int \mathbf{g}(\mathbf{x}) \frac{1}{\left[(2\pi)^n \|\mathbf{\Sigma}\|\right]^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \widehat{\mathbf{x}})^{\intercal} \mathbf{\Sigma}^{-1} (\mathbf{x} - \widehat{\mathbf{x}})\right\} d\mathbf{x}. \tag{25}$$

Let $\mathbf{\Sigma} = \mathbf{S}^{\intercal}\mathbf{S}$ using a Cholesky decomposition, and define

$$\mathbf{z} \triangleq \frac{1}{\sqrt{2}} \mathbf{S}^{-1} (\mathbf{x} - \widehat{\mathbf{x}}). \tag{26}$$

Then, noting that the state vector $\mathbf{x}$ is of dimension n, Eq. 25 reduces to

$$I = \frac{\sqrt{2}}{(2\pi)^{n/2}} \int \mathbf{g}(\mathbf{z}) e^{-\mathbf{z}^{\intercal}\mathbf{z}} d\mathbf{z}. \tag{27}$$

For the univariate case, $n = 1$ and $z = (x - \widehat{x}) / \left(\sqrt{2}\sigma\right)$ and Eq. 27 becomes

$$\mathcal{I} = \pi^{-1/2} \int_{-\infty}^{\infty} f(z) e^{-z^2} dz. \tag{28}$$

Eq. 28 can be approximated by the well known Gauss-Hermite quadrature rule [19] of the form

$$\int_{-\infty}^{\infty} e^{-z^2} f(z) dz \simeq \sum_{i=1}^{M} w_i f(z_i). \tag{29}$$

The quadrature points z_i and weights w_i can be determined as follows [20]-[22]. A set of orthonormal Hermite polynomials, $H_j(t)$, can be generated from the recurrence relationship

$$\begin{aligned} H_{-1}(t) &= 0, \quad H_0(t) = 1/\pi^{1/4}, \\ H_{j+1}(z) &= z\sqrt{\frac{2}{j+1}} H_j(z) - \sqrt{\frac{j}{j+1}} H_{j-1}(z); \quad j = 0, 1, \ldots, M. \end{aligned} \tag{30}$$

Letting $\beta_j \triangleq \sqrt{j/2}$, and rearranging terms yields

$$z H_j(z) = \beta_j H_{j-1}(z) + \beta_{j+1} H_{j+1}(z). \tag{31}$$

Eq. 31 can now be written in matrix form as

$$z\mathbf{h}(z) = \mathbf{J}_M \mathbf{h}(z) + \beta_M H_M(z)\,\mathbf{e}_M, \tag{32}$$

where

$$\begin{aligned} \mathbf{h}(z) &= [H_0(z), H_1(z), \ldots, H_{M-1}(z)]^{\intercal}, & (33)\\ \mathbf{e}_M &= [0, 0, \ldots, 1]^{\intercal}, & (34) \end{aligned}$$

and $\mathbf{J}_M$ is the $M \times M$ symmetric tridiagonal matrix

$$\mathbf{J}_M = \begin{bmatrix} 0 & \beta_1 & & & & \\ \beta_1 & 0 & \beta_2 & & \mathbf{0} & \\ & \beta_2 & 0 & & & \\ & & & \ddots & & \\ & \mathbf{0}^{\intercal} & & & 0 & \beta_{M-1} \\ & & & & \beta_{M-1} & 0 \end{bmatrix}. \tag{35}$$

.

The eigenvectors of $\mathbf{J}_M$ are vectors that, when multiplied by $\mathbf{J}_M$, generate vectors in the same direction but with a new length. The factor by which the length changes is the corresponding eigenvalue. By convention, the eigenvectors are orthonormal. So, if the term on the far right of Eq. 32 were not there, $\mathbf{h}(z)$ would be an eigenvector with corresponding eigenvalue z.

If Eq. 32 is evaluated for those values of z for which $H_M(z) = 0$, the unwanted term vanishes, and this equation determines the eigenvectors of $\mathbf{J}_M$ for the eigenvalues that are the M roots, z_i, of $H_M(z)$, with $i = 1, 2, \ldots, M$. The eigenvectors are given by

$$v_j^i = H_j(z_i) / \sqrt{W_i}, \tag{36}$$

where the normalizing constant $\sqrt{W_i}$ is given by

$$W_i = \sum_{j=0}^{M-1} H_j^2(z_i). \tag{37}$$

Now, the orthogonality and completeness conditions of the eigenvectors can be expressed as

$$\sum_{j=0}^{M-1} v_j^i v_j^k = \delta_{ik}, \tag{38}$$

and

$$\sum_{i=1}^{M} v_j^i v_l^i = \sum_{i=1}^{M} H_j(z_i) H_l(z_i) / W_i = \delta_{jl}. \tag{39}$$

Comparing Eq. 39 with the orthogonality relationship for the Hermite polynomials given by

$$\int_{-\infty}^{\infty} dzw\left(z\right) H_j\left(z\right) H_l\left(z\right) = \delta_{jl}, \tag{40}$$

we can see that in the discrete space, the weights $1/W_i$ replace the continuous weight $dzw\left(z\right)$ for functions evaluated at z_i. In addition, for products of polynomials up to order M, this quadrature will yield exact results. The integral of the product $H_M\left(z\right) H_{M-1}\left(z\right)$ will also be zero, because $H_M\left(z\right)$ vanishes on the nodes. Since any polynomial of order $2M-2$ can be written as a sum of products of pairs of polynomials up to order $M-1$, for any polynomial of order $2M-1$ or less, the quadrature equations will yield exact results. That is, Eq. 29 is valid for $w_i = 1/W_i$, with W_i given by Eq. 37 and z_i given by the eigenvalues of $\mathbf{J}_M$.

For the univariate case with $M = 3$, $\{z_1, z_2, z_3\} = \left\{-\sqrt{3/2}, 0, \sqrt{3/2}\right\}$ and $\{q_1, q_2, q_3\} \triangleq \pi^{-1/2}\left\{w_1, w_2, w_3\right\} = \{1/6, 2/3, 1/6\}$. Since $x_i = \widehat{x} + \sqrt{2} z_i \sigma$, Eq. 28 becomes

$$\mathcal{I} = \pi^{-1/2} \int_{-\infty}^{\infty} f\left(z\right) e^{-z^2} dz \simeq \sum_{i=1}^{3} q_i f\left(x_i\right). \tag{41}$$

By re-indexing, $\mathcal{I}$ can be evaluated as

$$\mathcal{I} \simeq \sum_{j=0}^{2} q_j f\left(x_j\right), \tag{42}$$

where

$$\begin{aligned} x_0 &= \widehat{x} \\ x_1 &= \widehat{x} + \sqrt{3}\sigma \\ x_2 &= \widehat{x} - \sqrt{3}\sigma. \end{aligned} \tag{43}$$

with $q_0 = 2/3$ and $q_1 = q_2 = 1/6$.

The mathematical theory of Gaussian quadrature described above is inherently one-dimensional. For the multivariate case, it must be applied sequentially, one state variable at a time. The weights in Eq. 41 will then be products of weights from each of the n variables. With $M = 3$ and an n-dimensional state vector, it follows from Eq. 27 that

$$\begin{aligned} I &= \frac{\sqrt{2}}{\left(2\pi\right)^{n/2}} \int \mathbf{g}\left(\mathbf{z}\right) e^{-\mathbf{z}^{\intercal}\mathbf{z}} d\mathbf{z} \\ &= \sqrt{2} \sum_{i_1=1}^{3} \cdots \sum_{i_n=1}^{3} \mathbf{g}\left(x_{i_1}, x_{i_2}, \ldots, x_{i_n}\right) p_{i_1} p_{i_2} \ldots p_{i_n}. \end{aligned} \tag{44}$$

where $p_{i_n} \triangleq q_{i_n}/\sqrt{2}$.

When $\mathbf{g}(\mathbf{z}) = \mathbf{1}$, Eq. 44 is the integral of the multivariate Gaussian probability distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and must therefore integrate to 1. Thus, we must apply the normalization criteria

$$\widetilde{p}_{j_i} = \frac{p_{j_i}}{\sqrt{2}\sum\cdots\sum p_{j_1}\cdots p_{j_n}}. \tag{45}$$

For a two-dimensional state vector, after reindexing and weight normalization, Eq. 44 can be written as

$$I_2 = \sum_{j=0}^{8} \mathbf{g}(\mathbf{x}_j)\,\alpha_j, \tag{46}$$

with the quadrature points given by

$$\begin{aligned}
\mathbf{x}_0 &= \widehat{\mathbf{x}} \\
\mathbf{x}_j &= \widehat{\mathbf{x}} + \sqrt{3}\left(\mathbf{\Sigma}^{1/2}\right)_j, \quad j = 1,2 \\
\mathbf{x}_j &= \widehat{\mathbf{x}} - \sqrt{3}\left(\mathbf{\Sigma}^{1/2}\right)_{j-2}, \quad j = 3,4 \\
\mathbf{x}_j &= \widehat{\mathbf{x}} + \sqrt{3}\left(\mathbf{\Sigma}^{1/2}\right)_1 + (-1)^{j-1}\sqrt{3}\left(\mathbf{\Sigma}^{1/2}\right)_2, \quad j = 5,6 \\
\mathbf{x}_j &= \widehat{\mathbf{x}} - \sqrt{3}\left(\mathbf{\Sigma}^{1/2}\right)_1 + (-1)^{j-1}\sqrt{3}\left(\mathbf{\Sigma}^{1/2}\right)_2, \quad j = 7,8,
\end{aligned} \tag{47}$$

and the normalized weights $\{\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8\}$ are given by $\{4/9, 1/9, 1/9, 1/9, 1/9, 1/36, 1/36, 1/36, 1/36\}$. Here, $\left(\mathbf{\Sigma}^{1/2}\right)_j$, is the j^{th} column or row of $\mathbf{\Sigma}^{1/2}$.

For the general case of an n-dimensional state vector, we can write

$$I_n = \sum_{j=0}^{M^n-1} \mathbf{g}(\mathbf{x}_j)\,\alpha_j, \tag{48}$$

where

$$\begin{aligned}
\mathbf{x}_0 &= \widehat{\mathbf{x}} \\
\mathbf{x}_j &= \widehat{\mathbf{x}} + \sqrt{3}\left(\mathbf{\Sigma}^{1/2}\right)_j, \quad j = 1,\ldots,n \\
\mathbf{x}_j &= \widehat{\mathbf{x}} - \sqrt{3}\left(\mathbf{\Sigma}^{1/2}\right)_{j-n}, \quad j = n+1,\ldots,2n. \\
\mathbf{x}_j &= \widehat{\mathbf{x}} \pm \text{ higher order terms}, \quad j = 2n+1,\ldots,M^n-1
\end{aligned} \tag{49}$$

The higher order terms are additional terms at the edges of an n-dimensional hypercube. The weights, after normalization, can be shown to be products of the form $q_{i_1} q_{i_2} \ldots q_{i_n}$.

In [4], the unscented filter is presented as

$$\begin{aligned}
\mathbf{x}_0 &= \widehat{\mathbf{x}} \\
\mathbf{x}_j &= \widehat{\mathbf{x}} + \sqrt{\frac{n}{1-w_0}}\left(\mathbf{\Sigma}^{1/2}\right)_j, \quad j = 1,\ldots,n \\
\mathbf{x}_n &= \widehat{\mathbf{x}} - \sqrt{\frac{n}{1-w_0}}\left(\mathbf{\Sigma}^{1/2}\right)_{j-n}, \quad j = n+1,\ldots,2n,
\end{aligned} \tag{50}$$

with

$$w_j = \frac{1 - w_0}{2n}, \qquad j = 1, \ldots, 2n. \tag{51}$$

w_0 provides control of how the positions of the Sigma points lie relative to the mean. In the unscented filter, the support points, $\mathbf{x}_j$, are called Sigma points, with associated weights w_j. In [6], several one-dimensional non-linear estimation examples are given in which Ito and Xiong show that the full Gauss-Hermite filter gives slightly better estimates than an unscented filter and both give far better estimates than the extended Kalman filter.

By comparing Eq. 50 with Eq. 49, it is easy to see that the unscented filter is a modified version of a Gauss-Hermite quadrature filter. It uses just the first $2n + 1$ terms of the Gauss-Hermite quadrature filter and will be almost identical in form with the Gauss-Hermite filter. The computational requirements for the Gauss-Hermite filter grow rapidly with n, and the number of operations required for each iteration will be of the order M^n. The number of operations for the unscented filter grows much more slowly, of the order $2n + 1$, and is therefore more attractive to use. If the PDF's are non-Gaussian or unknown, the unscented filter can be used by choosing an appropriate value for w_0. In addition, other, more general quadrature filters can be used [22]. These more general quadrature filters are referred to as deterministic particle filters.

The estimation procedure for the first two moments of $\mathbf{x}_n$ using the output of either the Gauss-Hermite quadrature filter or the unscented filter as input to a Kalman filter result in the nonlinear Kalman filter procedures shown in Fig. 2. In the figure, $c_j = \sqrt{3}$ and $N_s = M^n - 1$ for the Gauss-Hermite filter and $c_j = \sqrt{n/(n - w_0)}$ and $N_s = 2n$ for the unscented filter. Also, the higher order terms are only present in the Gauss-Hermite quadrature filter. Note that the weights for both filters are generally computed off-line. The Track File block is used to store the successive filter estimates. These filter structures are called the Gauss-Hermite Kalman filter (GHKF) and the unscented Kalman filter (UKF)

3.2. Numerical Integration Using a Monte Carlo Approximation

A Monte Carlo approximation of the expected value integrals uses a discrete approximation to the PDF $\mathcal{N}(\mathbf{x};\widehat{\mathbf{x}}, \mathbf{P}^{\mathbf{xx}})$. Draw N_s samples from $\mathcal{N}(\mathbf{x};\widehat{\mathbf{x}}, \mathbf{P}^{\mathbf{xx}})$, where $\left\{\mathbf{x}^{(i)}, i = 1, 2, \ldots, N_s\right\}$ are a set of support points (random samples or particles) with weights $\left\{w^{(i)} = 1/N_s,\ i = 1, 2, \ldots, N_s\right\}$. Now, $\mathcal{N}(\mathbf{x};\widehat{\mathbf{x}}, \mathbf{P}^{\mathbf{xx}})$ can be approximated by

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x};\widehat{\mathbf{x}}, \mathbf{P}^{\mathbf{xx}}) \simeq \sum_{i=1}^{N_s} w^{(i)} \delta\left(\mathbf{x} - \mathbf{x}^{(i)}\right). \tag{52}$$

Note that $w^{(i)}$ is not the probability of the point $\mathbf{x}^{(i)}$. The probability density near $\mathbf{x}^{(i)}$ is given by the density of points in the region around $\mathbf{x}^{(i)}$, which can be obtained from a normalized histogram of all $\mathbf{x}^{(i)}$. $w^{(i)}$ only has meaning when Eq. 52 is used inside an

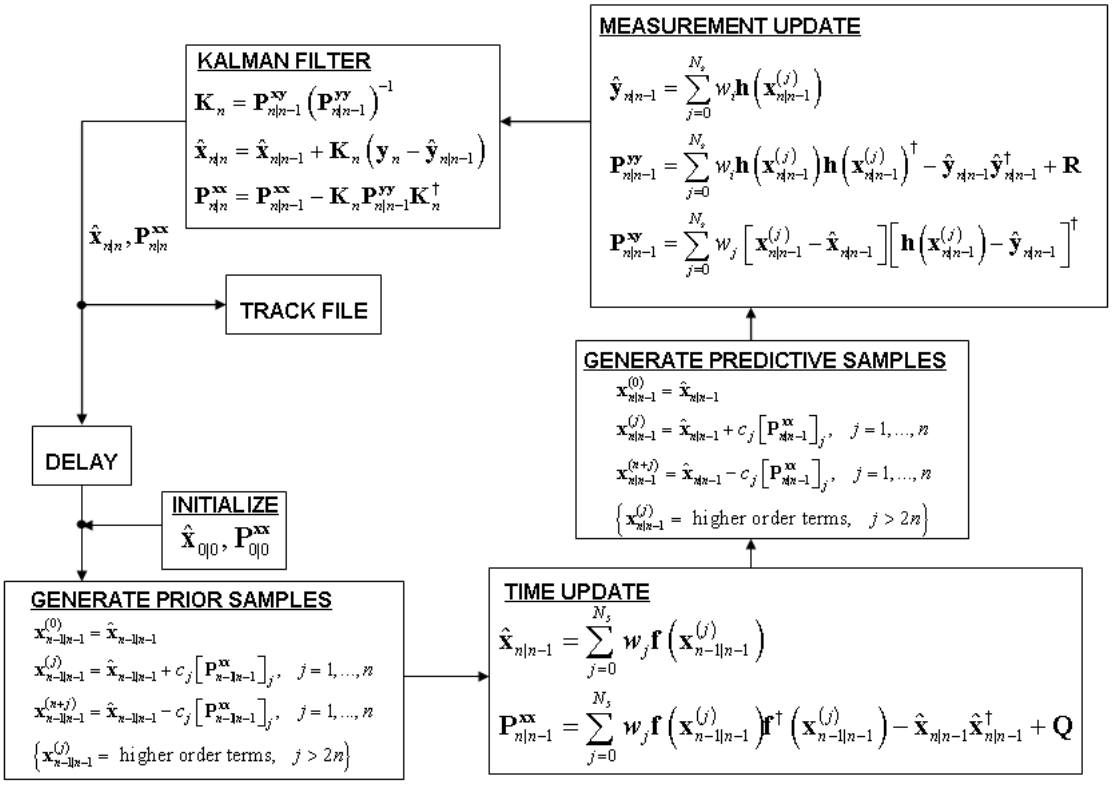

Figure 2: Nonlinear Gauss-Hermite/Unscented Kalman Filter Approximation

integral to turn the integral into its discrete approximation, as will be shown below. As $N_s \longrightarrow \infty$, this integral approximation approaches the true value of the integral.

Now, the expected value of any function of $\mathbf{g}(\mathbf{x})$ can be estimated from

$$\begin{aligned} \langle \mathbf{g}(\mathbf{x}) \rangle_{p(\mathbf{x})} &= \int \mathbf{g}(\mathbf{x})\, p(\mathbf{x})\, d\mathbf{x} \\ &\simeq \int \mathbf{g}(\mathbf{x}) \sum_{i=1}^{N_s} w^{(i)} \delta\left(\mathbf{x} - \mathbf{x}^{(i)}\right) d\mathbf{x} \\ &\simeq \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{g}\left(\mathbf{x}^{(i)}\right), \end{aligned} \tag{53}$$

which is obviously the sample mean. We used the above form to show the similarities between Monte Carlo integration and the quadrature integration of the last section. In quadrature integration, the support points $\mathbf{x}^{(i)}$ are at fixed intervals, while in Monte Carlo integration they are random.

Now, drawing samples of $\mathbf{x}_{n-1}$ from it's distribution $p(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1})$, we can write

$$\mathbf{x}_{n-1|n-1}^{(i)} \sim p(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}) = \mathcal{N}\left(\mathbf{x}_{n-1}; \widehat{\mathbf{x}}_{n-1|n-1}, \mathbf{P}_{n-1|n-1}^{\mathbf{xx}}\right), \tag{54}$$

for $i = 1, 2, \ldots, N_s$. Then, letting $\widehat{\mathbf{x}}_{n|n-1}$ be an approximation of $\langle \mathbf{x}_n | \mathbf{y}_{1:n-1} \rangle$, Eq's. 14 and 15 become

$$\widehat{\mathbf{x}}_{n|n-1} = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{f}\left(\mathbf{x}_{n-1|n-1}^{(i)}\right), \tag{55}$$

and

$$\begin{aligned} \mathbf{P}_{n|n-1}^{\mathbf{xx}} &= \mathbf{Q} + \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{f}\left(\mathbf{x}_{n-1|n-1}^{(i)}\right) \mathbf{f}^{\mathsf{T}}\left(\mathbf{x}_{n-1|n-1}^{(i)}\right) \\ &- \left[\frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{f}\left(\mathbf{x}_{n-1|n-1}^{(i)}\right)\right] \left[\frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{f}\left(\mathbf{x}_{n-1|n-1}^{(i)}\right)\right]. \end{aligned} \tag{56}$$

Now, we approximate the predictive PDF, $p(\mathbf{x}_n|\mathbf{y}_{1:n-1})$, as $\mathcal{N}\left(\mathbf{x}_n; \widehat{\mathbf{x}}_{n|n-1}, \mathbf{P}_{n|n-1}^{\mathbf{xx}}\right)$ and draw new samples

$$\mathbf{x}_{n|n-1}^{(i)} \sim \mathcal{N}\left(\mathbf{x}_n; \widehat{\mathbf{x}}_{n|n-1}, \mathbf{P}_{n|n-1}^{\mathbf{xx}}\right). \tag{57}$$

Using these samples from $p(\mathbf{x}_n|\mathbf{y}_{1:n-1})$, Eq's. 18, 19 and 20 reduce to

$$\widehat{\mathbf{y}}_{n|n-1} = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{h}\left(\mathbf{x}_{n|n-1}^{(i)}\right), \tag{58}$$

$$\begin{aligned} \mathbf{P}_{n|n-1}^{\mathbf{yy}} &= \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{h}\left(\mathbf{x}_{n|n-1}^{(i)}\right) \mathbf{h}\left(\mathbf{x}_{n|n-1}^{(i)}\right) \\ &- \left[\frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{h}\left(\mathbf{x}_{n|n-1}^{(i)}\right)\right] \left[\frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{h}\left(\mathbf{x}_{n|n-1}^{(i)}\right)\right]^{\mathsf{T}} + \mathbf{R}, \end{aligned} \tag{59}$$

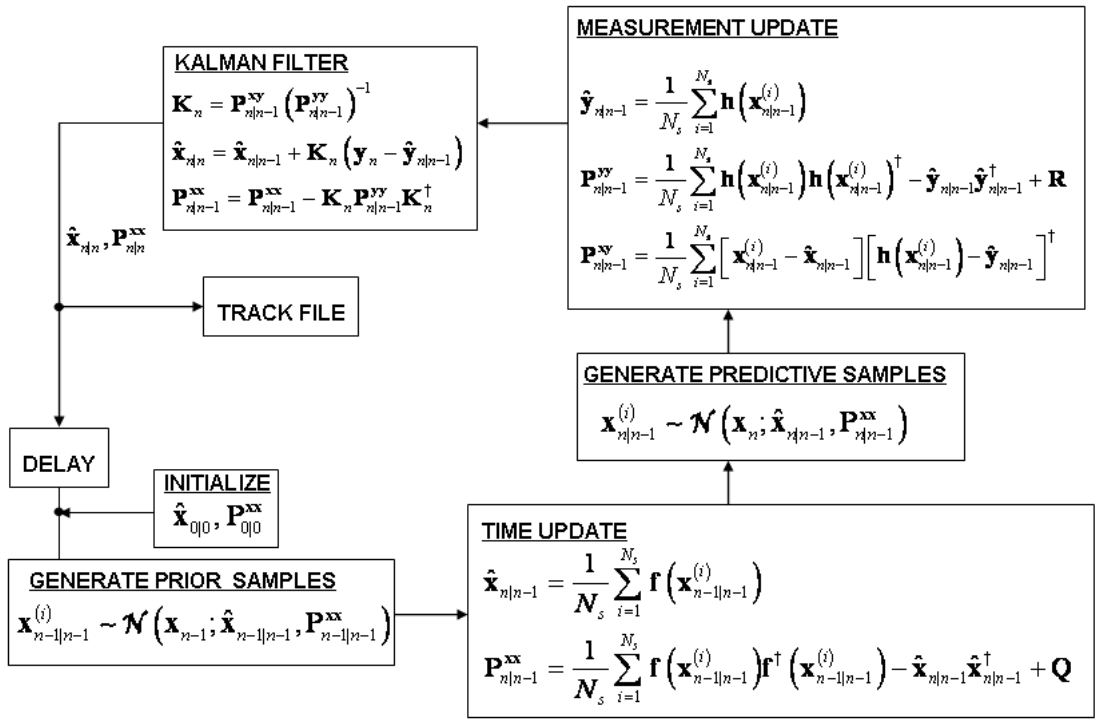

Figure 3: Nonlinear Monte Carlo Kalman Filter (MCKF) Approximation

and

$$\begin{aligned}\mathbf{P}^{\mathbf{xy}}_{n|n-1} &= \frac{1}{N_s}\sum_{i=1}^{N_s}\mathbf{x}^{(i)}_{n|n-1}\mathbf{h}\left(\mathbf{x}^{(i)}_{n|n-1}\right) \\ &\quad - \left[\frac{1}{N_s}\sum_{i=1}^{N_s}\mathbf{x}^{(i)}_{n|n-1}\right]\left[\frac{1}{N_s}\sum_{i=1}^{N_s}\mathbf{h}\left(\mathbf{x}^{(i)}_{n|n-1}\right)\right]^{\mathsf{T}}. \qquad (60)\end{aligned}$$

Using Eq's. 55, 56, and 58-60 in Eq's. 21-23 results in a procedure that we call the nonlinear Monte Carlo approximation to the Kalman filter (MCKF). The MCKF procedure is shown in Figure 3.

For Monte Carlo integration, the estimated variance is proportional to $1/\sqrt{N_s}$, so for 10,000 samples, the error in the variance is still 1%. Since the MCKF uses multiple integrations in a recursive manner, the errors can build up and the filter can diverge rapidly. However, the computational load, as well as the error in the variance, are independent of

the number of dimensions of the integrand. The computational load for Gauss-Hermite quadrature integration approximations goes as M^n, which grows rapidly with the dimension n. For large n, which is the case for multitarget tracking problems, Monte Carlo integration becomes more attractive than Gauss-Hermite quadrature. However, the UKF computational load grows only as $2n+1$, which makes the UKF the technique of choice as the number of dimensions increases.

4. Non-Linear Estimation using Particle Filters

In the previous section we assumed that if a general density function $p(\mathbf{x}_n|\mathbf{y}_{1:n})$ is Gaussian, we could generate Monte Carlo samples from it and use a discrete approximation to the density function given by Eq. 52. In many cases, $p(\mathbf{x}_n|\mathbf{y}_{1:n})$ may be multivariate and non-standard (i.e. not represented by any analytical PDF), or multimodal. For these cases, it may be difficult to generate samples from $p(\mathbf{x}_n|\mathbf{y}_{1:n})$. To overcome this difficulty we utilize the principle of *Importance Sampling*. Suppose $p(\mathbf{x}_n|\mathbf{y}_{1:n})$ is a PDF from which it is difficult to draw samples. Also, suppose that $q(\mathbf{x}_n|\mathbf{y}_{1:n})$ is another PDF from which samples can be easily drawn (referred to as the *Importance Density*) [9]. For example, $p(\mathbf{x}_n|\mathbf{y}_{1:n})$ could be a PDF for which we have no analytical expression and $q(\mathbf{x}_n|\mathbf{y}_{1:n})$ could be an analytical Gaussian PDF. Now we can write $p(\mathbf{x}_n|\mathbf{y}_{1:n}) \propto q(\mathbf{x}_n|\mathbf{y}_{1:n})$, where the symbol $\propto$ means that $p(\mathbf{x}_n|\mathbf{y}_{1:n})$ is proportional to $q(\mathbf{x}_n|\mathbf{y}_{1:n})$ at every $\mathbf{x}_n$. Since $p(\mathbf{x}_n|\mathbf{y}_{1:n})$ is a normalized PDF, then $q(\mathbf{x}_n|\mathbf{y}_{1:n})$ must be a scaled unnormalized version of $p(\mathbf{x}_n|\mathbf{y}_{1:n})$ with a different scaling factor at each $\mathbf{x}_n$. Thus, we can write the scaling factor or weight as

$$w(\mathbf{x}_n) = \frac{p(\mathbf{x}_n|\mathbf{y}_{1:n})}{q(\mathbf{x}_n|\mathbf{y}_{1:n})}. \tag{61}$$

Now, Eq. 5 can be written as

$$\langle \mathbf{g}(\mathbf{x}_n) \rangle_{p(\mathbf{x}_n|\mathbf{y}_{1:n})} = \frac{\int \mathbf{g}(\mathbf{x}_n)\, w(\mathbf{x}_n)\, q(\mathbf{x}_n|\mathbf{y}_{1:n})\, d\mathbf{x}_n}{\int w(\mathbf{x}_n)\, q(\mathbf{x}_n|\mathbf{y}_{1:n})\, d\mathbf{x}_n}, \tag{62}$$

If one generates N_s particles (samples) $\left\{\mathbf{x}_n^{(i)}, i = 1, \ldots, N_s\right\}$ from $q(\mathbf{x}_n|\mathbf{y}_{1:n})$, then a possible Monte Carlo estimate of $\langle \mathbf{g}(\mathbf{x}_n) \rangle_{p(\mathbf{x}_n|\mathbf{y}_{1:n})}$ is

$$\widehat{\mathbf{g}}(\mathbf{x}_n) = \frac{\frac{1}{N_s}\sum_{i=1}^{N_s} \mathbf{g}\left(\mathbf{x}_n^{(i)}\right) \widetilde{w}\left(\mathbf{x}_n^{(i)}\right)}{\frac{1}{N_s}\sum_{i=1}^{N_s} w\left(\mathbf{x}_n^{(i)}\right)} = \sum_{i=1}^{N_s} \mathbf{g}\left(\mathbf{x}_n^{(i)}\right) \widetilde{w}\left(\mathbf{x}_n^{(i)}\right), \tag{63}$$

where the normalized importance weights $\widetilde{w}\left(\mathbf{x}_n^{(i)}\right)$ are given by

$$\widetilde{w}\left(\mathbf{x}_n^{(i)}\right) = \frac{\widetilde{w}\left(\mathbf{x}_n^{(i)}\right)}{\frac{1}{N_s}\sum_{i=1}^{N_s} w\left(\mathbf{x}_n^{(i)}\right)}. \tag{64}$$

However, it would be useful if the importance weights could be generated recursively. So, using Eq. 4, we can write

$$w\left(\mathbf{x}_n\right) = \frac{p\left(\mathbf{x}_n|\mathbf{y}_{1:n}\right)}{q\left(\mathbf{x}_n|\mathbf{y}_{1:n}\right)} = \frac{cp\left(\mathbf{y}_n|\mathbf{x}_n\right)p\left(\mathbf{x}_n|\mathbf{y}_{1:n-1}\right)}{q\left(\mathbf{x}_n|\mathbf{y}_{1:n}\right)}. \tag{65}$$

Using the expansion of $p\left(\mathbf{x}_n|\mathbf{y}_{1:n-1}\right)$ found in Eq. 3 and expanding the importance density in a similar fashion, Eq. 65 can be written as

$$w\left(\mathbf{x}_n\right) = \frac{cp\left(\mathbf{y}_n|\mathbf{x}_n\right)\int p\left(\mathbf{x}_n|\mathbf{x}_{n-1},\mathbf{y}_{1:n-1}\right)p\left(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}\right)d\mathbf{x}_{n-1}}{\int q\left(\mathbf{x}_n|\mathbf{x}_{n-1},\mathbf{y}_{1:n}\right)q\left(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}\right)d\mathbf{x}_{n-1}}. \tag{66}$$

When Monte Carlo samples are drawn from the importance density, this leads to a recursive formulation for the importance weights, as will be shown in the next section.

4.1. Particle Filters that Require Resampling: The Sequential Importance Sampling Particle Filter

Now, suppose we have available a set of particles (random samples from the distribution) and weights, $\left\{\mathbf{x}_{n-1|n-1}^{(i)}, w_{n-1}^{(i)}\right\}_{i=1}^{N_s}$, that constitute a random measure which characterizes the posterior PDF for times up to t_{n-1}. Then this previous posterior PDF, $p\left(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}\right)$, can be approximated by

$$p\left(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}\right) \approx \sum_{i=1}^{N_s} w_{n-1}^{(i)}\delta\left(\mathbf{x}_{n-1} - \mathbf{x}_{n-1|n-1}^{(i)}\right). \tag{67}$$

So, if the particles $\mathbf{x}_{n-1|n-1}^{(i)}$ were drawn from the importance density $q\left(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}\right)$, the weights in Eq. 67 are defined by Eq. 61 to be

$$w_{n-1}^{(i)} = \frac{p\left(\mathbf{x}_{n-1|n-1}^{(i)}\right)}{q\left(\mathbf{x}_{n-1|n-1}^{(i)}\right)}. \tag{68}$$

For the sequential case, called *sequential importance sampling* (SIS) [10], at each iteration one could have the random measure $\left\{\mathbf{x}_{n-1|n-1}^{(i)}, w_{n-1}^{(i)}\right\}_{i=1}^{N_s}$ constituting an approximation to $p\left(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}\right)$ (i.e., not drawn from $q\left(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}\right)$) and want to approximate $p\left(\mathbf{x}_n|\mathbf{y}_{1:n}\right)$ with a new set of samples and weights. By substituting Eq. 67 in Eq. 66, and using a similar formulation for $q\left(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}\right)$, the weight update equation for *each* particle becomes

$$\begin{aligned} w_n^{(i)} &\propto \frac{p\left(\mathbf{y}_n|\mathbf{x}_{n|n-1}^{(i)}\right)p\left(\mathbf{x}_{n|n-1}^{(i)}|\mathbf{x}_{n-1|n-1}^{(i)},\mathbf{y}_{1:n-1}\right)p\left(\mathbf{x}_{n-1|n-1}^{(i)}\right)}{q\left(\mathbf{x}_{n|n-1}^{(i)}|\mathbf{x}_{n-1|n-1}^{(i)},\mathbf{y}_{1:n-1}\right)q\left(\mathbf{x}_{n-1|n-1}^{(i)}\right)} \\ &= w_{n-1}^{(i)}\frac{p\left(\mathbf{y}_n|\mathbf{x}_{n|n-1}^{(i)}\right)p\left(\mathbf{x}_{n|n-1}^{(i)}|\mathbf{x}_{n-1|n-1}^{(i)}\right)}{q\left(\mathbf{x}_{n|n-1}^{(i)}|\mathbf{x}_{n-1|n-1}^{(i)}\right)}, \end{aligned} \tag{69}$$

where we obtain $\mathbf{x}_{n|n-1}^{(i)}$ from Eq. 1, rewritten here as

$$\mathbf{x}_{n|n-1}^{(i)} = \mathbf{f}\left(\mathbf{x}_{n-1|n-1}^{(i)}, \mathbf{w}_{n-1}^{(i)}\right). \tag{70}$$

This form of the time update equation requires an additional step, that of generating samples of the dynamic noise, $\mathbf{w}_{n-1}^{(i)} \sim p(\mathbf{w})$, which must be addressed in the implementation of these filters.

The posterior filtered PDF $p(\mathbf{x}_n|\mathbf{y}_{1:n})$ can then be approximated by

$$p(\mathbf{x}_n|\mathbf{y}_{1:n}) \approx \sum_{i=1}^{N_s} w_n^{(i)} \delta\left(\mathbf{x}_n - \mathbf{x}_{n|n}^{(i)}\right), \tag{71}$$

where the updated weights are generated recursively using Eq. 69.

Problems occur with SIS based particle filters. Repeated applications of Eq. 70 causes particle dispersion, because the variance of $\mathbf{x}_n$ increases without bound as $n \to \infty$. Thus, for those $\mathbf{x}_{n|n-1}^{(i)}$ that disperse away from the expected value $\widehat{\mathbf{x}}_n$, their probability weights $w_n^{(i)}$ go to zero. This problem has been labeled the degeneracy problem of the particle filter [9]. To measure the degeneracy of the particle filter, the effective sample size, N_{eff}, has been introduced, as noted in [11]. N_{eff} can be estimated from $\hat{N}_{eff} = 1/\sum_{i=1}^{N_s}\left(w_n^{(i)}\right)^2$. Clearly, the degeneracy problem is an undesirable effect in particle filters. The brute force approach to reducing its effect is to use a very large N_s. This is often impractical, so for SIS algorithms an additional step called resampling must be added to the SIS procedure (*sequential importance sampling with resampling* (SISR)). Generally, a resampling step is added at each time interval (*systematic resampling*) [10] that replaces low probability particles with high probability particles, keeping the number of particles constant. The resampling step need only be done when $\hat{N}_{eff} \leq N_s$. This *adaptive resampling* allows the particle filter to keep it's memory during the interval when no resampling occurs. In this paper, we will discuss only systematic resampling.

One method for resampling, the inverse transformation method, is discussed in [23]. In [23], Ross presents a proof (Inverse Transform Method, pages 477-478) that if u is a uniformly distributed random variable, then for any continuous distribution function F, the random variable defined by $x = F^{-1}(u)$ has distribution F. We can use this Inverse Transform Method for resampling. We first form the discrete approximation of the cumulative distribution function

$$\begin{aligned} F(\mathbf{x}) &= P(\mathbf{z} \leq \mathbf{x}) = \int_{-\infty}^{\mathbf{x}} p(\mathbf{z})\, d\mathbf{z} \\ &= \int_{-\infty}^{\mathbf{x}} \sum_{i=1}^{N_s} w^{(i)} \delta\left(\mathbf{z} - \mathbf{z}^{(i)}\right) d\mathbf{z} \\ &= \sum_{i=1}^{j} w^{(i)}, \end{aligned} \tag{72}$$

where j is the index for the $\mathbf{x}^{(i)}$ nearest but below $\mathbf{x}$. We can write this discrete approximation to the cumulative distribution function as $F\left(\mathbf{x}^{(j)}\right) = \sum_{i=1}^{j} w^{(i)}$. Now, we select $u^{(i)} \sim U(0,1),\ i = 1, \ldots, N_s$ and for each value of $u^{(i)}$, interpolate a value of $\mathbf{x}^{(i)}$ from $\mathbf{x}^{(i)} = F^{-1}\left(u^{(i)}\right)$. Since the $u^{(i)}$ are uniformly distributed, the probability that $\mathbf{x}^{(i)} = \mathbf{x}$ is $1/N_s$, i.e., all $\mathbf{x}^{(i)}$ in the sample set are equally probable. Thus, for the resampled particle set, $\tilde{w}^{(i)} = 1/N_s, \forall i$. The procedure for SIS with resampling is straightforward and is presented in Fig. 4

Several other techniques for generating samples from an unknown PDF, besides importance sampling, have been presented in the literature. If the PDF is stationary, Markov Chain Monte Carlo (MCMC) methods have been proposed, with the most famous being the Metropolis-Hastings (MH) algorithm, the Gibbs sampler (which is a special case of MH), and the coupling from the past (CFTP) perfect sampler [24], [25]. These techniques work very well for off-line generation of PDF samples but they are not suitable in recursive estimation applications since they frequently require in excess of 100,000 iterations. These sampling techniques will not be discussed further.

Before the SIS algorithm can be implemented, one needs to quantify the specific probabilities for $q\left(\mathbf{x}_{n|n-1}^{(i)}|\mathbf{x}_{n-1|n-1}^{(i)}\right)$, $p\left(\mathbf{x}_{n|n-1}^{(i)}|\mathbf{x}_{n-1|n-1}^{(i)}\right)$ and the likelihood $p\left(\mathbf{y}_n|\mathbf{x}_{n|n-1}^{(i)}\right)$. If the noise in the respective process or observation models cannot be modeled as additive and Gaussian, quantification of these density functions can sometimes be difficult.

4.1.1. The Bootstrap Approximation and the Bootstrap Particle Filter

In the bootstrap particle filter [10], we make the approximation that the importance density is equal to the prior density, i.e., $q\left(\mathbf{x}_{n|n-1}^{(i)}|\mathbf{x}_{n-1|n-1}^{(i)}\right) = p\left(\mathbf{x}_{n|n-1}^{(i)}|\mathbf{x}_{n-1|n-1}^{(i)}\right)$. This eliminates two of the densities needed to implement the SIS algorithm, since they now cancel each other from Eq. 69. The weight update equation then becomes

$$w_n^{(i)} = w_{n-1}^{(i)} p\left(\mathbf{y}_n|\mathbf{x}_{n|n-1}^{(i)}\right). \tag{73}$$

The procedure for the *bootstrap particle filter* is identical to that of the SIS particle filter given above, except that Eq. 73 is used instead of Eq. 69 in the importance weight update step. Notice that the dimensionality of both the observation vector and the state vector only appear in the likelihood function $p\left(\mathbf{y}_n|\mathbf{x}_{n|n-1}^{(i)}\right)$. Regardless of the number of dimensions, once the likelihood function is specified for a given problem the computational load becomes proportional to the number of particles, which can be much less than the number of support points required for the GHKF, UKF, or MCKF. Since the bootstrap particle filter can also be applied to problems in which the noise is additive and Gaussian, this filter can be applied successfully to almost any tracking problem. The only flaw is that it is highly dependent on the initialization estimates and can quickly diverge if the initialization mean of the state vector is far from the true state vector, since the observations are only used in the likelihood function.

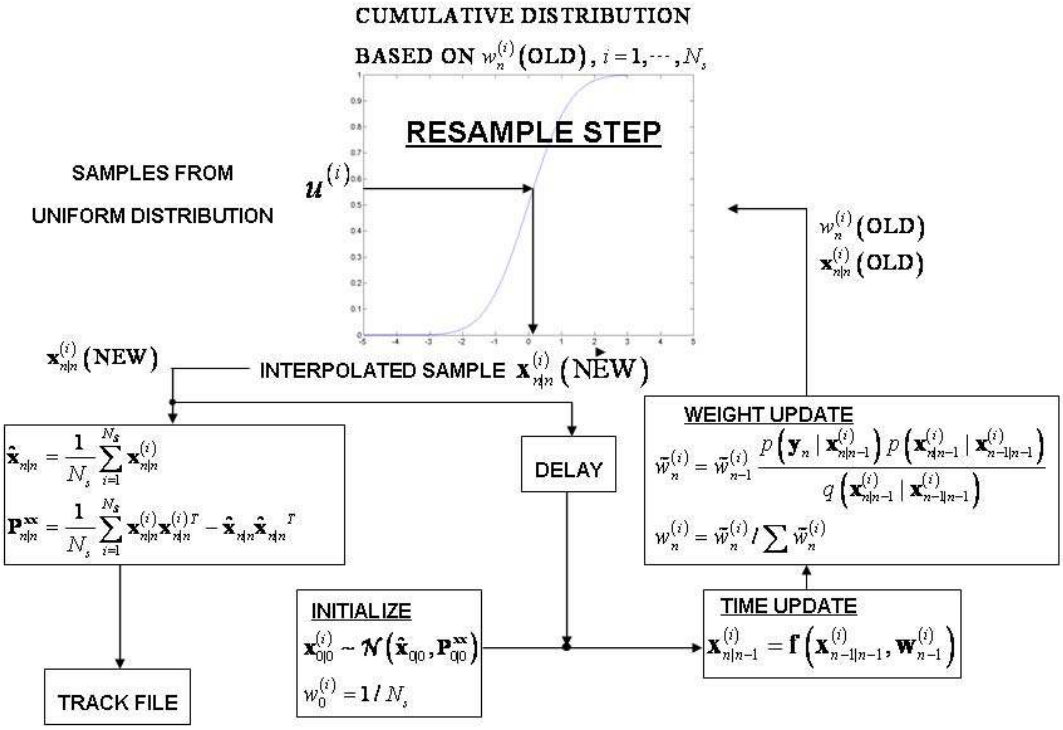

Figure 4: The General Sequential Importance Sampling Particle Filter

4.2. Particle Filters That Do Not Require Resampling

There are several particle filter approximation techniques that do not require resampling and most of them stem from Eq. 65. If samples are drawn from the importance density $\left[\mathbf{x}_{n|n}^{(i)} \sim q\left(\mathbf{x}_n|\mathbf{y}_{1:n}\right)\right]$, and we can calculate the importance weights in a non-iterative fashion from

$$w_n^{(i)} \propto \frac{p\left(\mathbf{y}_n|\mathbf{x}_{n|n}^{(i)}\right) p\left(\mathbf{x}_{n|n}^{(i)};\mathbf{x}_n|\mathbf{y}_{1:n-1}\right)}{q\left(\mathbf{x}_{n|n}^{(i)};\mathbf{x}_n|\mathbf{y}_{1:n-1}\right)}. \tag{74}$$

This is followed by a normalization step given in Eq. 64.

This more general particle filter is illustrated in the block diagram of Fig. 5, which uses Eq. 74 to calculate the weights. In the paragraphs that follow, we will show how to fill in the boxes and make approximations for the predictive density $p\left(\mathbf{x}_n|\mathbf{y}_{1:n-1}\right)$ and the importance density $q\left(\mathbf{x}_n|\mathbf{y}_{1:n}\right)$. Note that terms in Eq. 74 are not the PDFs, but instead are the PDFs evaluated at a particle position and are therefore probabilities between zero and one.

4.2.1. The Gaussian Particle Filter

The so-called Gaussian particle filter [12] approximates the previous posterior density $p\left(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}\right)$ by the Gaussian distribution $\mathcal{N}\left(\mathbf{x}_{n-1};\widehat{\mathbf{x}}_{n-1|n-1},\mathbf{P}_{n-1|n-1}^{\mathbf{xx}}\right)$. Samples are drawn

$$\mathbf{x}_{n-1|n-1}^{(i)} \sim \mathcal{N}\left(\mathbf{x}_{n-1};\widehat{\mathbf{x}}_{n-1|n-1},\mathbf{P}_{n-1|n-1}^{\mathbf{xx}}\right), \tag{75}$$

and $\mathbf{x}_{n|n-1}^{(i)}$ is obtained from $\mathbf{x}_{n-1|n-1}^{(i)}$ using Eq. 70. Then, the prior density $p\left(\mathbf{x}_n;\mathbf{x}_n|\mathbf{y}_{1:n-1}\right)$ is approximated by the Gaussian distribution $\mathcal{N}\left(\mathbf{x}_n;\widehat{\mathbf{x}}_{n|n-1},\mathbf{P}_{n|n-1}^{\mathbf{xx}}\right)$, where

$$\widehat{\mathbf{x}}_{n|n-1} = \sum_{i=1}^{N_s} w_{n-1}^{(i)} \mathbf{x}_{n|n-1}^{(i)}, \tag{76}$$

$$\mathbf{P}_{n|n-1}^{\mathbf{xx}} = \sum_{i=1}^{N_s} w_{n-1}^{(i)} \left(\mathbf{x}_{n|n-1}^{(i)} - \widehat{\mathbf{x}}_{n|n-1}\right)\left(\mathbf{x}_{n|n-1}^{(i)} - \widehat{\mathbf{x}}_{n|n-1}\right)^{\mathsf{T}}, \tag{77}$$

After samples are drawn from the importance density, the weights are calculated from

$$w_n^{(i)} \propto \frac{p\left(\mathbf{y}_n|\mathbf{x}_{n|n}^{(i)}\right)\mathcal{N}\left(\mathbf{x}_{n|n}^{(i)};\widehat{\mathbf{x}}_{n|n-1},\mathbf{P}_{n|n-1}^{\mathbf{xx}}\right)}{q\left(\mathbf{x}_{n|n}^{(i)};\mathbf{x}_n|\mathbf{y}_{1:n-1}\right)}$$

Now, the first and second moments of $\mathbf{x}_{n|n}$ can then be calculated from

$$\widehat{\mathbf{x}}_{n|n} = \sum_{i=1}^{N_s} w_n^{(i)} \mathbf{x}_{n|n}^{(i)}, \tag{78}$$

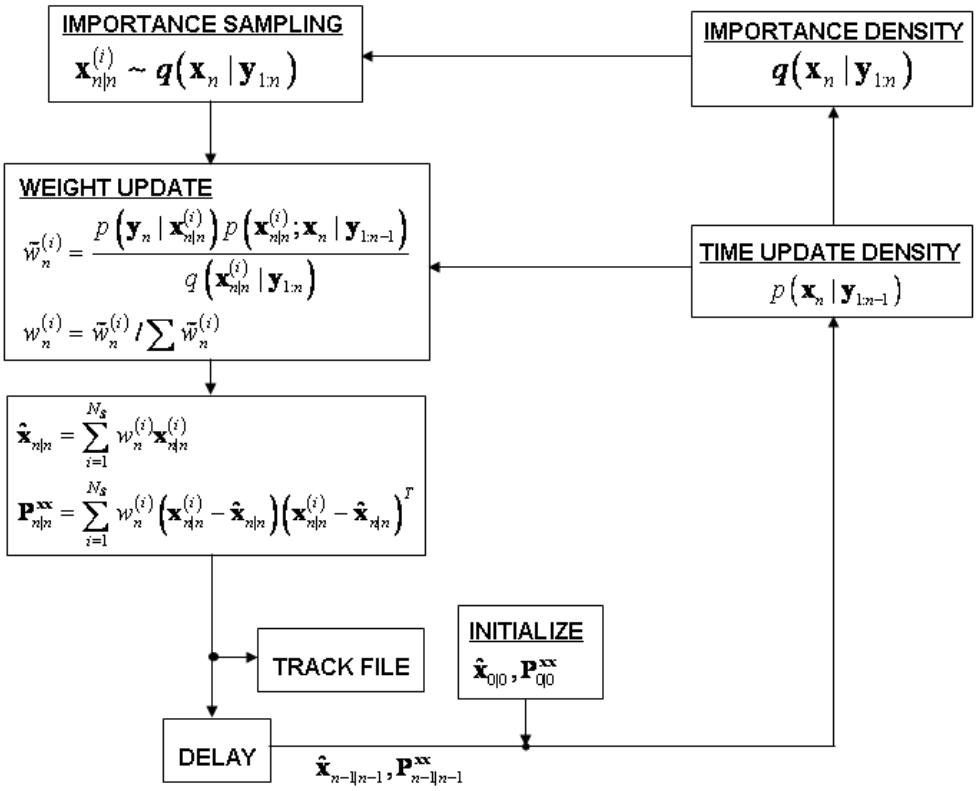

Figure 5: A General Particle Filter Without Resampling

and

$$\mathbf{P}_{n|n}^{\mathbf{xx}} = \sum_{i=1}^{N_s} w_n^{(i)} \left(\mathbf{x}_{n|n}^{(i)} - \widehat{\mathbf{x}}_{n|n} \right) \left(\mathbf{x}_{n|n}^{(i)} - \widehat{\mathbf{x}}_{n|n} \right)^{\mathsf{T}}. \tag{79}$$

The Gaussian particle filter (GPF) process is shown in Fig. 6.

4.2.2. The Monte Carlo, Gauss-Hermite, and Unscented Particle Filters

In Fig. 6, the importance density is not specified. Thus the Gaussian particle filter is not complete and still requires the specification of an importance density. In [12], it is suggested that a Gaussian distribution be used as the importance density, i.e., let $q\left(\mathbf{x}_n|\mathbf{y}_{1:n-1}\right) = \mathcal{N}\left(\mathbf{x}_n; \widehat{\mathbf{x}}_{n|n}, \mathbf{P}_{n|n}^{\mathbf{xx}}\right)$, where $\widehat{\mathbf{x}}_{n|n}$ and $\mathbf{P}_{n|n}^{\mathbf{xx}}$ are obtained from the prior density, as in the SIS algorithm, or from an EKF or UKF measurement update of the prior. In this section, we will show how this can be accomplished with the previously introduced MCKF, GHKF, or UKF.

Two composite particle filters are presented in Figures 7 and 8. In both figures, we have taken the Gaussian particle filter of Fig. 6 and used it to replace the prior $p\left(\mathbf{x}_n|\mathbf{y}_{1:n-1}\right)$ with the Gaussian density $\mathcal{N}\left(\mathbf{x}_n; \widehat{\mathbf{x}}_{n|n-1}, \mathbf{P}_{n|n-1}^{\mathbf{xx}}\right)$, where $\widehat{\mathbf{x}}_{n|n-1}$, and $\mathbf{P}_{n|n-1}^{\mathbf{xx}}$ are obtained from a time update step. In Fig. 7 we used the MCKF let the importance density $q\left(\mathbf{x}_n|\mathbf{y}_{1:n}\right)$ be the Gaussian density $\mathcal{N}\left(\mathbf{x}_n; \boldsymbol{\mu}_{n|n}^{\mathbf{x}}, \boldsymbol{\Sigma}_{n|n}^{\mathbf{xx}}\right)$, where $\boldsymbol{\mu}_{n|n}^{\mathbf{x}}$, and $\boldsymbol{\Sigma}_{n|n}^{\mathbf{xx}}$ are the outputs of the MCKF. Along the bottom of Fig. 7, you can identify the Gaussian particle filter structure and along the top and upper right the MCKF structure can be seen. The mean and covariance output of the MCKF are then used in a Gaussian importance density from which the particles are sampled. Then, the particles are used in a particle filter down the left hand side. The estimated mean and covariance outputs for the current sample time are then stored in a track file.

In Fig. 8 we use the GHKF/UKF to replace the importance density $q\left(\mathbf{x}_n|\mathbf{y}_{1:n}\right)$ by the Gaussian density $\mathcal{N}\left(\mathbf{x}_n; \boldsymbol{\mu}_{n|n}^{\mathbf{x}}, \boldsymbol{\Sigma}_{n|n}^{\mathbf{xx}}\right)$, where $\boldsymbol{\mu}_{n|n}^{\mathbf{x}}$, and $\boldsymbol{\Sigma}_{n|n}^{\mathbf{xx}}$ are the outputs of the GHKF/UKF. In [13], an unscented particle filter is presented that is similar, but does not include the Gaussian approximation for the prior.

When applying these filters to real-world problems, both the Gauss-Hermite and Unscented particle filters work very well and can usually be implemented in such a way that they run in real-time. However, the Monte Carlo particle filter is very difficult to implement due to the large number of particles required and numerical instabilities caused by outlier samples. In the example shown in Section 6, below, we do not present results for the Monte Carlo particle filter due to difficulties with numerical instabilities.

4.2.3. Rao-Blackwellization to Reduce Computational Load

When either the process model or the observation model is linear, the computational load can be reduced for any of the techniques presented above using Rao-Blackwellization

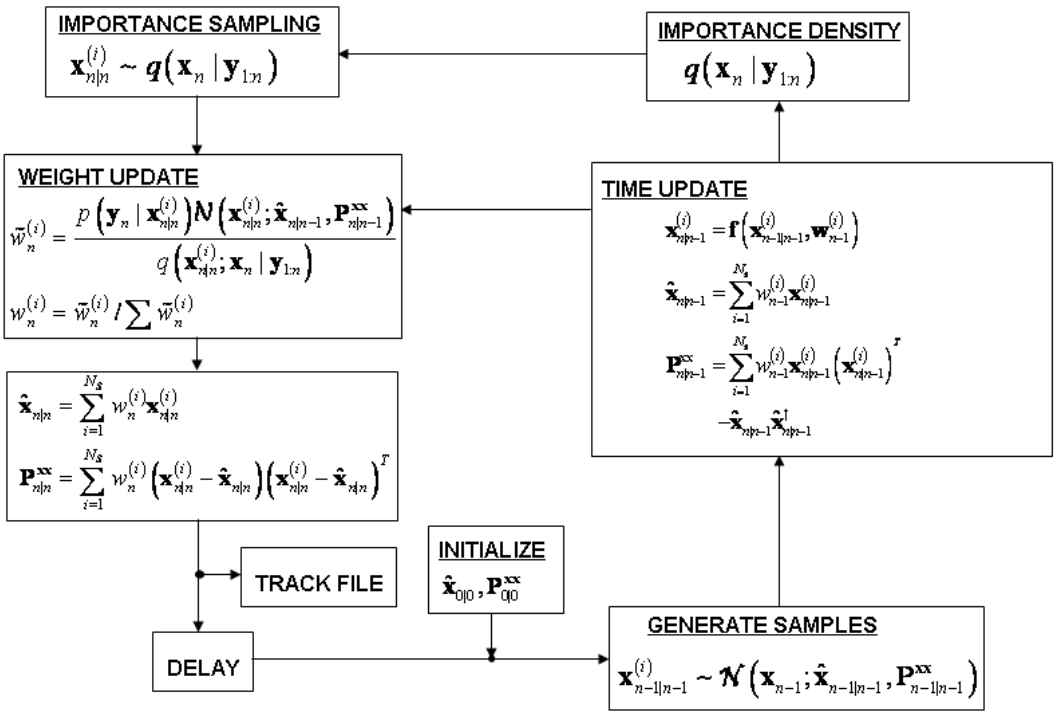

Figure 6: Gaussian Particle Filter

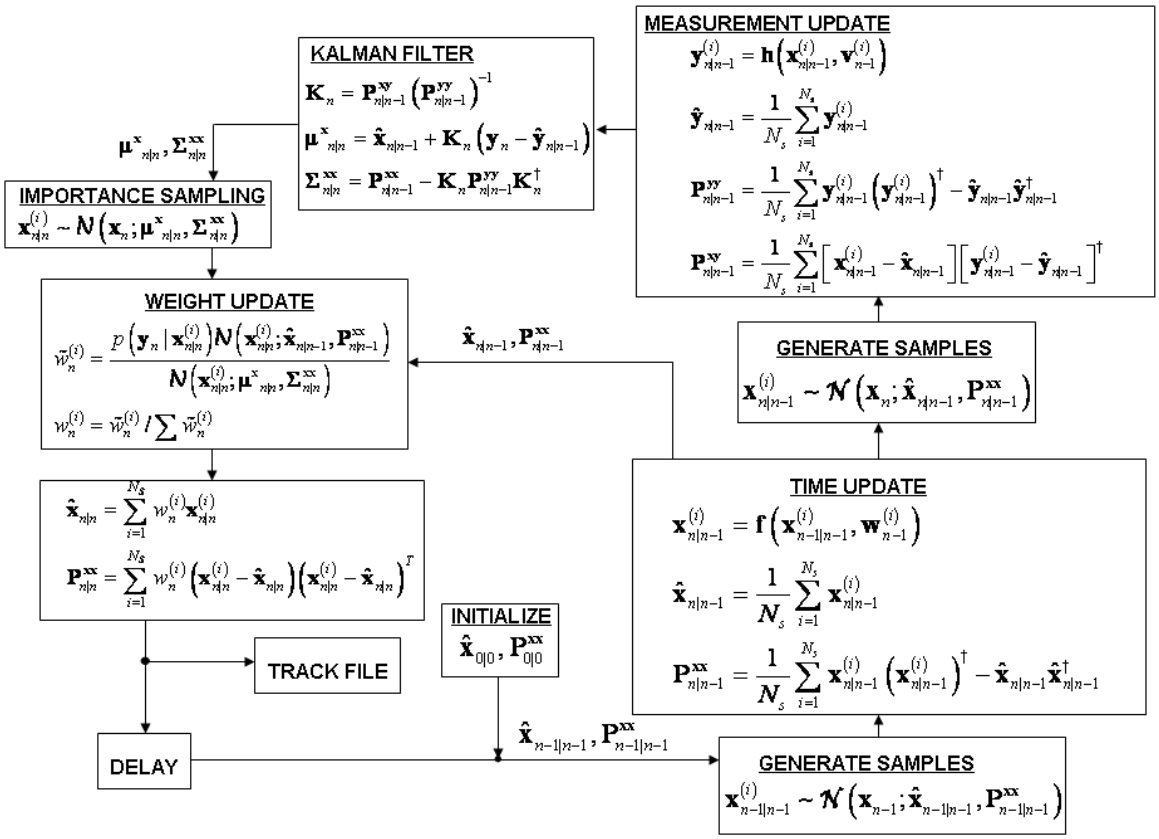

Figure 7: Monte Carlo Particle Filter.

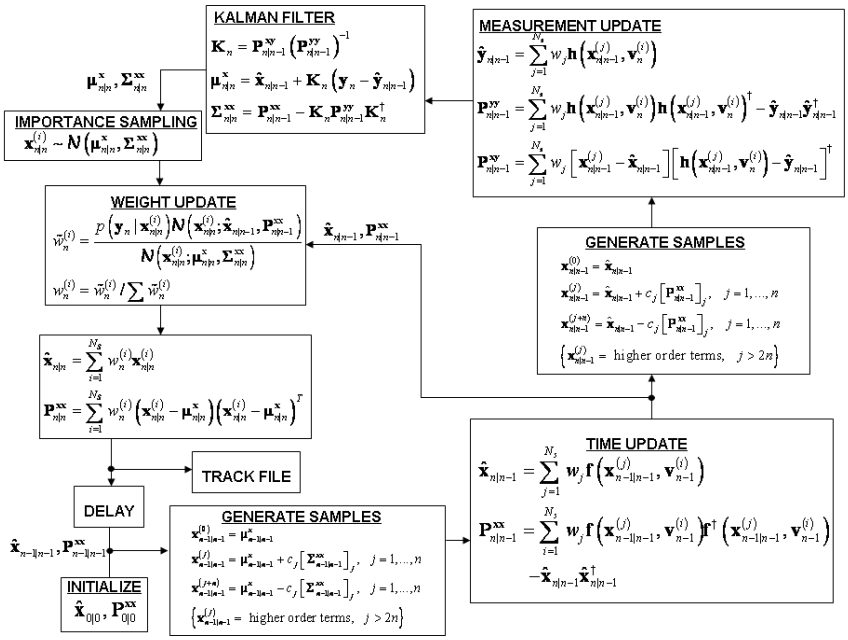

Figure 8: Gauss-Hermite or Unscented Particle Filter.

[14], [26], [27]. For any equation of the linear form

$$\mathbf{z} = \mathbf{C}\mathbf{d} + \mathbf{w}, \tag{80}$$

where $\mathbf{w}$ is $\mathcal{N}(\mathbf{0}, \mathbf{P}^{\mathbf{ww}})$ noise, $\mathbf{z}$ and $\mathbf{d}$ are stochastic n-vectors and $\mathbf{C}$ is an $n \times n$ deterministic matrix, then

$$\widehat{\mathbf{z}} = \mathbf{C}\widehat{\mathbf{d}}, \tag{81}$$

and

$$\mathbf{P}^{\mathbf{zz}} = \mathbf{C}\mathbf{P}^{\mathbf{dd}}\mathbf{C}^{\intercal} + \mathbf{P}^{\mathbf{ww}}. \tag{82}$$

Thus, if the process equation is linear, for the Rao-Blackwellization procedure an equation of the form given by Eq's. 81 and 82 can be used in the time update step in place of the combination sampling and time update steps used for the nonlinear case. Similarly, if the observation equation is linear, an equation of the form given by Eq. 81 and 82 can be used to replace the combination sampling and measurement update step.

For example, in Fig. 2, if $\mathbf{x}_n = \mathbf{C}\mathbf{x}_{n-1} + \mathbf{w}_{n-1}$, then the Generate Prior Samples and Time Update blocks can be replace by the equations

$$\widehat{\mathbf{x}}_{n|n-1} = \mathbf{C}\widehat{\mathbf{x}}_{n-1|n-1} \tag{83}$$

and

$$\mathbf{P}^{\mathbf{xx}}_{\mathbf{n|n-1}} = \mathbf{C}\mathbf{P}^{\mathbf{xx}}_{\mathbf{n-1|n-1}}\mathbf{C}^{\intercal} + \mathbf{Q}, \tag{84}$$

thus reducing the computational load considerably.

A similar reduction can be obtained if the observation equation is linear.

5. A Tracking Example Where all Noise is Additive Gaussian

Consider the problem of successively estimating (tracking) a constant velocity target using only noisy broadband passive array sensor data from an array mounted on a maneuvering observer ship. In a fixed two dimensional geographic coordinate system, the geo-tracks of the observer and a target ship are shown in Fig. 9.

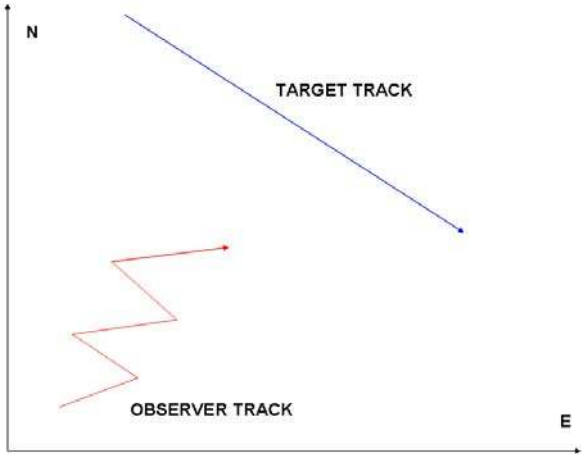

Figure 9: Tracks of Target and Observer Ships in a Fixed Geographic Coordinate System

Now define an array coordinate system whose origin is at the phase center of an array mounted on the moving observer ship, whose Y'-axis is positive in the forward direction (direction of motion of the moving array) and whose X'-axis is perpendicular in a clockwise direction (athwartship), as shown in Fig. 10. The instantaneous observer heading, $\vartheta(t)$, is defined as the angle between the geographic Y-axis (North) and the array coordinate Y'-axis (forward), with $0 \leq \vartheta(t) \leq 2\pi$. The relative bearing, $\theta(t)$, is defined as the angle *clockwise* from the Y'-axis to the observer-to-target line-of-sight line, where $0 \leq \theta(t) \leq 2\pi$, shown in Fig. 11.

The source state vector in the array coordinate system, $\mathbf{x}_a(t)$, is defined by

$$\mathbf{x}_a(t) = \mathbf{A}(\vartheta(t))\left[\mathbf{x}_s(t) - \mathbf{x}_o(t)\right], \tag{85}$$

where

$$\mathbf{A}(\vartheta(t)) \triangleq \begin{bmatrix} \mathbf{\Lambda}(\vartheta(t)) & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{\Lambda}(\vartheta(t)) \end{bmatrix}, \tag{86}$$

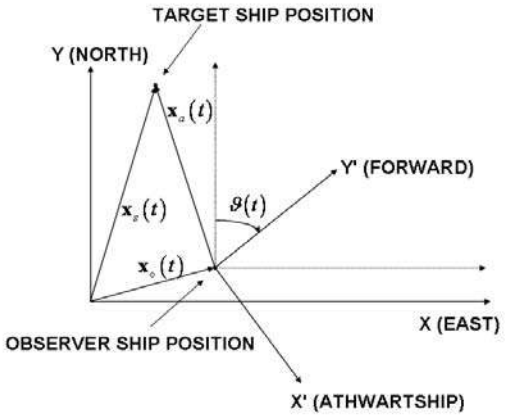

Figure 10: Relationship Between Fixed Geograhic Coordinate System and Moving Array Centered Coordinate System.

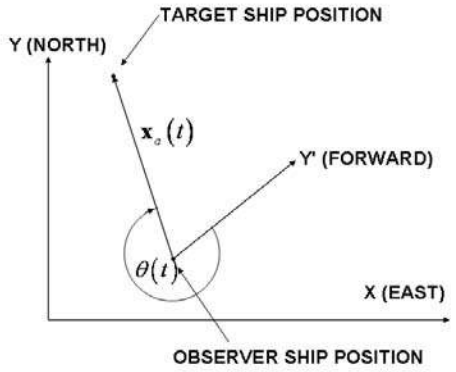

Figure 11: Definition of Relative Bearing

and

$$\mathbf{\Lambda}\left(\vartheta\left(t\right)\right) \triangleq \left[\begin{array}{cc} \cos\vartheta\left(t\right) & -\sin\vartheta\left(t\right) \\ \sin\vartheta\left(t\right) & \cos\vartheta\left(t\right) \end{array}\right], \tag{87}$$

where $\mathbf{0}_2$ is a 2×2 matrix of zeros. $\mathbf{x}_s\left(t\right)$ and $\mathbf{x}_o\left(t\right)$ are the constant-velocity target and maneuvering observer state vectors, respectively, in the fixed geographic coordinate system, defined by

$$\mathbf{x}_s\left(t\right) \triangleq \left[r_{xs}\left(t\right), r_{ys}\left(t\right), v_{xs}, v_{ys}\right]^{\mathsf{T}}, \tag{88}$$

and

$$\mathbf{x}_o\left(t\right) \triangleq \left[r_{xo}\left(t\right), r_{yo}\left(t\right), v_{xo}\left(t\right), v_{yo}\left(t\right)\right]^{\mathsf{T}}. \tag{89}$$

Also, define the relative target state vector, $\mathbf{x}_a\left(t\right)$, as

$$\mathbf{x}_a\left(t\right) \triangleq \left[r_{xa}\left(t\right), r_{ya}\left(t\right), v_{xa}\left(t\right), v_{ya}\left(t\right)\right]^{\mathsf{T}}. \tag{90}$$

Note that $\mathbf{x}_o\left(t\right)$ and $\vartheta\left(t\right)$ are assumed to be known by the observer.

Now, from Fig.11, we can identify the relative bearing as

$$\theta\left(t_n\right) = \tan^{-1}\left[\frac{r_{xa}\left(t_n\right)}{r_{ya}\left(t_n\right)}\right]. \tag{91}$$

For a constant-velocity target ship, Eq. 1 can be written as

$$\mathbf{x}_s\left(t_n\right) = \left[\begin{array}{cc} \mathbf{I}_2 & T\mathbf{I}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 \end{array}\right]\mathbf{x}_s\left(t_{n-1}\right) + \left[\begin{array}{c} \frac{T^2}{2}\mathbf{I}_2 \\ T\mathbf{I}_2 \end{array}\right]\mathbf{u}\left(t_n\right), \tag{92}$$

where $\mathbf{u}\left(t_n\right) \triangleq \left[u_x\left(t_n\right), u_y\left(t_n\right)\right]^{\mathsf{T}}$, $\mathbf{I}_2$ is a two-dimensional identity matrix, and T is the sample period. The acceleration noise component $\mathbf{u}\left(t_n\right)$ is $\mathcal{N}\left(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{v}}\right)$, with
$\mathbf{\Sigma}_{\mathbf{v}} \triangleq \left\langle \mathbf{u}\left(t_n\right)\mathbf{u}^H\left(t_n\right)\right\rangle = \sigma_{\mathbf{v}}^2\mathbf{I}_2$.

In vector-matrix form, the broadband complex output of the array can be written as

$$\mathbf{y}\left(t\right) = \mathbf{D}\left(\theta\left(t\right)\right)\mathbf{s}\left(t\right) + \mathbf{w}\left(t\right), \tag{93}$$

where $\mathbf{y}\left(t\right)$, $\mathbf{s}\left(t\right)$, and $\mathbf{w}\left(t\right)$ are concatenated complex vectors over element number and the discrete frequencies f_k, $k = 1, ..., K$. The array manifold matrix, $\mathbf{D}\left(\theta\left(t\right)\right)$, is a block diagonal matrix

$$\mathbf{D}\left(\theta\left(t\right)\right) = \left[\begin{array}{cccc} \mathbf{d}\left(f_1;\theta\left(t\right)\right) & & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{d}\left(f_2;\theta\left(t\right)\right) & & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{d}\left(f_K;\theta\left(t\right)\right) \end{array}\right], \tag{94}$$

with steering vector elements, $\mathbf{d}\left(f_k;\theta\left(t\right)\right)$, defined as

$$\mathbf{d}\left(f_k;\theta\left(t\right)\right) \triangleq \left[a_0\left(f_k;t\right), a_1\left(f_k;t\right), \ldots, a_{M-1}\left(f_k;t\right)\right]^{\mathsf{T}}, \tag{95}$$

with $a_m(f_k;t) \triangleq e^{j2\pi f_k \tau_m(\theta(t))}$. Here,

$$\tau_m(\theta(t)) \triangleq -\frac{r_m}{c}\cos\phi_m \cos(\theta_m - \theta(t)), \tag{96}$$

where the spherical coordinates of the m^{th} sensor relative to the array phase center are (r_m, θ_m, ϕ_m) and c is the speed of sound in water.

Let the observation noise, $\mathbf{w}(t)$, be $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_w)$, with $\mathbf{\Sigma}_w = \left\langle \mathbf{w}(t)\mathbf{w}^{\mathrm{H}}(t)\right\rangle = \sigma_w^2 \mathbf{I}_{MK}$. For the signal, $\mathbf{s}(t)$, we will assume the identical structure, with $\mathbf{\Sigma}_s = \sigma_s^2 \mathbf{I}_{MK}$. We will also assume that the signal components are uncorrelated with the noise components, with σ_w^2 and $s(f_k, t)$ known or estimated using a maximum likelihood approach.

Since Eq's. 92 and 93 are of the same form as Eq's. 6 and 7, the Gaussian approximation filter framework developed in the Section 3 can now be applied to this single target tracking problem. Since Eq. 92 is linear, we can simplify the "Generate Prior Samples" and the "Time Update" blocks shown in Figs. 2 and 3 using a Rao-Blackwell procedure [14]. If we let

$$\mathbf{\Phi} \triangleq \begin{bmatrix} \mathbf{I}_2 & T\mathbf{I}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 \end{bmatrix}, \tag{97}$$

prior samples need not be generated and the time update can be accomplished directly, as follows

$$\left\langle \mathbf{x}_{n|n-1}\right\rangle = \mathbf{\Phi}\left\langle \mathbf{x}_{n-1|n-1}\right\rangle, \tag{98}$$

and

$$\mathbf{P}^{\mathbf{xx}}_{n|n-1} = \mathbf{\Phi}\mathbf{P}^{\mathbf{xx}}_{n-1|n-1}\mathbf{\Phi}^{\mathsf{T}} + \mathbf{Q}. \tag{99}$$

In Hue, et. al. [28], a multitarget tracking problem was defined, with four targets tracked by a single observer ship. We have extracted the tracks for one of their targets and the observer ship. The initial positions and velocities of the chosen constant velocity target and observer ships are

$$\mathbf{x}_s(t_0) = \begin{bmatrix} 200 \text{ yds} \\ 1500 \text{ yds} \\ 1.0 \text{ yds/sec} \\ -0.5 \text{ yds/sec} \end{bmatrix} \quad \mathbf{x}_o(t_0) = \begin{bmatrix} 200 \text{ yds} \\ -3000 \text{ yds} \\ 1.2 \text{ yds/sec} \\ +0.5 \text{ yds/sec} \end{bmatrix}.$$

The observer is following a noise-free leg-by-leg trajectory with a constant velocity vector on each leg which is modified at the following instants (see Fig. 9):

$$\begin{aligned}
\begin{pmatrix} v_{xo}(t_{200}) \\ v_{yo}(t_{200}) \end{pmatrix} &= \begin{pmatrix} -0.6 \\ +0.3 \end{pmatrix}; \quad \begin{pmatrix} v_{xo}(t_{400}) \\ v_{yo}(t_{400}) \end{pmatrix} = \begin{pmatrix} +2.0 \\ +0.3 \end{pmatrix}; \\
\begin{pmatrix} v_{xo}(t_{600}) \\ v_{yo}(t_{600}) \end{pmatrix} &= \begin{pmatrix} -0.6 \\ +0.3 \end{pmatrix}; \quad \begin{pmatrix} v_{xo}(t_{800}) \\ v_{yo}(t_{800}) \end{pmatrix} = \begin{pmatrix} +2.0 \\ +0.3 \end{pmatrix}; \\
\begin{pmatrix} v_{xo}(t_{900}) \\ v_{yo}(t_{900}) \end{pmatrix} &= \begin{pmatrix} -0.6 \\ +0.3 \end{pmatrix}.
\end{aligned}$$

The acceleration noise used in the dynamic model is a normal zero-mean Gaussian vector with $\sigma_x = \sigma_y = 2 \times 10^{-3}$ yds/sec^2. A total of 1000 seconds of data was simulated for this example, with $T = 6$ seconds.

The array used for this simulation is a horizontally flat ring array with 60 elements at a radius of 2.5 yards evenly spaced in θ_m, with $\theta_0 = 0^{\text{o}}$ and $\theta_{59} = 354^{\text{o}}$. The noise at each element is a broadband zero-mean complex Gaussian random process with $\sigma_w^2 = 1$. The source signal arriving at all elements is a broadband complex zero mean Gaussian random signal modeled as 31 equally spaced frequency bins across the band 1400 Hz to 2200 Hz. All tracking filters were tested with a signal level adjusted to generate a signal-to-noise ratio (SNR) of -20 dB.

For all tracking filters, $\widehat{\mathbf{x}}_s(t_0)$ was initialized by offsetting the initial value, i.e. $\widehat{\mathbf{x}}_{0|0} \triangleq \mathbf{x}_s(t_0) + \mathbf{q}_s^{\text{off}}$, where $\mathbf{x}_s(t_0)$ is the true initial state and $\mathbf{q}_s^{\text{off}} = [q_x, q_y, 0, 0]^\intercal$. The initialization offset used for this example was designed to maximally stress the tracker. It assumes that the initial bearing is almost known (the initial target position is at 0^{o} true bearing), but that the initial range guess is off considerably, i.e. $q_x = 100$, $q_y = 5000$. The covariance $\mathbf{P}_{0|0}^{\mathbf{xx}}$ was initialized to

$$\mathbf{P}_{0|0}^{\mathbf{xx}} = \begin{bmatrix} 10^3 & 0 & 0 & 0 \\ 0 & 2 \times 10^3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The three nonlinear tracking filters developed in Section 3 have been applied to this problem. Specifically, Fig. 12 presents a generic Kalman filter for the above single target tracking problem.

In our formulation, the observations vector variable $\mathbf{y}$ represents the FFT output of all array elements and is thus complex and of dimension $(M \cdot K) \times 1$. Instead of dealing with complex variables, we transformed $\mathbf{y}$ into a matrix of dimension $(2 \cdot M \cdot K) \times 1$, with the factor of two now representing the in-phase and quadrature components of $\mathbf{y}$. Now, all variables are real and we can apply the Kalman filter equations directly.

A comparison of the outputs of all three nonlinear Kalman filters for this tracking problem are presented in Figure 13. For each subplot, the upper solid line is the true target track, the lower solid jagged line is the observer ship track, the blue dotted lines are observer-to-target line-of-site lines, the dotted line is the target track mean position estimate, and the red ellipsoids are the 1-sigma uncertainty ellipsoids. Examination of the figure shows that all filters converge shortly after the first own-ship maneuver, even though the initialization guess was very long in range. It is obvious from the figure that the results for the UKF and the GHKF are nearly identical and that both have better performance than the MCKF. The performance of the MCKF will improve if the number of Monte Carlo samples is increased from the 2000 samples used for this example. From a computational point of view, the UKF was the fastest followed by the GHKF, with the MCKF requiring the most computation time. The MCKF took approximately 1.5 times the computation time of the UKF, with the GHKF at an increase in time by a factor of 1.2.

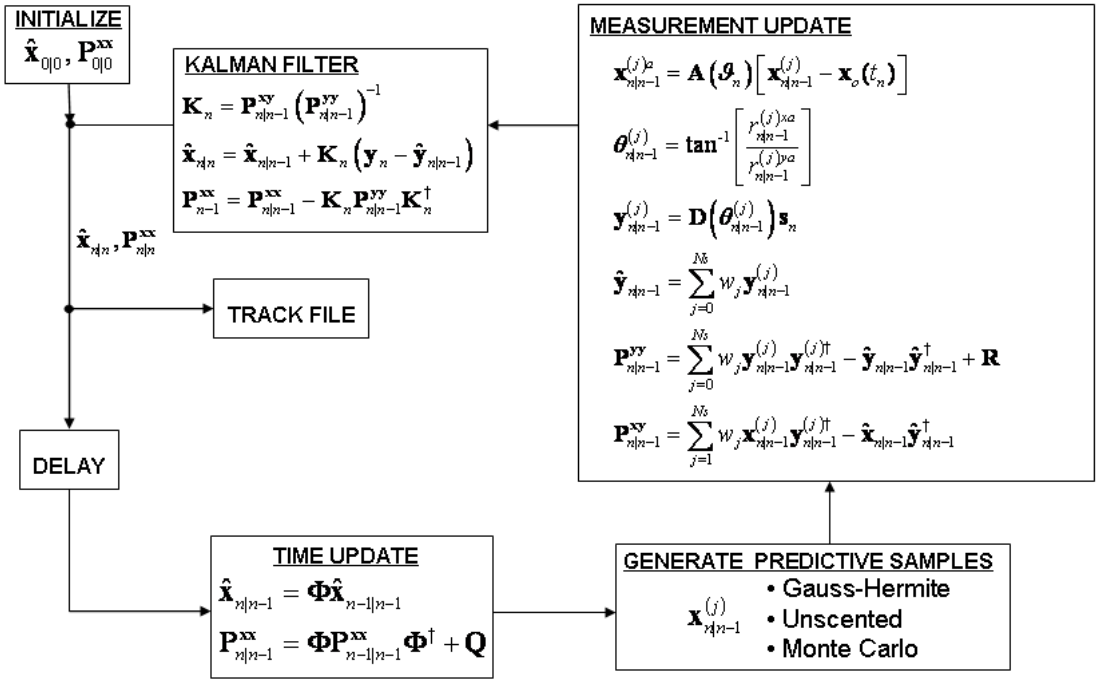

Figure 12: Block Diagram of Nonlinear Track Estimation for a Singler Target

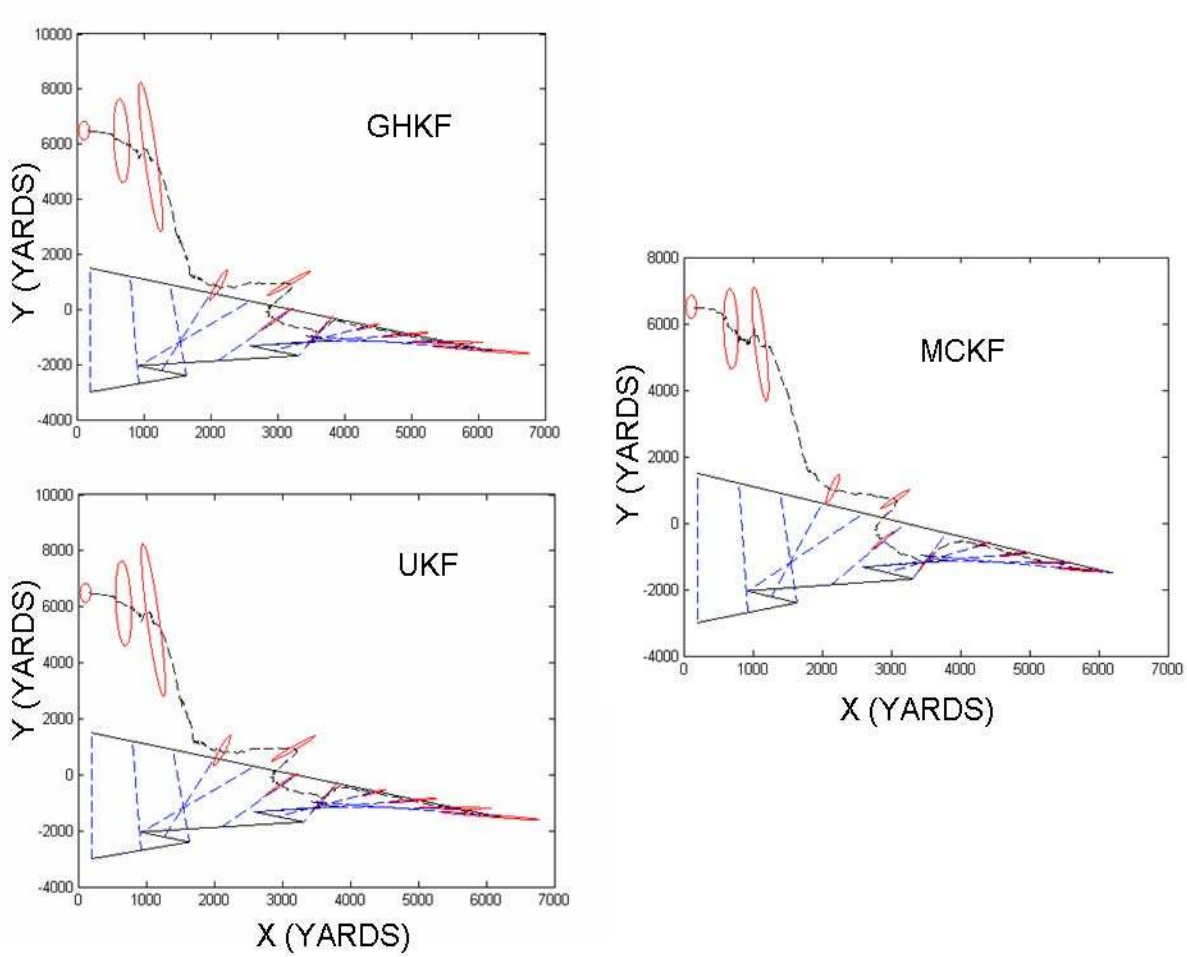

Figure 13: A Comparison of Three Nonlinear Kalman Tracking Filters

6. A Tracking Example with Non-Gaussian Noise Embedded in the Observation Model

Consider the problem of tracking a vehicle moving through a distributed field of acoustic DIFAR sensors. A DIFAR sensor is composed of three elements, a monopole and two dipole sensors, with one dipole sensor oriented in a north-south direction and the other in an east-west direction. The output of the three sensor channels can be expressed [29]-[32] as

$$\begin{aligned} y_o(t) &= s(t) + n_0(t), && \text{(100a)} \\ y_c(t) &= s(t)\cos(\theta(t)) + n_c(t), && \text{(100b)} \\ y_s(t) &= s(t)\sin(\theta(t)) + n_s(t), && \text{(100c)} \end{aligned}$$

where $s(t)$ is the signal received at the monopole sensor and $n_0(t)$, $n_c(t)$,and $n_s(t)$ are the noise components on the three channels. $\theta(t)$ is the bearing of the vehicle at the sensor, relative to true North, with $(0 \leq \theta(t_n) \leq 2\pi)$. In references [29], [31], and [32], a time domain approach is used, while reference [30] uses a frequency domain approach. We will follow the lead of reference [30], and use the frequency domain.

To estimate the bearing observations, the DIFAR channel outputs are passed through an FFT with an integration time $T = t_n - t_{n-1}$, where the output of the FFT for one frequency bin at block time t_n is defined by

$$Y(f;t_n) = \frac{1}{T}\int_{t_{n-1}}^{t_n} y(t)\, e^{-j2\pi ft} dt. \tag{101}$$

After the FFT for each channel, we can write the channel outputs as

$$\begin{aligned} Y_{\mathrm{o}}(f_k;t_n) &= S(f_k;t_n) + N_{\mathrm{o}}(f_k;t_n), && \text{(102a)} \\ Y_{\mathrm{c}}(f_k;t_n) &= S(f_k;t_n)\cos(\theta(t_n)) + N_{\mathrm{c}}(f_k;t_n), && \text{(102b)} \\ Y_{\mathrm{s}}(f_k;t_n) &= S(f_k;t_n)\sin(\theta(t_n)) + N_{\mathrm{s}}(f_k;t_n), && \text{(102c)} \end{aligned}$$

with f_k defined as the frequency of the k^{th} frequency bin, $k = 1, \ldots, N$. Let $N_{\mathrm{o}}(f_k;t_n) \sim \mathcal{N}\left(0, \sigma^2_{N_{\mathrm{o}}}\right)$, $N_{\mathrm{c}}(f_k;t_n) \sim \mathcal{N}\left(0, \rho\sigma^2_{N_{\mathrm{o}}}\right)$, $N_{\mathrm{s}}(f_k;t_n) \sim \mathcal{N}\left(0, \rho\sigma^2_{N_{\mathrm{o}}}\right)$, and $S(f_k;t_n) \sim \mathcal{N}(0, S(f_k;t_n))$, where ρ is a dipole gain factor and $\mathcal{N}(\mu, \sigma^2)$ is a Gaussian distribution with mean μ and variance σ^2. Also assume that all signal and noise components are independent of each other and independent over frequency bin and time index.

Next, the cross-correlations $C_1(f_k;t_n)$ and $C_2(f_k;t_n)$ are formed, where

$$\begin{aligned} C_1(f_k;t_n) &= \mathrm{Re}\left\{Y_{\mathrm{o}}(f_k;t_n)\, Y_{\mathrm{c}}^*(f_k;t_n)\right\}, && \text{(103a)} \\ C_2(f_k;t_n) &= \mathrm{Re}\left\{Y_{\mathrm{o}}(f_k;t_n)\, Y_{\mathrm{s}}^*(f_k;t_n)\right\}. && \text{(103b)} \end{aligned}$$

Then a bearing observation is obtained from

$$\theta(t_n) = \tan^{-1}\left\{\sum_k \frac{C_2(f_k;t_n)}{C_1(f_k;t_n)}\right\}. \tag{104}$$

Note that the sum over frequency constitutes an estimate of the expected value of the correlations outputs. In references [29], [31], and [32], this expectation is accomplished using averaging of samples in the time domain.

For our tracking problem demonstration, we simulated a scenario with a vehicle moving with a constant velocity through a field of 5 DIFAR sensors. We used Eq's. 102 - 104 to generate a vector of simulated noisy observations
$\mathbf{y}(t_n) = \boldsymbol{\theta}(t_n) \triangleq [\theta_1(t_n), \theta_2(t_n), \ldots, \theta_5(t_n)]^{\intercal}$; $t_n = 0, T, \ldots, nT$. For the m^{th} DIFAR sensor at position (x_m, y_m), the true values of $\theta_m(t_n)$ used in Eq. 102 were computed from

$$\theta_m(t_n) = \tan^{-1}\left\{\frac{x(t_n) - x_m}{y(t_n) - y_m}\right\}, \tag{105}$$

with the y-axis pointing North and the x-axis pointing East. Although the noise for each individual channel is additive and Gaussian, the noise in the observation $\boldsymbol{\theta}(t_n)$ is embedded (not additive) and may not necessarily be Gaussian. Thus, estimation and tracking of the expected value of the vehicle state vector, $\langle \mathbf{x}(t_n) \rangle$, from noisy observations from multiple DIFAR sensor outputs, can be most readily accomplished using one of the particle filters developed in Section 4.

If we assume a constant-velocity model for the vehicle motion, we can use the model developed in our previous example. Thus, the state transition equations governing our constant-velocity vehicle are given by Eq. 92. Therefore, since the transition equations are linear, we can use a Rao-Blackwell procedure for simplifying the time update equation. The outputs of the time update equations are $\widehat{\mathbf{x}}_{n|n-1}$ and $\mathbf{P}^{\mathbf{xx}}_{n|n-1}$.

Looking at figures 5 - 8, we see that for all versions of the particle filter we need an analytical expression for the likelihood function, $p\left(\mathbf{y}_n|\mathbf{x}^{(i)}\right)$, where $\mathbf{x}^{(i)} \triangleq \mathbf{x}^{(i)}_{n|n-1}$ are particles obtained from the time update equations for the SIS filters (Fig. 5) and $\mathbf{x}^{(i)} \triangleq \mathbf{x}^{(i)}_{n|n}$ are samples drawn from the importance density for the more general particle filters (Figs. 7 and 8). Davies [29] developed a DIFAR sensor likelihood function for the time domain case which is not applicable here. Mars [32] also developed a time-domain tracking technique for DIFAR sensors that uses a second-order Taylor expansion of the log-likelihood about the prior estimate, which is also not applicable here. Maranda [31] presented frequency-domain likelihood functions, but only for single frequency signals and signals with a Gaussian frequency distribution. Since we wish to track vehicles which radiate wideband acoustic sound, we developed our own wideband likelihood function. The particulars of the development of the wideband likelihood function for a single DIFAR sensor is presented in Appendix B.

First, assuming that the sensors likelihood functions are independent of each other and letting the dependence of $\theta_m(t_n)$ on t_n be understood , then

$$p\left(\mathbf{y}_n|\mathbf{x}^{(i)}\right) \triangleq p\left(\boldsymbol{\theta}|\mathbf{x}^{(i)}\right) = \prod_{m=1}^{5} p\left(\theta_m|\mathbf{x}^{(i)}\right). \tag{106}$$

Putting $\mathbf{x}^{(i)}$ into Eq. 105 yields an expression for $\theta_m^{(i)}$

$$\theta_m^{(i)} = \tan^{-1}\left\{\frac{x^{(i)} - x_m}{y^{(i)} - y_m}\right\}. \tag{107}$$

From Appendix B, if we let $z_m \triangleq \tan\theta_m$, then

$$p\left(\theta_m|\mathbf{x}^{(i)}\right) = \left[1 + (z_m)^2\right] p\left(z_m|\theta_m^{(i)}\right), \tag{108}$$

with

$$p\left(z_m|\theta_m^{(i)}\right) = A\left\{\frac{\sqrt{(1-r^2)}}{\pi}e^{-\alpha} + \frac{\delta\beta^{1/2}}{\sqrt{2\pi}}\left[\Phi\left(\frac{-\delta\beta^{1/2}}{\sqrt{(1-r^2)}}\right) - \Phi\left(\frac{\delta\beta^{1/2}}{\sqrt{(1-r^2)}}\right)\right]e^{-\gamma}\right\}. \tag{109}$$

Here

$$A \triangleq \frac{\widetilde{\sigma}_1\widetilde{\sigma}_2}{\widetilde{\sigma}_2^2 - 2r\widetilde{\sigma}_1\widetilde{\sigma}_2 z_m + \widetilde{\sigma}_1^2 z_m^2}, \tag{110a}$$

$$\alpha \triangleq \frac{\widetilde{\sigma}_2^2\eta_1^2 - 2r\widetilde{\sigma}_1\widetilde{\sigma}_2\widetilde{\eta}_1\widetilde{\eta}_2 + \widetilde{\sigma}_1^2\widetilde{\eta}_2^2}{2\widetilde{\sigma}_1^2\widetilde{\sigma}_2^2(1-r^2)}, \tag{110b}$$

$$\beta \triangleq \frac{\left[\left(\widetilde{\sigma}_2^2\widetilde{\eta}_1 - r\widetilde{\sigma}_1\widetilde{\sigma}_2\widetilde{\eta}_2\right) + \left(\widetilde{\sigma}_1^2\widetilde{\eta}_2 - r\widetilde{\sigma}_1\widetilde{\sigma}_2\widetilde{\eta}_1\right)z_m\right]^2}{\widetilde{\sigma}_1^2\widetilde{\sigma}_2^2\left(\widetilde{\sigma}_2^2 - 2r\widetilde{\sigma}_1\widetilde{\sigma}_2 z_m + \widetilde{\sigma}_1^2 z_m^2\right)}, \tag{110c}$$

$$\gamma \triangleq \frac{\left(\widetilde{\eta}_2 - \widetilde{\eta}_1 z_m\right)^2}{2\left(\widetilde{\sigma}_2^2 - 2r\widetilde{\sigma}_1\widetilde{\sigma}_2 z_m + \widetilde{\sigma}_1^2 z_m^2\right)}, \tag{110d}$$

$$\delta \triangleq \text{sign}\left[\left(\widetilde{\sigma}_2^2\widetilde{\eta}_1 - r\widetilde{\sigma}_1\widetilde{\sigma}_2\widetilde{\eta}_2\right) + \left(\widetilde{\sigma}_1^2\widetilde{\eta}_2 - r\widetilde{\sigma}_1\widetilde{\sigma}_2\widetilde{\eta}_1\right)z_m\right]. \tag{110e}$$

The components of Eq. 110e are given by

$$\widetilde{\eta}_1 \triangleq \sqrt{2BT}\text{SNR}\cos\theta_m^{(i)}, \tag{111}$$

$$\widetilde{\eta}_2 \triangleq \sqrt{2BT}\text{SNR}\sin\theta_m^{(i)}, \tag{112}$$

$$\widetilde{\sigma}_1^2 \triangleq (2\text{SNR}+1)\,\text{SNR}\cos^2\theta_m^{(i)} + \rho\,(\text{SNR}+1), \tag{113}$$

$$\widetilde{\sigma}_2^2 \triangleq (2\text{SNR}+1)\,\text{SNR}\sin^2\theta_m^{(i)} + \rho\,(\text{SNR}+1), \tag{114}$$

$$r \triangleq \frac{(2\text{SNR}+1)\,\text{SNR}\cos\theta_m^{(i)}\sin\theta_m^{(i)}}{\widetilde{\sigma}_1\widetilde{\sigma}_2}, \tag{115}$$

$$\Phi(x) \triangleq \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-u^2/2}du. \tag{116}$$

SNR is the signal-to-noise ratio at the monopole channel, BT is the time-bandwidth product of the vehicle frequency signature and ρ is the noise gain (loss) factor applied to either dipole channel. The value of $\rho = 1/2$ or $1/3$ for 2D-isotropic or 3D-isotropic noise, respectively [31].

Since the particles are reused in the bootstrap filter, only the likelihood function is needed to update the particle weights. But in the more general particle filter approaches shown in Figures 7 and 8, where the particles are not reused, we also need to generate particles (samples) from an importance density. To generate a Gaussian importance density, one of the techniques shown in Section 3 can be used, as long as it is assumed that the noise in the dynamic and observation equations is additive and Gaussian. This is permissible because we will be modifying the Gaussian density at the output of the Kalman filter in the weight update portion of the filter. Thus, in the measurement update we use

$$\widehat{\boldsymbol{\theta}}_{n|n-1} = \sum_i w_i \mathbf{h}\left(\mathbf{x}_{n|n-1}^{(i)}\right) = \sum_i w_i \tan^{-1}\left\{\frac{r_{n|n-1}^{x(i)} - \mathbf{x}_m}{r_{n|n-1}^{y(i)} - \mathbf{y}_m}\right\}, \tag{117}$$

$$\begin{aligned}\mathbf{P}_{n|n-1}^{\boldsymbol{\theta\theta}} &= \sum_i w_i \left(\tan^{-1}\left\{\frac{r_{n|n-1}^{x(i)} - \mathbf{x}_m}{r_{n|n-1}^{y(i)} - \mathbf{y}_m}\right\}\right)\left(\tan^{-1}\left\{\frac{r_{n|n-1}^{x(i)} - \mathbf{x}_m}{r_{n|n-1}^{y(i)} - \mathbf{y}_m}\right\}\right)^{\mathsf{T}} \\ &\quad -\widehat{\boldsymbol{\theta}}_{n|n-1}\widehat{\boldsymbol{\theta}}_{n|n-1}^{\mathsf{T}} + \mathbf{R},\end{aligned} \tag{118}$$

and

$$\mathbf{P}_{n|n-1}^{\mathbf{x}\theta} = \sum_i w_i \mathbf{x}_{n|n-1}^{(i)} \left(\tan^{-1}\left\{\frac{r_{n|n-1}^{x(i)} - \mathbf{x}_m}{r_{n|n-1}^{y(i)} - \mathbf{y}_m}\right\}\right)^{\mathsf{T}} - \widehat{\mathbf{x}}_{n|n-1}\widehat{\boldsymbol{\theta}}_{n|n-1}^{\mathsf{T}}. \tag{119}$$

Here, $\mathbf{x}_{n|n-1}^{(i)}$ are either Monte Carlo samples from $\mathcal{N}\left(\widehat{\mathbf{x}}_{n|n-1}, \mathbf{P}_{n|n-1}^{\mathbf{xx}}\right)$ or Gauss-Hermite quadrature or unscented filter support points obtained using $\widehat{\mathbf{x}}_{n|n-1}$ and $\mathbf{P}_{n|n-1}^{\mathbf{xx}}$. The quantities $r_{n|n-1}^{x(i)}$ and $r_{n|n-1}^{y(i)}$ are the x and y components of $\mathbf{x}_{n|n-1}^{(i)}$, $w_i = 1/N$ for Monte Carlo sampling, and $\mathbf{R}$ is the covariance matrix of the Gaussian observation noise model. The outputs of Eq's. 117-119 are then used as inputs to a Kalman filter. The outputs of the Kalman filter are used to specify a Gaussian importance density $\mathcal{N}\left(\boldsymbol{\mu}_{n|n}^{\mathbf{x}}, \boldsymbol{\Sigma}_{n|n}^{\mathbf{xx}}\right)$ from which samples $\mathbf{x}_{n|n}^{(i)}$, $i = 1, 2, \ldots, N_s$ are drawn. And then it is easy to calculate $\mathcal{N}\left(\mathbf{x}_{n|n}^{(i)}; \boldsymbol{\mu}_{n|n}^{\mathbf{x}}, \boldsymbol{\Sigma}_{n|n}^{\mathbf{xx}}\right)$ for each particle in the weight update equation. In addition, since we already know $\widehat{\mathbf{x}}_{n|n-1}$ and $\mathbf{P}_{n|n-1}^{\mathbf{xx}}$, we can calculate $\mathcal{N}\left(\mathbf{x}_{n|n}^{(i)}; \widehat{\mathbf{x}}_{n|n-1}, \mathbf{P}_{n|n-1}^{\mathbf{xx}}\right)$ directly and everything we need for the weight update equation has been specified. Substitution of the above into figures 7 or 8 yields the appropriate fully realized particle filter for tracking a vehicle through a field of DIFAR sensors.

For our DIFAR vehicle tracking example, we have implemented the Gauss-Hermite and the unscented particle filters. The measurement covariance has been set to one, the time-bandwidth product to 240 and the SNR was varied from -10 dB to +10 dB. A comparison of the true tracks (black solid line) and the estimated tracks (blue dashed line) generated by both trackers for the three different SNRs is shown in Fig. 14, along with the positions of all five DIFAR sensors (green circles). In Fig. 14, the left column contains tracks for the unscented particle filter (UPF) and the right column those of the Gauss-Hermite particle

filter (GHPF), with SNR decreasing from bottom to top. Examination of the figure shows that the tracks of the Gauss-Hermite particle filter are just a little closer to the true track that those of the unscented particle filter at all SNRs. The run time for both trackers was comparable. No results are shown for the Monte Carlo particle filter or the bootstrap particle filter because of numerical instabilities in the tracker implementations.

7. Summary and Conclusions

The techniques presented above for Bayesian estimation of nonlinear processes with non-Gaussian noise is applicable to a wide variety of problems. As we have shown, for the special case where the noise is additive and Gaussian, estimation involves the numerical solution of integrals with nonlinear arguments. Three techniques for this case were developed, the Gauss-Hermite, unscented, and Monte Carlo Kalman filters. All three were applied to the simulated problem of geo-tracking a constant-velocity target using complex broadband sensor-level measurements from an array mounted on a maneuvering observer. All three tracking algorithms successfully started tracking the target immediately following the first observer ship maneuver, with the Gauss-Hermite and unscented Kalman filters performing identically, tracking very well, and the Monte Carlo Kalman filter tracking with slightly less accuracy. From a numerical point of view, the unscented Kalman filter had the shortest run time, followed closely by the Gauss-Hermite Kalman filter, with the Monte Carlo Kalman filter having much longer run times. Based on these results, the unscented Kalman filter would be the best choice for nonlinear estimation in the presence of additive Gaussian noise.

For nonlinear processes in the presence of embedded non-Gaussian noise, we presented four estimation techniques, the SIS (bootstrap), and three combination particle filters. The three combination particle filters use a Gaussian particle filter combined with one of the three nonlinear filters for additive Gaussian noise as an importance sample generator. All four filters were applied to the problem of tracking a moving constant-velocity vehicle through a field of acoustic DIFAR sensors. Several of the filters, specifically the bootstrap and the Monte Carlo particle filters, were plagued by numerical instabilities and could not track the vehicle. Both the Gauss-Hermite and the unscented particle filters tracked the target at all SNRs, with the Gauss-Hermite particle filter having the best track performance and the shortest run times. Based on these results, the Gauss-Hermite particle filter would be the best choice for single target tracking in the presence of non-Gaussian embedded noise.

Many other advances have recently taken place in the field of Bayesian estimation that we have not addressed in this technical memorandum. Some of these advances are contained in special issues of the IEEE Transactions on Signal Processing [33] and the Proceedings of the IEEE [34]. Additional Ph.D. dissertations on various aspects of nonlinear, non-Gaussian estimation and filtering are given in [35]-[37].

For many nonlinear estimation problems, there are slowly varying parameters that must also be estimated. For example, in the geo-tracking example addressed above, we

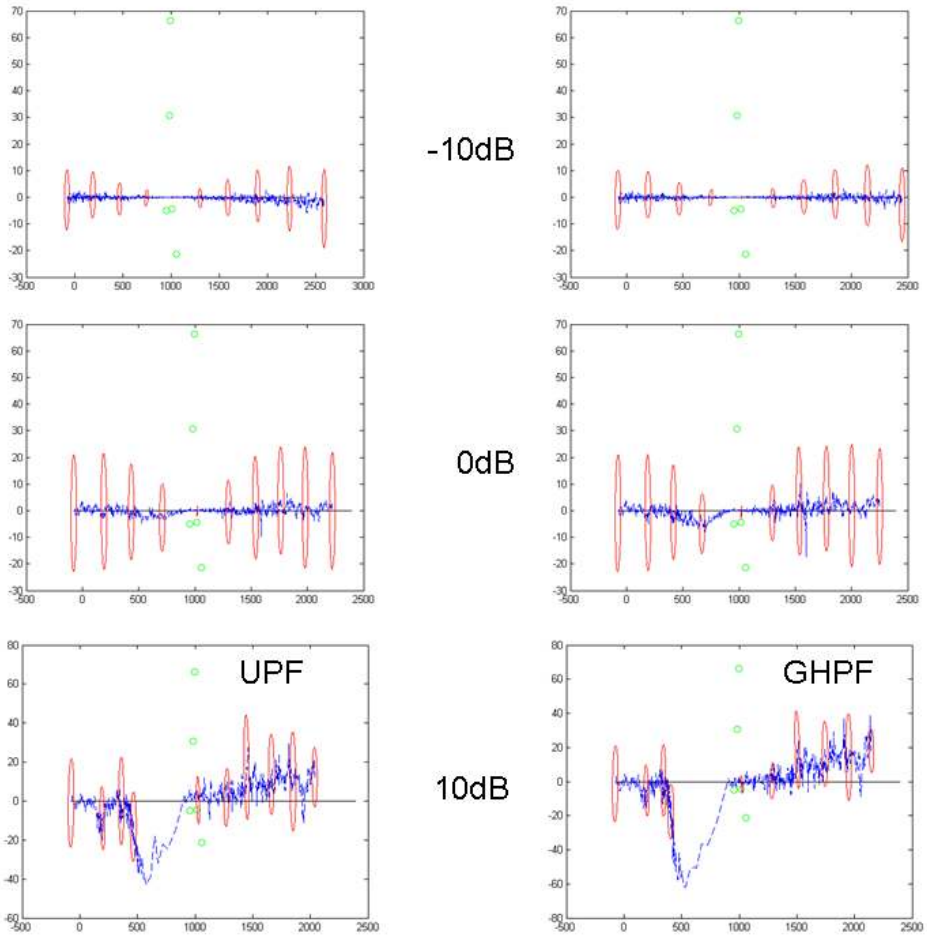

Figure 14: Comparison of GHPF and UPF Tracking Results for Three SNRs.

assumed that the signal and the noise covariance were known. But in a real world problem, they must be estimated simultaneously with the state vector. In addition, estimation of the number of sources present at any given time in a multitarget tracking scenario is also necessary. Several papers address these issues of joint parameter estimation and model change detection [39]-[43].

In this paper, we have addressed nonlinear models with both additive Gaussian and embedded non-Gaussian noise. Recently, the Kalman-Levy filter has been developed for continuous linear models driven by Levy process noise [44]. A Levy process is a continuous stochastic process with independent increments. Examples are Brownian motion, Poisson processes, stable processes (such as Cauchy processes), and subordinators (such as Gamma processes.) They can be used to represent infinitely divisible distributions, which find applications in the analysis of signals associated with compexity theory. Such distributions tend to follow a power-law with much longer tails than the Gaussian distribution and thus give rise to impulse-like signals. Gordon, et. al., presented an extended Kalman-Levy filter and applied it to the tracking of maneuvering targets [45]. Maneuvering targets, like a fighter jet or an anti-ship missile, have trajectories that exhibit relatively long periods of quiescent motion interspersed with high acceleration turns and should be well matched to the heavy-tailed system noise models.

Appendix

A. A Kalman Filter for Nonlinear and Complex Observation Processes

The posterior (conditional) density is the PDF of $\mathbf{x}(t_n) \triangleq \mathbf{x}(n)$ given the observations $\mathbf{y}_{1:n} \triangleq \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$. It can be written in terms of the joint density of $\mathbf{x}(n)$ and $\mathbf{y}_{1:n}$ as

$$p(\mathbf{x}(n)|\mathbf{y}_{1:n}) = \frac{p(\mathbf{x}(n), \mathbf{y}_{1:n})}{p(\mathbf{y}_{1:n})}. \tag{A-1}$$

Since the left-hand side is a density defined in terms of the real variable $\mathbf{x}(n)$, the right-hand side must also be written in terms of real variables. For the case were $p(\mathbf{x}(n)|\mathbf{y}_{1:n})$ is a normalized Gaussian PDF, if we define a normalized jointly Gaussian PDF $p(\mathbf{x}(n), \mathbf{y}_{1:n})$ such that the normalization constant $p(\mathbf{y}_{1:n})$ can be ignored, then

$$p(\mathbf{x}(n)|\mathbf{y}_{1:n}) = p(\mathbf{x}(n), \mathbf{y}_{1:n}). \tag{A-2}$$

We will approximate the joint density $p(\mathbf{x}(n), \mathbf{y}_{1:n})$ by the predictive density, i.e.,

$$p(\mathbf{x}(n), \mathbf{y}_{1:n}) \simeq p(\mathbf{x}(n), \mathbf{y}(n)|\mathbf{x}(n-1), \mathbf{y}_{1:n-1}). \tag{A-3}$$

Now, let

$$\widehat{\mathbf{x}} \triangleq \widehat{\mathbf{x}}_{n|n} = \mathcal{E}\{\mathbf{x}(n)|\mathbf{y}_{1:n}\}, \tag{A-4}$$

$$\mathbf{P}^{\widehat{\mathbf{x}}\widehat{\mathbf{x}}} \triangleq \mathbf{P}^{\mathbf{x}\mathbf{x}}_{n|n} = \mathcal{E}\{[\mathbf{x}(n) - \widehat{\mathbf{x}}][\mathbf{x}(n) - \widehat{\mathbf{x}}]^{\intercal}|\mathbf{y}_{1:n}\}. \tag{A-5}$$

The Gaussian posterior density can then be written as

$$\begin{aligned} p(\mathbf{x}(n)|\mathbf{y}_{1:n}) &= \mathcal{N}\left(\widehat{\mathbf{x}}, \mathbf{P}^{\widehat{\mathbf{x}}\widehat{\mathbf{x}}}\right) \\ &= \frac{1}{(2\pi)^{N_x/2}\left|\mathbf{P}^{\widehat{\mathbf{x}}\widehat{\mathbf{x}}}\right|^{1/2}} \exp\left\{-\frac{1}{2}A\right\}, \end{aligned} \tag{A-6}$$

where N_x is the dimension of $\mathbf{x}$, and

$$\begin{aligned} A &\triangleq [\mathbf{x}(n) - \widehat{\mathbf{x}}]^{\intercal}\left(\mathbf{P}^{\widehat{\mathbf{x}}\widehat{\mathbf{x}}}\right)^{-1}[\mathbf{x}(n) - \widehat{\mathbf{x}}] \\ &= \mathbf{x}^{\intercal}(n)\left(\mathbf{P}^{\widehat{\mathbf{x}}\widehat{\mathbf{x}}}\right)^{-1}\mathbf{x}(n) - \mathbf{x}^{\intercal}(n)\left(\mathbf{P}^{\widehat{\mathbf{x}}\widehat{\mathbf{x}}}\right)^{-1}\widehat{\mathbf{x}} \\ &\quad -\widehat{\mathbf{x}}^{\intercal}\left(\mathbf{P}^{\widehat{\mathbf{x}}\widehat{\mathbf{x}}}\right)^{-1}\mathbf{x}(n) + \widehat{\mathbf{x}}^{\intercal}\left(\mathbf{P}^{\widehat{\mathbf{x}}\widehat{\mathbf{x}}}\right)^{-1}\widehat{\mathbf{x}}. \end{aligned} \tag{A-7}$$

Returning to the joint density $p(\mathbf{x}(n), \mathbf{y}(n))$, we want to address the case where $\mathbf{y}(n)$ is complex. Since the joint PDF must be written in terms of real variables, let

$$\mathbf{y}(n) \triangleq \begin{bmatrix} \mathbf{y}^{\mathrm{I}}(n) \\ \mathbf{y}^{\mathrm{Q}}(n) \end{bmatrix}, \tag{A-8}$$

where $\mathbf{y}^{\mathrm{I}}(n)$ and $\mathbf{y}^{\mathrm{Q}}(n)$ are the in-phase and quadrature parts of $\mathbf{y}(n)$, respectively. Now define the joint vector

$$\mathbf{z}(n) \triangleq \begin{bmatrix} \mathbf{x}(n) \\ \mathbf{y}(n) \end{bmatrix}. \tag{A-9}$$

Assuming that the joint PDF is Gaussian,

$$p(\mathbf{z}(n)) \sim \mathcal{N}(\overline{\mathbf{z}}(n), \mathbf{P}^{\mathbf{zz}}), \tag{A-10}$$

where

$$\overline{\mathbf{z}}(n) = \begin{bmatrix} \overline{\mathbf{x}}(n) \\ \overline{\mathbf{y}}(n) \end{bmatrix} \triangleq \mathcal{E}\left\{ \begin{bmatrix} \mathbf{x}(n) \\ \mathbf{y}(n) \end{bmatrix} | \mathbf{x}(n-1), \mathbf{y}_{1:n-1} \right\} = \begin{bmatrix} \widehat{\mathbf{x}}_{n|n-1} \\ \widehat{\mathbf{y}}_{n|n-1} \end{bmatrix}, \tag{A-11}$$

and

$$\begin{aligned} \mathbf{P}^{\mathbf{zz}} &= \mathcal{E}\left\{ [\mathbf{z}(n) - \overline{\mathbf{z}}(n)] [\mathbf{z}(n) - \overline{\mathbf{z}}(n)]^{\intercal} | \mathbf{x}(n-1), \mathbf{y}_{1:n-1} \right\} \\ &= \begin{bmatrix} \mathbf{P}^{\mathbf{xx}} & \mathbf{P}^{\mathbf{xy}} \\ \mathbf{P}^{\mathbf{yx}} & \mathbf{P}^{\mathbf{yy}} \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{P}^{\mathbf{xx}}_{n|n-1} & \mathbf{P}^{\mathbf{xy}}_{n|n-1} \\ \mathbf{P}^{\mathbf{yx}}_{n|n-1} & \mathbf{P}^{\mathbf{yy}}_{n|n-1} \end{bmatrix}. \end{aligned} \tag{A-12}$$

The inverse of $\mathbf{P}^{\mathbf{zz}}$ is given by

$$(\mathbf{P}^{\mathbf{zz}})^{-1} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix}, \tag{A-13}$$

with

$$\mathbf{C}_{11} \triangleq \left(\mathbf{P}^{\mathbf{xx}} - \mathbf{P}^{\mathbf{xy}} (\mathbf{P}^{\mathbf{yy}})^{-1} \mathbf{P}^{\mathbf{yx}} \right)^{-1}, \tag{A-14a}$$

$$\mathbf{C}_{12} \triangleq -\mathbf{C}_{11} \mathbf{P}^{\mathbf{xy}} (\mathbf{P}^{\mathbf{yy}})^{-1}, \tag{A14b}$$

$$\mathbf{C}_{21} \triangleq -\mathbf{C}_{22} \mathbf{P}^{\mathbf{yx}} (\mathbf{P}^{\mathbf{xx}})^{-1}, \tag{A-14c}$$

$$\mathbf{C}_{22} \triangleq \left(\mathbf{P}^{\mathbf{yy}} - \mathbf{P}^{\mathbf{yx}} (\mathbf{P}^{\mathbf{xx}})^{-1} \mathbf{P}^{\mathbf{xy}} \right)^{-1}. \tag{A-14d}$$

Thus, the joint PDF is

$$p(\mathbf{z}(n)) = \frac{1}{(2\pi)^{(N_x+2N_y)/2} |\mathbf{P}^{zz}|^{1/2}} \exp\left\{ -\frac{1}{2} B \right\}, \tag{A-15}$$

with

$$\begin{aligned} B &\triangleq \left[(\mathbf{x}(n) - \overline{\mathbf{x}}(n)) \vdots (\mathbf{y}(n) - \overline{\mathbf{y}}(n)) \right]^{\intercal} (\mathbf{P}^{\mathbf{zz}})^{-1} \left[(\mathbf{x}(n) - \overline{\mathbf{x}}(n)) \vdots (\mathbf{y}(n) - \overline{\mathbf{y}}(n)) \right] \\ &= [\mathbf{x}(n) - \overline{\mathbf{x}}(n)]^{\intercal} \mathbf{C}_{11} [\mathbf{x}(n) - \overline{\mathbf{x}}(n)] + [\mathbf{x}(n) - \overline{\mathbf{x}}(n)]^{\intercal} \mathbf{C}_{12} [\mathbf{y}(n) - \overline{\mathbf{y}}(n)] \\ &\quad + [\mathbf{y}(n) - \overline{\mathbf{y}}(n)]^{\intercal} \mathbf{C}_{21} [\mathbf{x}(n) - \overline{\mathbf{x}}(n)] + [\mathbf{y}(n) - \overline{\mathbf{y}}(n)]^{\intercal} \mathbf{C}_{21} [\mathbf{y}(n) - \overline{\mathbf{y}}(n)] \\ &= \mathbf{x}^{\intercal}(n) \mathbf{C}_{11} \mathbf{x}(n) \\ &\quad + \mathbf{x}^{\intercal}(n) [-\mathbf{C}_{11} \overline{\mathbf{x}}(n) + \mathbf{C}_{12} (\mathbf{y}(n) - \overline{\mathbf{y}}(n))] \\ &\quad + \cdots. \end{aligned} \tag{A-16}$$

Comparing the first term of Eq. A-7 with the first term of Eq. A-16 yields

$$\left(\mathbf{P}^{\widehat{\mathbf{xx}}}\right)^{-1} = \mathbf{C}_{11} = \left(\mathbf{P}^{\mathbf{xx}} - \mathbf{P}^{\mathbf{xy}}\left(\mathbf{P}^{\mathbf{yy}}\right)^{-1}\mathbf{P}^{\mathbf{yx}}\right)^{-1}. \tag{A-17}$$

Thus,

$$\mathbf{P}^{\widehat{\mathbf{xx}}} = \mathbf{P}^{\mathbf{xx}} - \mathbf{P}^{\mathbf{xy}}\left(\mathbf{P}^{\mathbf{yy}}\right)^{-1}\mathbf{P}^{\mathbf{yx}}. \tag{A-18}$$

Comparing the second term of Eq. A-7 with the second term of Eq. A-16 results in

$$\begin{aligned}\left(\mathbf{P}^{\widehat{\mathbf{xx}}}\right)^{-1}\widehat{\mathbf{x}} &= \mathbf{C}_{11}\overline{\mathbf{x}}(n) - \mathbf{C}_{12}\left(\mathbf{y}(n) - \overline{\mathbf{y}}(n)\right), \\ &= \left(\mathbf{P}^{\mathbf{xx}} - \mathbf{P}^{\mathbf{xy}}\left(\mathbf{P}^{\mathbf{yy}}\right)^{-1}\mathbf{P}^{\mathbf{yx}}\right)^{-1}\overline{\mathbf{x}}(n) \\ &\quad + \left(\mathbf{P}^{\mathbf{xx}} - \mathbf{P}^{\mathbf{xy}}\left(\mathbf{P}^{\mathbf{yy}}\right)^{-1}\mathbf{P}^{\mathbf{yx}}\right)^{-1}\mathbf{P}^{\mathbf{xy}}\left(\mathbf{P}^{\mathbf{yy}}\right)^{-1}\left(\mathbf{y}(n) - \overline{\mathbf{y}}(n)\right). \end{aligned} \tag{A-19}$$

Solving for $\widehat{\mathbf{x}}$ and using Eq. A-18 yields

$$\widehat{\mathbf{x}} = \overline{\mathbf{x}}(n) + \mathbf{P}^{\mathbf{xy}}\left(\mathbf{P}^{\mathbf{yy}}\right)^{-1}\left[\mathbf{y}(n) - \overline{\mathbf{y}}(n)\right]. \tag{A-20}$$

Now, defining the Kalman Gain as

$$\mathbf{K}_n \triangleq \mathbf{P}^{\mathbf{xy}}\left(\mathbf{P}^{\mathbf{yy}}\right)^{-1}, \tag{A-21}$$

and using Eq's. A-18, A-20, and A-21, the Kalman filter equations for complex observations are given by

$$\widehat{\mathbf{x}}_{n|n} = \widehat{\mathbf{x}}_{n|n-1} + \mathbf{K}_n\left[\mathbf{y}(n) - \widehat{\mathbf{y}}_{n|n-1}\right] \tag{A-22}$$

$$\mathbf{P}^{\mathbf{xx}}_{n|n} = \mathbf{P}^{\mathbf{xx}}_{n|n-1} - \mathbf{K}_n\mathbf{P}^{\mathbf{yy}}_{n|n-1}\mathbf{K}_n^{\intercal}, \tag{A-23}$$

where

$$\mathbf{P}^{\mathbf{xx}}_{n|n-1} = \mathcal{E}\left\{\left[\mathbf{x}_{n|n-1} - \widehat{\mathbf{x}}_{n|n-1}\right]\left[\mathbf{x}_{n|n-1} - \widehat{\mathbf{x}}_{n|n-1}\right]^{\intercal}\right\} \tag{A-24}$$

$$\mathbf{P}^{\mathbf{xy}}_{n|n-1} = \mathcal{E}\left\{\left[\mathbf{x}_{n|n-1} - \widehat{\mathbf{x}}_{n|n-1}\right]\left[\mathbf{y}_{n|n-1} - \widehat{\mathbf{y}}_{n|n-1}\right]^{\intercal}\right\} \tag{A-25}$$

$$\mathbf{P}^{\mathbf{yy}}_{n|n-1} = \mathcal{E}\left\{\left[\mathbf{y}_{n|n-1} - \widehat{\mathbf{y}}_{n|n-1}\right]\left[\mathbf{y}_{n|n-1} - \widehat{\mathbf{y}}_{n|n-1}\right]^{\intercal}\right\}. \tag{A-26}$$

B. Derivation of the Likelihood Function for a DIFAR Bearing Observation

We start by considering $\theta(t) = \tan^{-1}\left(C_2(t)/C_1(t)\right)$, with $C_1(t) \sim \mathcal{N}(\eta_1, \sigma_1^2)$ and $C_2(t) \sim \mathcal{N}(\eta_2, \sigma_2^2)$. $C_1(t)$ and $C_2(t)$ are defined by the time domain equivalent of Eq. 103b. We will first determine the probability density $p_z(z)$ when $z = C_2(t)/C_1(t)$ and then the likelihood function $p\left(\theta(t_n)\,|\mathbf{x}_{n|n}\right)$, where $\theta(t_n)$ is the bearing output of the DIFAR sensor at time t_n and $\mathbf{x}_{n|n}$ is the estimated vehicle position at time t_n based on all observations up to and including time t_n.

Consider the case where $C_2(t)/C_1(t) \leq z$. Then, it follows that:

$$\begin{aligned} C_2(t) &\leq zC_1(t) \text{ if } C_1(t) \geq 0 \\ C_2(t) &\geq zC_1(t) \text{ if } C_1(t) \leq 0. \end{aligned} \tag{B-1}$$

The distribution function $F_z(z)$ can therefore be written as

$$F_z(z) = \int_0^{\infty} \int_{-\infty}^{C_1 z} p(C_1, C_2)\, dC_2 dC_1 + \int_{-\infty}^{0} \int_{C_1 z}^{\infty} p(C_1, C_2)\, dC_2 dC_1. \tag{B-2}$$

Then, the probability density function can be obtained from

$$\begin{aligned} p_z(z) &\triangleq \frac{d}{dz} F_z(z) = \int_0^{\infty} C_1 p(C_1, C_1 z)\, dC_1 - \int_{-\infty}^{0} C_1 p(C_1, C_1 z)\, dC_1 \\ &= \int_0^{\infty} C_1 p(C_1, C_1 z)\, dC_1 + \int_0^{\infty} C_1 p(-C_1, -C_1 z)\, dC_1. \end{aligned} \tag{B-3}$$

Since C_1 and C_2 are jointly Gaussian

$$p(C_1, C_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-r^2}} \exp\left\{-\frac{1}{2(1-r^2)}\left[\frac{(C_1-\eta_1)^2}{\sigma_1^2} - \frac{2r(C_1-\eta_1)(C_2-\eta_2)}{\sigma_1\sigma_2} + \frac{(C_2-\eta_2)^2}{\sigma_2^2}\right]\right\}, \tag{B-4}$$

where r is the normalized correlation coefficient. If we note that $C_2 = C_1 z$ and expand the exponential and complete the square, $p(C_1, C_2)$ can be written as

$$\begin{aligned} p(C_1, C_1 z) &= B \exp\left\{-\frac{\alpha_1}{2(1-r^2)}\left(C_1 - \frac{\alpha_2}{\alpha_1}\right)^2\right\} \\ &\quad \times \exp\left\{-\frac{1}{2(1-r^2)}\left(\alpha_3 - \frac{\alpha_2^2}{\alpha_1}\right)\right\}, \end{aligned} \tag{B-5}$$

where

$$B \triangleq \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-r^2}} \tag{B-6a}$$

$$\alpha_1 \triangleq \frac{1}{\sigma_1^2} - \frac{2rz}{\sigma_1\sigma_2} + \frac{z^2}{\sigma_2^2}, \tag{B-6b}$$

$$\alpha_2 \triangleq \frac{\eta_1}{\sigma_1^2} - \frac{r(\eta_1 z + \eta_2)}{\sigma_1\sigma_2} + \frac{\eta_2 z}{\sigma_2^2}, \tag{B-6c}$$

$$\alpha_3 \triangleq \frac{\eta_1^2}{\sigma_1^2} - \frac{2r\eta_1\eta_2}{\sigma_1\sigma_2} + \frac{\eta_2^2}{\sigma_2^2}. \tag{B-6d}$$

Now, the first half of Eq. B-3 becomes

$$\int_0^{\infty} C_1 p(C_1, C_1 z)\, dC_1 = B \exp\left\{-\frac{1}{2(1-r^2)}\left(\alpha_3 - \frac{\alpha_2^2}{\alpha_1}\right)\right\} I, \tag{B-7}$$

where

$$I \triangleq \int_0^{\infty} C_1 \exp\left\{-\frac{\alpha_1}{2(1-r^2)}\left(C_1 - \frac{\alpha_2}{\alpha_1}\right)^2\right\} dC_1. \tag{B-8}$$

Now, we modify I in the following way:

$$\begin{aligned} I &= \int_0^\infty \left(C_1 - \frac{\alpha_2}{\alpha_1}\right) \exp\left\{-\frac{\alpha_1}{2\left(1-r^2\right)}\left(C_1 - \frac{\alpha_2}{\alpha_1}\right)^2\right\} dC_1 \\ &\quad + \int_0^\infty \frac{\alpha_2}{\alpha_1} \exp\left\{-\frac{\alpha_1}{2\left(1-r^2\right)}\left(C_1 - \frac{\alpha_2}{\alpha_1}\right)^2\right\} dC_1. \end{aligned} \tag{B-9}$$

Defining $u \triangleq C_1 - \frac{\alpha_2}{\alpha_1}$, this becomes

$$I = \int_{-\frac{\alpha_2}{\alpha_1}}^\infty \exp\left\{-\frac{\alpha_1}{2\left(1-r^2\right)}u^2\right\} u du + \frac{\alpha_2}{\alpha_1}\int_{-\frac{\alpha_2}{\alpha_1}}^\infty \exp\left\{-\frac{\alpha_1}{2\left(1-r^2\right)}u^2\right\} du. \tag{B-10}$$

The first integral can be evaluated directly to yield

$$\int_{-\frac{\alpha_2}{\alpha_1}}^\infty \exp\left\{-\frac{\alpha_1}{2\left(1-r^2\right)}u^2\right\} u du = \frac{1-r^2}{\alpha_1}\exp\left\{-\frac{\alpha_2^2/\alpha_1}{2\left(1-r^2\right)}\right\}. \tag{B-11}$$

If we let $v \triangleq \sqrt{\frac{\alpha_1}{(1-r^2)}}u$, then the second integral in Eq. B-10 becomes

$$\frac{\alpha_2}{\alpha_1}\int_{-\frac{\alpha_2}{\alpha_1}}^\infty \exp\left\{-\frac{\alpha_1}{2\left(1-r^2\right)}u^2\right\} du = \delta\left(\alpha_2\right)\sqrt{\frac{\alpha_2^2\left(1-r^2\right)}{\alpha_1^3}}\int_{\delta(\alpha_2)\sqrt{\frac{\alpha_2^2/\alpha_1}{1-r^2}}}^\infty e^{-v^2/2}dv, \tag{B-12}$$

where $\delta\left(x\right) \triangleq$ sign of x. Defining a function $\Phi\left(x\right)$ (related to the error function) as

$$\Phi\left(x\right) \triangleq \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-u^2/2}du, \tag{B-13}$$

we can write Eq. B-12 as

$$\frac{\alpha_2}{\alpha_1}\int_{-\frac{\alpha_2}{\alpha_1}}^\infty \exp\left\{-\frac{\alpha_1}{2(1-r^2)}u^2\right\} du = \delta\left(\alpha_2\right)\sqrt{\frac{2\pi\alpha_2^2\left(1-r^2\right)}{\alpha_1^3}}\Phi\left(-\delta\left(\alpha_2\right)\sqrt{\frac{\alpha_2^2/\alpha_1}{1-r^2}}\right). \tag{B-14}$$

Substituting Eq's. B-11 and B-14 into Eq. B-10 yields

$$I = \frac{1-r^2}{\alpha_1}\exp\left\{-\frac{\alpha_2^2/\alpha_1}{2(1-r^2)}\right\} + \delta\left(\alpha_2\right)\sqrt{\frac{2\pi\alpha_2^2\left(1-r^2\right)}{\alpha_1^3}}\Phi\left(-\delta\left(\alpha_2\right)\left[\frac{\alpha_2^2/\alpha_1}{2(1-r^2)}\right]^{1/2}\right), \tag{B-15}$$

and Eq. B-7 becomes

$$\begin{aligned} \int_0^\infty C_1 p\left(C_1, C_1 z\right) dC_1 &= B\tfrac{1-r^2}{\alpha_1}\exp\left\{-\tfrac{\alpha_3}{2(1-r^2)}\right\} + \delta\left(\alpha_2\right)B\sqrt{\tfrac{2\pi\alpha_2^2\left(1-r^2\right)}{\alpha_1^3}} \\ &\quad \times \exp\left\{-\tfrac{1}{2(1-r^2)}\left(\alpha_3 - \tfrac{\alpha_2^2}{\alpha_1}\right)\right\}\Phi\left(-\delta\left(\alpha_2\right)\left[\tfrac{\alpha_2^2/\alpha_1}{2(1-r^2)}\right]^{1/2}\right) \end{aligned} \tag{B-16}$$

Define

$$\alpha\left(\sigma_1,\sigma_2,\eta_1,\eta_2,r\right) \triangleq \frac{\alpha_3}{2\left(1-r^2\right)}=\frac{\sigma_2^2\eta_1^2-2r\sigma_1\sigma_2\eta_1\eta_2+\sigma_1^2\eta_2^2}{2\sigma_1^2\sigma_2^2\left(1-r^2\right)}, \tag{B-17}$$

$$\beta\left(z;\sigma_1,\sigma_2,\eta_1,\eta_2,r\right) \triangleq \frac{\alpha_2^2}{\alpha_1}=\frac{\left[\left(\sigma_2^2\eta_1-r\sigma_1\sigma_2\eta_2\right)+\left(\sigma_1^2\eta_2-r\sigma_1\sigma_2\eta_1\right)z\right]^2}{\sigma_1^2\sigma_2^2\left(\sigma_2^2-2r\sigma_1\sigma_2z+\sigma_1^2z^2\right)}, \tag{B-18}$$

$$\gamma\left(z;\sigma_1,\sigma_2,\eta_1,\eta_2,r\right) \triangleq \frac{1}{2(1-r^2)}\left(\alpha_3-\frac{\alpha_2^2}{\alpha_1}\right)=\frac{\left(\eta_2-\eta_1z\right)^2}{2\left(\sigma_2^2-2r\sigma_1\sigma_2z+\sigma_1^2z^2\right)}, \tag{B-19}$$

$$\delta\left(z;\sigma_1,\sigma_2,\eta_1,\eta_2,r\right) = \operatorname{sign}\left[\left(\sigma_2^2\eta_1-r\sigma_1\sigma_2\eta_2\right)+\left(\sigma_1^2\eta_2-r\sigma_1\sigma_2\eta_1\right)z\right]. \tag{B-20}$$

After extensive algebra, using the above definitions, we obtain

$$\begin{aligned}\int_0^\infty C_1p\left(C_1,C_1z\right)dC_1 &= \frac{\left(1-r^2\right)^{1/2}\sigma_1\sigma_2}{2\pi\left(\sigma_2^2-2r\sigma_1\sigma_2z+\sigma_1^2z^2\right)}e^{-\alpha(\sigma_1,\sigma_2,\eta_1,\eta_2,r)}\\ &\quad+\delta\frac{\sigma_1\sigma_2\beta(z;\sigma_1,\sigma_2,\eta_1,\eta_2,r)^{1/2}}{\sqrt{2\pi}\left(\sigma_2^2-2r\sigma_1\sigma_2z+\sigma_1^2z^2\right)}e^{-\gamma(z;\sigma_1,\sigma_2,\eta_1,\eta_2,r)}\\ &\quad\times\Phi\left(-\delta\left(z;\sigma_1,\sigma_2,\eta_1,\eta_2,r\right)\frac{\beta(z;\sigma_1,\sigma_2,\eta_1,\eta_2,r)^{1/2}}{(1-r^2)}\right).\end{aligned} \tag{B-21}$$

Returning to Eq. B-7, we can note that $\int_0^\infty C_1p\left(-C_1,-C_1z\right)dC_1$ is identical in form to $\int_0^\infty C_1p\left(C_1,C_1z\right)dC_1$ if we replace η_1 by $-\eta_1$ and η_2 by $-\eta_2$. Thus, we can finally write

$$\begin{aligned}p_z\left(z\right) &= \frac{\sigma_1\sigma_2}{\left(\sigma_2^2-2r\sigma_1\sigma_2z+\sigma_1^2z^2\right)}\left\{\frac{\left(1-r^2\right)^{1/2}}{2\pi}\left[e^{-\alpha(\sigma_1,\sigma_2,\eta_1,\eta_2,r)}+e^{-\alpha(\sigma_1,\sigma_2,-\eta_1,-\eta_2,r)}\right]\right.\\ &\quad+\frac{\delta}{\sqrt{2\pi}}\beta\left(z;\sigma_1,\sigma_2,\eta_1,\eta_2,r\right)^{1/2}e^{-\gamma(z;\sigma_1,\sigma_2,\eta_1,\eta_2,r)}\\ &\quad\times\Phi\left(-\delta\left(z;\sigma_1,\sigma_2,\eta_1,\eta_2,r\right)\frac{\beta(z;\sigma_1,\sigma_2,\eta_1,\eta_2,r)^{1/2}}{(1-r^2)}\right)\\ &\quad-\frac{\delta}{\sqrt{2\pi}}\beta\left(z;\sigma_1,\sigma_2,-\eta_1,-\eta_2,r\right)^{1/2}e^{-\gamma(z;\sigma_1,\sigma_2-,\eta_1,-\eta_2,r)}\\ &\quad\left.\times\Phi\left(\delta\left(z;\sigma_1,\sigma_2,-\eta_1,-\eta_2,r\right)\frac{\beta(z;\sigma_1,\sigma_2,-\eta_1,-\eta_2,r)^{1/2}}{(1-r^2)}\right)\right\},\end{aligned} \tag{B-22}$$

where we have used the fact that $\delta\left(z;\sigma_1,\sigma_2,-\eta_1,-\eta_2,r\right)=-\delta\left(z;\sigma_1,\sigma_2,\eta_1,\eta_2,r\right)$.

The likelihood function of the form $p_\theta\left(\theta\left(t\right)|\theta_{n|n}\right)$ is given by

$$p_\theta\left(\theta\left(t\right)|\theta_{n|n}\right)=\frac{p_z\left(z\left(t\right)|\theta_{n|n}\right)}{\left|\frac{d\theta(t)}{dz(t)}\right|}, \tag{B-23}$$

where $\theta\left(t\right)$ is the output of the DIFAR sensor at time t, $\theta_{n|n}$ is the estimate of θ obtained from $\mathbf{x}_{n|n}$ using Eq. 107, and $z=\tan\theta\left(t\right)$. In what follows, let $\theta\left(t\right)\rightarrow\theta$ and $\theta_{n|n}\rightarrow\theta_0$. Note that η_i, σ_i, and r are functions of θ_0. It follows immediately that

$$p_\theta\left(\theta|\theta_0\right)=p_z\left(z|\theta_0\right)\left(1+z^2\right), \tag{B-24}$$

where $0 \leq \theta \leq 2\pi$. We must now determine $\eta_1, \eta_2, \sigma_1, \sigma_2$, and r.

Examine the correlation output defined by

$$C_i = \frac{1}{T}\int_{-T/2}^{T/2} y_i(t)\, y_0(t)\, dt;\ i = 1, 2, \tag{B-25}$$

where

$$\begin{aligned} y_0(t) &\triangleq s(t) + n_0(t), && \text{(B.4a)} \\ y_1(t) &\triangleq s(t)\cos\theta_0 + n_1(t), && \text{(B.4b)} \\ y_2(t) &\triangleq s(t)\sin\theta_0 + n_2(t). && \text{(B-26)} \end{aligned}$$

Assume that $s(t), n_0(t), n_1(t)$, and $n_2(t)$ are independent zero-mean Gaussian processes. Now

$$\begin{aligned} \eta_1 &= \mathcal{E}\{C_1\} = \mathcal{E}\left\{s^2(t)\right\}\cos\theta_0 \\ &= \cos\theta_0 \int_{-\infty}^{\infty} \mathcal{S}_s(f)\, df, \end{aligned} \tag{B-27}$$

where $\mathcal{S}_s(f)$ is the source spectrum. Now examine

$$\mathcal{E}\left\{C_i^2\right\} = \frac{1}{T^2}\int_{-T/2}^{T/2}\int_{-T/2}^{T/2} \mathcal{E}\left\{y_i(t)\, y_i(s)\, y_0(t)\, y_0(s)\right\} dtds. \tag{B-28}$$

But

$$\begin{aligned} \mathcal{E}\left\{y_i(t)\, y_i(s)\, y_0(t)\, y_0(s)\right\} &= \cos^2\theta_0 \mathcal{E}\left\{s^2(t)\, s^2(s)\right\} + \mathcal{E}\left\{s(t)\, s(s)\right\}\mathcal{E}\left\{n_i(t)\, n_i(s)\right\} \\ &\quad + \cos^2\theta_0 \mathcal{E}\left\{s(t)\, s(s)\right\}\mathcal{E}\left\{n_0(t)\, n_0(s)\right\} \\ &\quad + \mathcal{E}\left\{n_0(t)\, n_0(s)\right\}\mathcal{E}\left\{n_i(t)\, n_i(s)\right\} \\ &= \left[R_s^2(0) + 2R_s^2(t-s)\right]\cos^2\theta_0 \\ &\quad + R_s(t-s)\, R_{n_i}(t-s) \\ &\quad + R_s(t-s)\, R_{n_0}(t-s)\cos^2\theta_0 \\ &\quad + R_{n_0}(t-s)\, R_{n_i}(t-s), \end{aligned} \tag{B-30}$$

where $R_s(\tau)$ and $R_{n_i}(\tau)$ are the signal and noise autocorrelation functions under a wide-sense stationary assumption.

Examine

$$\begin{aligned} J &= \frac{1}{T^2}\int_{-T/2}^{T/2}\int_{-T/2}^{T/2} R_1(t-s)\, R_2(t-s)\, dtds \\ &= \frac{1}{T^2}\int_{-T/2}^{T/2}\int_{-T/2-s}^{T/2-s} R_1(u)\, R_2(u)\, duds. \end{aligned} \tag{B-31}$$

Interchanging the order of integration, it follows that

$$\begin{aligned} J &= \frac{1}{T^2}\int_0^T \int_{-T/2}^{T/2-u} R_1(u) R_2(u)\, du + \frac{1}{T^2}\int_{-T}^0 \int_{-T/2-u}^{T/2} R_1(u) R_2(u)\, du \\ &= \frac{1}{T^2}\int_{-T}^{T} (T-|u|) R_1(u) R_2(u)\, du \\ &= \frac{1}{T}\int_{-T}^{T} \left(1-\frac{|u|}{T}\right) R_1(u) R_2(u)\, du. \end{aligned} \tag{B-32}$$

If the correlation time $\tau_i \ll T$, implying that $BT \gg 1$, where B is the bandwidth, then

$$\begin{aligned} J &= \frac{1}{T}\int_{-\infty}^{\infty} R_1(u) R_2(u)\, du \\ &= \frac{1}{T}\int_{-\infty}^{\infty} \mathcal{S}_1(f) \mathcal{S}_2(f)\, df, \end{aligned} \tag{B-33}$$

where $\mathcal{S}_1(f)$ and $\mathcal{S}_2(f)$ are the corresponding spectra.

Using the above results, we conclude that

$$\begin{aligned} \mathcal{E}\left\{C_1^2\right\} &= R_s^2(0)\cos^2\theta_0 + \frac{2}{T}\cos^2\theta_0 \int_{-\infty}^{\infty} \mathcal{S}_s^2(f)\, df + \frac{1}{T}\cos^2\theta_0 \int_{-\infty}^{\infty} \mathcal{S}_s(f)\mathcal{S}_{n_0}(f)\, df \\ &\quad + \frac{1}{T}\int_{-\infty}^{\infty} \mathcal{S}_{n_1}(f)\mathcal{S}_s(f)\, df + \frac{1}{T}\int_{-\infty}^{\infty} \mathcal{S}_{n_0}(f)\mathcal{S}_{n_1}(f)\, df. \end{aligned} \tag{B-34}$$

Thus

$$\sigma_1^2 = \frac{1}{T}\int_{-\infty}^{\infty} \left\{2\mathcal{S}_s^2(f)\cos^2\theta_0 + \mathcal{S}_s(f)\mathcal{S}_{n_0}(f)\cos^2\theta_0 + \mathcal{S}_s(f)\mathcal{S}_{n_1}(f) + \mathcal{S}_{n_0}(f)\mathcal{S}_{n_1}\right\} df. \tag{B-35}$$

Similarly

$$\sigma_2^2 = \frac{1}{T}\int_{-\infty}^{\infty} \left\{2\mathcal{S}_s^2(f)\sin^2\theta_0 + \mathcal{S}_s(f)\mathcal{S}_{n_0}(f)\sin^2\theta_0 + \mathcal{S}_s(f)\mathcal{S}_{n_2}(f) + \mathcal{S}_{n_0}(f)\mathcal{S}_{n2}\right\} df \tag{B-36}$$

In addition, it follows that

$$\begin{aligned} \mathcal{E}\left\{C_1 C_2\right\} &= \mathcal{E}\left\{y_1(t)\, y_0(t)\, y_2(t)\, y_0(t)\right\} \\ &= \left[R_s^2(0) + 2R_s^2(t-s) + R_s(t-s) R_{n_0}(t-s)\right]\cos\theta_0 \sin\theta_0. \end{aligned} \tag{B-37}$$

Thus

$$\mathcal{E}\left\{(C_1-\eta_1)(C_2-\eta_2)\right\} = \frac{1}{T}\left\{\int_{-\infty}^{\infty}\left[2\mathcal{S}_s^2(f) + \mathcal{S}_s(f)\mathcal{S}_{n_0}(f)\right] df\right\}\cos\theta_0\sin\theta_0, \tag{B-38}$$

and now we can identify

$$\begin{aligned} r &= \frac{\mathcal{E}\left\{(C_1-\eta_1)(C_2-\eta_2)\right\}}{\sigma_1\sigma_2} \\ &= \frac{1}{T\sigma_1\sigma_2}\left\{\int_{-\infty}^{\infty}\left[2\mathcal{S}_s^2(f) + \mathcal{S}_s(f)\mathcal{S}_{n_0}(f)\right] df\right\}\cos\theta_0\sin\theta_0. \end{aligned} \tag{B-39}$$

Note that provided $BT \gg 1$, both C_1 and C_2 are approximately jointly Gaussian random variables.

For computational purposes, we make some simplifying assumptions. Let

$$\mathcal{S}_{n_0}(f) = \frac{\sigma_N^2}{2B}, \tag{B-40a}$$
$$\mathcal{S}_{n_i}(f) = \rho \mathcal{S}_{n_0}(f), \tag{B-40b}$$
$$\mathcal{S}_s(f) = \frac{\sigma_S^2}{2B} \tag{B-40c}$$

which assumes that the noise has a bandlimited white spectrum. Note that this can be modified for any given bandlimited spectra, requiring integration over the frequency of the band. ρ is the dipole channel noise gain, which is 1/2 or 1/3 for 2-D or 3-D isotropic noise, respectively [27]. Now we obtain

$$\eta_1 = \sigma_S^2 \cos\theta_0, \tag{B-41a}$$
$$\eta_2 = \sigma_S^2 \sin\theta_0, \tag{B-41b}$$
$$\begin{aligned}\sigma_1^2 &= \frac{2B}{T}\left[\frac{2\sigma_S^4}{4B^2} + \frac{\sigma_S^2\sigma_N^2}{4B^2}\right]\cos^2\theta_0 + \rho\frac{2B}{T}\left[\frac{\sigma_S^2\sigma_N^2}{4B^2} + \frac{2\sigma_N^4}{4B^2}\right] \\ &= \frac{\sigma_N^4}{4BT}\left[\mathrm{SNR}\,(2\mathrm{SNR}+1)\cos^2\theta_0 + \rho\,(\mathrm{SNR}+1)\right],\end{aligned} \tag{B-41c}$$
$$\sigma_2^2 = \frac{\sigma_N^4}{4BT}\left[\mathrm{SNR}\,(2\mathrm{SNR}+1)\sin^2\theta_0 + \rho\,(\mathrm{SNR}+1)\right], \tag{B-41d}$$

where $\mathrm{SNR} = \sigma_S^2/\sigma_N^2$. In addition,

$$\begin{aligned}\mathcal{E}\left\{(C_1-\eta_1)(C_2-\eta_2)\right\} &= \frac{2B}{T}\left[\frac{2\sigma_S^4}{4B^2} + \frac{\sigma_S^2\sigma_N^2}{4B^2}\right]\cos\theta_0\sin\theta_0 \\ &= \frac{\sigma_N^4}{4BT}\mathrm{SNR}\,(2\mathrm{SNR}+1)\cos\theta_0\sin\theta_0,\end{aligned}$$

and, thus

$$r = \frac{\mathrm{SNR}(2\mathrm{SNR}+1)\cos\theta_0\sin\theta_0}{\left[\mathrm{SNR}(2\mathrm{SNR}+1)\cos^2\theta_0+\rho(\mathrm{SNR}+1)\right]^{1/2}\left[\mathrm{SNR}(2\mathrm{SNR}+1)\sin^2\theta_0+\rho(\mathrm{SNR}+1)\right]^{1/2}}. \tag{B-42}$$

Given the form of Eq's. B-17 - B-20, we can normalize $\eta_1, \eta_2, \sigma_1, \sigma_2$ by $\sigma_N^2/2BT$ such that Eq. B-41d becomes

$$\widetilde{\eta}_1 = \sqrt{2BT}\mathrm{SNR}\cos\theta_0, \tag{B-43a}$$
$$\widetilde{\eta}_2 = \sqrt{2BT}\mathrm{SNR}\sin\theta_0, \tag{B-43b}$$
$$\widetilde{\sigma}_1^2 = \mathrm{SNR}\,(2\mathrm{SNR}+1)\cos^2\theta_0 + \rho\,(\mathrm{SNR}+1), \tag{B-43c}$$
$$\widetilde{\sigma}_2^2 = \mathrm{SNR}\,(2\mathrm{SNR}+1)\sin^2\theta_0 + \rho\,(\mathrm{SNR}+1). \tag{B-43d}$$

To summarize, for $z = \tan\theta$:

$$p_\theta\left(\theta|\theta_0\right) = \left(1+z\right) p_z\left(z|\theta_0\right), \tag{B-44}$$

$$\begin{aligned} p_z\left(z|\theta_0\right) &= \frac{\widetilde{\sigma}_1\widetilde{\sigma}_2}{\widetilde{\sigma}_2^2 - 2r\widetilde{\sigma}_1\widetilde{\sigma}_2 z + \widetilde{\sigma}_1^2 z^2}\left\{\frac{\left(1-r^2\right)^{1/2}}{2\pi} e^{-\alpha\left(\widetilde{\sigma}_1,\widetilde{\sigma}_2,\widetilde{\eta}_1,\widetilde{\eta}_2,r\right)}\right. \\ &+ \frac{\delta\left(z;\widetilde{\sigma}_1,\widetilde{\sigma}_2,\widetilde{\eta}_1,\widetilde{\eta}_2,r\right)}{\sqrt{2\pi}} \beta\left(z;\widetilde{\sigma}_1,\widetilde{\sigma}_2,\widetilde{\eta}_1,\widetilde{\eta}_2,r\right)^{1/2} e^{-\gamma\left(z;\widetilde{\sigma}_1,\widetilde{\sigma}_2,\widetilde{\eta}_1,\widetilde{\eta}_2,r\right)} \\ &\times \left[\Phi\left(-\delta\left(z;\widetilde{\sigma}_1,\widetilde{\sigma}_2,\widetilde{\eta}_1,\widetilde{\eta}_2,r\right)\frac{\beta\left(z;\widetilde{\sigma}_1,\widetilde{\sigma}_2,\widetilde{\eta}_1,\widetilde{\eta}_2,r\right)^{1/2}}{\left(1-r^2\right)}\right)\right. \\ &\left.\left. - \Phi\left(\delta\left(z;\widetilde{\sigma}_1,\widetilde{\sigma}_2,\widetilde{\eta}_1,\widetilde{\eta}_2,r\right)\frac{\beta\left(z;\widetilde{\sigma}_1,\widetilde{\sigma}_2,\widetilde{\eta}_1,\widetilde{\eta}_2,r\right)^{1/2}}{\left(1-r^2\right)}\right)\right]\right\}, \end{aligned} \tag{B-45}$$

with

$$\alpha\left(\widetilde{\sigma}_1,\widetilde{\sigma}_2,\widetilde{\eta}_1,\widetilde{\eta}_2,r\right) \triangleq \frac{\widetilde{\sigma}_2^2\widetilde{\eta}_1^2 - 2r\widetilde{\sigma}_1\widetilde{\sigma}_2\widetilde{\eta}_1\widetilde{\eta}_2 + \widetilde{\sigma}_1^2\widetilde{\eta}_2^2}{2\widetilde{\sigma}_1^2\widetilde{\sigma}_2^2\left(1-r^2\right)}, \tag{B-46a}$$

$$\beta\left(z;\widetilde{\sigma}_1,\widetilde{\sigma}_2,\widetilde{\eta}_1,\widetilde{\eta}_2,r\right) \triangleq \frac{\left[\left(\widetilde{\sigma}_2^2\widetilde{\eta}_1 - r\widetilde{\sigma}_1\widetilde{\sigma}_2\widetilde{\eta}_2\right) + \left(\widetilde{\sigma}_1^2\widetilde{\eta}_2 - r\widetilde{\sigma}_1\widetilde{\sigma}_2\widetilde{\eta}_1\right)z\right]^2}{\widetilde{\sigma}_1^2\widetilde{\sigma}_2^2\left(\widetilde{\sigma}_2^2 - 2r\widetilde{\sigma}_1\widetilde{\sigma}_2 z + \widetilde{\sigma}_1^2 z^2\right)}, \tag{B-46b}$$

$$\gamma\left(z;\widetilde{\sigma}_1,\widetilde{\sigma}_2,\widetilde{\eta}_1,\widetilde{\eta}_2,r\right) \triangleq \frac{\left(\widetilde{\eta}_2 - \widetilde{\eta}_1 z\right)^2}{2\left(\widetilde{\sigma}_2^2 - 2r\widetilde{\sigma}_1\widetilde{\sigma}_2 z + \widetilde{\sigma}_1^2 z^2\right)}, \tag{B-46c}$$

$$\delta\left(z;\widetilde{\sigma}_1,\widetilde{\sigma}_2,\widetilde{\eta}_1,\widetilde{\eta}_2,r\right) = \operatorname{sign}\left[\left(\widetilde{\sigma}_2^2\widetilde{\eta}_1 - r\widetilde{\sigma}_1\widetilde{\sigma}_2\widetilde{\eta}_2\right) + \left(\widetilde{\sigma}_1^2\widetilde{\eta}_2 - r\widetilde{\sigma}_1\widetilde{\sigma}_2\widetilde{\eta}_1\right)z\right], \tag{B-46c}$$

and

$$\Phi\left(x\right) \triangleq \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-u^2/2}du. \tag{B-47}$$

A plot of $p_\theta\left(\theta|\theta_0\right)$ as a function of SNR and bearing for a target vehicle at 20^o is shown in Figure B1.

References

[1] A. Gelb, Ed. *Applied Optimal Estimation*, The MIT Press, Cambridge, MS, 1974.

[2] A. H. Jazwinski, *Stochastic Process and Filtering Theory*, Academic Press, San Diego, CA, 1970.

[3] M. S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," IEEE Trans. on Signal Processing, **50**(2), February 2002, pp. 174-188.

[4] S. J. Julier and J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation," Proceedings of the IEEE, **92**(3), March 2004, pp. 401-422.

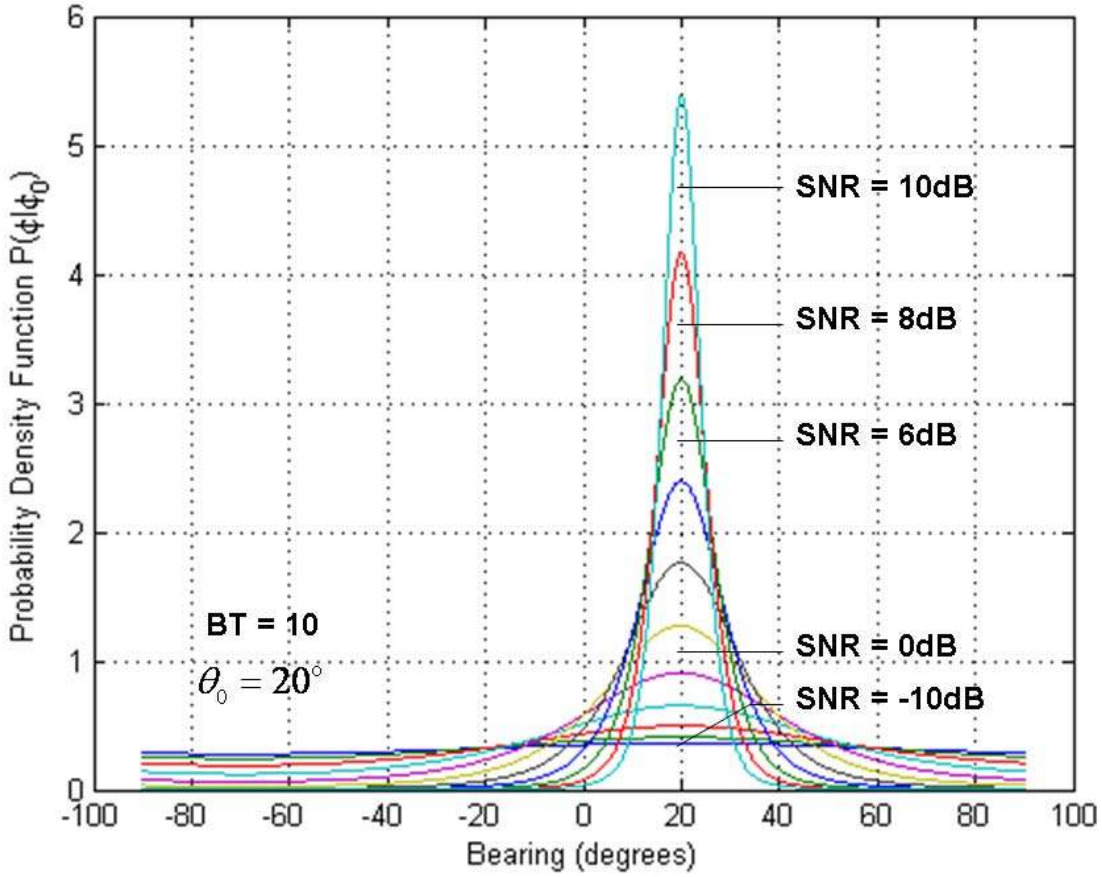

Figure B.1: Likelihood Function for a Signal at 20 Degrees for a Variety of SNRs.

[5] H. J. Kushner, "Approximations to Optimal Nonlinear Filters," IEEE Trans. on Automatic Control, AC-12, No. 5, October 1967, pp. 546-556.

[6] K. Ito and K. Xiong, "Gaussian Filters for Nonlinear Filtering Problems," IEEE Trans. Automatic Control, **45**(5), May 2000, pp. 910-927.

[7] E. Bolviken and G. Storvik, "Deterministic and Stochastic Particle Filters in State-Space Models," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. de Freitas, and N. J. Gordon, Eds. New York, Springer-Verlag, 2001.

[8] A. Honkela, "Approximating Nonlinear Transformations of Probability Distributions for Nonlinear Independent Component Analysis," IEEE Int. Conf. on Neural Nets (IJCNN 2004), Proceedings of, Budapest, Hungary, pp. 2169-2174, 2004.

[9] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, "An Introduction to Sequential Monte Carlo Methods," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. de Freitas, and N. J. Gordon, Eds. New York, Springer-Verlag, 2001.

[10] N. Gordon, D. Salmond, and A. F. M. Smith, "Novel Approach to Nonlinear and Non-Gaussian State Estimation, Proc. Inst. Elect. Eng., F, Vol. 140, pp. 107-113, 1993.

[11] S. Maskell and N. Gordon, "A Tutorial on Particle Filters for On-line Nonlinear/Non-Gaussian Bayesian Tracking," Target Tracking: Algorithms and Applications, IEE Workshop on, pp. 2-1 to 2-15, 16 October 2001.

[12] J. H. Kotecha and P. M Djurić, "Gaussian Particle Filtering," IEEE Trans. on Signal Processing, **51**(10), October 2003, pp. 2592-2601.

[13] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, "The Unscented Particle Filter," Technical Report CUED/F-INTENG/TR 380, Cambridge University Engineering Department, August 2000.

[14] K.Murphy and S. Russell, "Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. de Freitas, and N. J. Gordon, Eds., Springer-Verlag, New York, NY 2001.

[15] R. E. Zarnich, K. L. Bell, and H. L. Van Trees, "A Unified Method for Measurement and Tracking of Contacts from an Array of Sensors," IEEE Trans. Signal Processing, **49**(12), December 2001, pp. 2950-2961.

[16] M. Orton and W. Fitzgerald, "A Bayesian Approach to Tracking Multiple Targets Using Sensor Arrays and Particle Filters," IEEE Trans. Signal Processing, **50**(2), February 2002, pp. 216-223.

[17] S. C. Nardone, A. G. Lindgren, and K. F. Gong, "Fundamental Properties and Performance of Conventional Bearings-Only Target Motion Analysis," IEEE Trans. Automatic Control, AC-**29**(9), September 1984, pp. 775-787.

[18] R. L. Streit and M. J. Walsh, "A Linear Least Squares Algorithm for Bearing-Only Target Motion Analysis," Aerospace Conf., Proceedings of the 1999 IEEE, March 6-13, 1999, Vol. 4, pp. 6-13.

[19] J. M. Bernardo and A. F. Smith, *Bayesian Theory*, John Wiley & Sons, New York, NY, 1994.

[20] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes in C, The Art of Scientific Computing*, 2nd Edition. Cambridge University Press, New York, NY, 1992, pp. 147-161.

[21] G. H. Golub, "Some Modified Matrix Eigenvalue Problems," SIAM Review, **15**(2), April, 1973, pp. 318-334.

[22] J. S. Ball, "Orthogonal Polynomials, Gaussian Quadratures, and PDEs," Computing in Science and Engineering, November/December, 1999, pp. 92-95.

[23] S. M. Ross, *Introduction to Probability Models*, Fourth Edition, Academic Press, San Diego, CA, 1989.

[24] P. M. Djurić, Y. Huang, and T. Ghirmai, "Perfect Sampling: A Review and Application to Signal Processing," IEEE Trans. on Signal Processing, **50**(2), February 2002, pp. 345-356.

[25] B. D. Ripley, *Stochastic Simulation*, John Wiley & sons, New York, NY, 1987.

[26] M. Briers, S. R. Maskell, and R. Wright, "A Rao-Blackwellised Unscented Kalman Filter," Info. Fusion 2003, Proceedings of the 6th Int. Conf of, Vol 1, July 8-11, 2003.

[27] C. Casella and C. P. Robert, "Rao-Blackwellisation of Sampling Schemes," Biometrica, **83**(1), pp. 81-94, 1996.

[28] C. Hue, J. LeCadre, and P. Perez, "Sequential Monte Carlo Methods for Multiple Target Tracking and Data fusion," IEEE Trans. Signal Processing, **50**(2), pp. 309-325, February 2002.

[29] S. W. Davies, "Bearing Accuracies for Arctan Processing of Crossed Dipole Arrays," OCEANS 1987, 19 September 1987, pp. 351-356.

[30] H. Cox and R. M. Zeskind, "Adaptive Cardioid Processing," Signals, Systems and Computers, 26th Asilomar Conference on, 26-28 October 1992, pp. 1058-1062, Vol. 2.

[31] B. H. Maranda, "The Statistical Accuracy of an Arctangent Bearing Estimator," OCEANS 2003, Proceedings, 22-26 September 2003, pp. 2127-2132, Vol. 4.

[32] A. D. Mars, "Asynchronous Multi-Sensor Tracking in Clutter with Uncertain Sensor Locations using Bayesian Sequential Monte Carlo Methods," Aerospace Conference, 2001, IEEE Proceedings, Vol. 5, 10-17 March 2001, pp. 2171-2178.

[33] "Special Issue on Monte Carlo Methods for Statistical Signal Processing," IEEE Transactions on Signal Processing, **50** (2), February 2002.

[34] "Special Issue on Sequential State Estimation," Proceedings of the IEEE, **92** (3) March 2004.

[35] P. Fearnhead, "Sequential Monte Carlo Methods in Filter Theory," Mercer College, University of Oxford, 1998.

[36] R. Karlsson, "Simulation Based Methods for Target Tracking," Division of Automatic Control & Communication Systems, Department of Electrical Engineering, Linköpings universitet, Linköping, Sweden, 2002.

[37] S. M. Herman, "A Particle Filtering Approach to Joint Passive Radar Tracking and Target Classification," University of Illinois at Urbana-Champaign, 2002,

[38] T. Schön, "On Computational Methods for Nonlinear Estimation," Division of Automatic Control & Communication Systems, Department of Electrical Engineering, Linköpings universitet, Linköping, Sweden, 2003.

[39] C. Andrieu, M. Davy, and A. Doucet, "Efficient particle Filtering for Jump Markov Systems. Application to Time-Varying Autoregressions," IEEE Trans. Signal Processing, **51**(7), July 2003, pp. 1762-1770.

[40] C. Andrieu and A. Doucet, "Joint Bayesian Model Selection and Estimation of Noisy Sinusoids via Reversible Jump MCMC," IEEE Trans. Signal Processing, **47**(10), October 1999, pp. 2667-2676.

[41] J. R. Larocque and J. P Reilly, "Reversible Jump MCMC for Joint Detection and Estimation of Sources in Colored Noise," IEEE Trans. Signal Processing, **50**(2), February 2002, pp. 231-240.

[42] J. J. Rajan and P. J. W. Rayner, "Parameter Estimation of Time-Varying Autoregressive Models using the Gibbs Sampler," Electronic Letters, **31**(13), June 1995, pp. 1035-1036.

[43] J. J. Rajan and A. Kawana, "Bayesian Model Order Selection using the Gibbs Sampler," Electronic Letters, **32**(13), June 1996, pp. 1156-1157.

[44] D. Sornette and K. Ide, "The Kalman-Levy Filter," Physica D, vol. 151, pp. 142-174, 2001.

[45] N. Gordon, J. Percival, and M. Robinson, "The Kalman-Levy Filter and Heavy-Tailed Models for Tracking Maneuvering Targets," Proc. 6^{th} Int. Conf. on Information Fusion, pp. 1024-1031, 7-10 July, 2003.

Distribution List

Anteon

A. Shah

ARL/UT

R. Graman

ASTO

R. Zarnich

BA & H

T. Oliver

GD-AIS

M. Cho

JHU/APL

J. Stapleton

Metron

L. Stone

NUWC

E. Giannopoulos
T. Luginbuhl
S. Grenadier

PMS 401

H. Megonigal

W400

F. Driscoll
A. Haug
G. Jacyna
S. Polk
V. Wrick

W407

C. Burmaster
C. Christou
K. McAdow
J. Messerschmidt

W901

R. Bethel
D. Clites
D. Colella
J. Creekmore
S. Pawlukiewicz