# The Antiparticle Filter - an Adaptive Nonlinear Estimator

John Folkesson

Abstract We introduce the antiparticle filter, AF, a new type of recursive Bayesian estimator that is unlike either the extended Kalman Filter, EKF, unscented Kalman Filter, UKF or the particle filter PF. We show that for a classic problem of robot localization the AF can substantially outperform these other filters in some situations. The AF estimates the posterior distribution as an auxiliary variable Gaussian which gives an analytic formula using no random samples. It adaptively changes the complexity of the posterior distribution as the uncertainty changes. It is equivalent to the EKF when the uncertainty is low while being able to represent non-Gaussian distributions as the uncertainty increases. The computation time can be much faster than a particle filter for the same accuracy. We have simulated comparisons of two types of AF to the EKF, the iterative EKF, the UKF, an iterative UKF, and the PF demonstrating that AF can reduce the error to a consistent accurate value.

#### 1 Introduction

Non-linear estimation is crucial to robotics as well as many other fields. Fundamentally one needs to describe uncertainty in some state based on indirect and noisy observations of it. This state is known to evolve in time and that evolution can introduce uncertainty. When both the observation and the evolution are described by linear system equations wrt the state estimate and the noise is white Gaussian then the Kalman filter is the optimal estimator, [1]. The problem becomes harder when the system is non-linear. One approach is

Centre for Autonomous Systems, Royal Institute of Technology (KTH), Stockholm, Sweden e-mail: johnf@csc.kth.se

John Folkesson

to linearize the system equations wrt the state around the current estimated state as in the extended Kalman Filter EKF [2, 3].

The EKF has led to the development of many variations. These mostly address the issue of inconsistency. For the EKF inconsistency arises as a consequence of the linearization. The EKF is a Gaussian estimator which implies that the posterior distribution is parametrized by a state mean and covariance. When the covariance is too small we say that the estimate is overconfident [4]. This can lead to divergence of the estimate or to bad inferences based on it. In order improve filter stability methods of better estimating or simple adjusting the covariance have been tried, such as, the robust extended Kalman filter [5, 6], or the method used for robot localization in [7].

A better estimate of the covariance is obtained by computing the estimate based on mapping a selected set of points through the non-linearity, allowing the estimate to reflect more than a single point. This approach led to the unscented Kalman filter, UKF [8, 9, 10], linear regression Kalman filters LRKF [11, 12, 13], the shifted Rayleigh filter [14], and the filter in [15].

Divergence or unstable behavior often occurs while incorporating observations that require large changes to the estimated mean. The iterative Kalman filter, IEKF [16, 17] can help this problem by applying the EKF formula and linearizations repeatedly until convergence to a stable solution.

All of the above estimators use a Gaussian posterior. As the uncertainty in the estimates becomes more significant wrt the non-linearity, the Gaussian distribution can no longer adequately represent the posterior. This situations can be addressed by the particle filter PF, [18, 19, 20, 21]. The PF can estimate any distribution and makes no requirements on the form of the noise. It does so by sampling the distribution. It essentially carries out many simulations, particles, of possible evolutions of the system, re-weighting the particle set based on the observations. The PF is very popular and powerful. It has one drawback: the estimate depends on having many particles near the true state. It is important to keep the particles spread evenly and densely. Unfortunately so called particle deprivation (or depletion) is inevitable (eventually) using the PF. Particle deprivation is the condition of having too few different particles. This can be made less significant by increasing the number of particles but this leads to higher computational costs.

Particle deprivation is even more of a problem in higher dimensions. This can sometimes be solved by Rao-Blackwellized particles filters, such as, Fast-SLAM [22, 23]. Alternatively, a novel Gaussian particle filter is proposed in [24] which does not require re-sampling but does assume a Gaussian posterior.

Other solutions to representing non-Gaussian distributions are the sum of Gaussians as used in [25] on the simultaneous localization and mapping, problem. Or the Gaussian sum quadrature filter as presented in [13]. Exact batch methods and some of the graphical algorithms [26, 27, 28, 29, 30] can solve the exact system by successive approximation. In [31] a method that combines graphical belief propagation, particle filters and uses auxiliary variables is presented.

#### 2 Overview of this article

We will developed a new class of estimators which we call *antiparticle filters* AF [32]. We shall describe two members of that class which we call the quadratic and the trigonometric AF, QAF, and TAF. We compare these filters to the EKF, IEKF, UKF, IUKF<sup>1</sup> and PF.

We start by discussing the AF posterior distribution. We show how it can change complexity adaptively as the uncertainty changes by introducing and removing auxiliary variables. We then discuss how to estimate this distribution with prediction and update steps.

This is then followed by our simulations which show how for a robot localization problem with large uncertainty the AF is more accurate, stable and consistent than the other types of filters. The PF can be made more accurate and consistent by increasing the number of particles. We show the PF for a number of particles that gives similar computation times as our filter and one that is significantly slower but still not as consistent or accurate.

## 3 The auxiliary variable Gaussian distribution

As state estimates evolve in time the uncertainty increases during the prediction step which moves the estimate forward in time using the dynamic equations. When observations are made the uncertainty is reduced in an update step. If no observations are made for a long time the uncertainty can become too large, making the nonlinearities significant. This uncertainty is often highly correlated. So that even if all the covariance matrix elements are growing this growth is really only in one (or few) problem directions. We propose to factor out part of this problem direction and model its uncertainty in a  $n^{\lambda}$  dimensional auxiliary variable vector called  $\lambda$ . So that the posterior distribution takes the form:

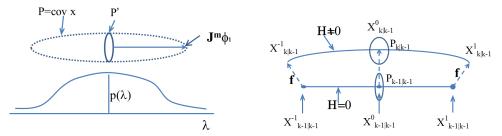
$$p(\mathbf{x}) = \int_{-\infty}^{\infty} G(\mathbf{x} - \mathbf{m}^{\lambda}, P) \ G(\lambda, C) (d\lambda)^{n_{\lambda}}$$
 (1)

where  $G(\lambda, C)$  is the zero mean Gaussian distribution with covariance C. We see that the mean of the state estimate  $\mathbf{m}^{\lambda}$  is conditionally parametrized by  $\lambda$ . This  $\mathbf{m}^{\lambda}$  is a differentiable function of  $\lambda$ . By assuming different forms for this we can generate different shapes for the distribution and different filter types. We will later use this fact:

$$E[\lambda \lambda^T] - E[\lambda]E[\lambda^T] = C. \tag{2}$$

We can derive some results by using the Taylor series expansion of  $\mathbf{m}^{\lambda}$ :

<sup>&</sup>lt;sup>1</sup> The iterative UKF is an iterative version of the UKF similar to the IEKF.



**Fig. 1** The left illustrates that the **x** covariance can be exactly moved between P and  $(P', \mathbf{J}^m, C)$ . The right shows the subsequent predict phase.

$$\mathbf{m}^{\lambda} = \mu + \mathbf{J}^{m}\lambda + \frac{1}{2}\lambda^{T}\mathbf{H}^{m}\lambda + O(\lambda^{3})$$
(3)

where  $\mu = \mathbf{m}^0$ ,  $\mathbf{H}^m$  is an  $\mathbf{x}$  space vector of symmetric  $\lambda$  space Hessian matrices and the Jacobian matrix  $\mathbf{J}^m$  is a matrix from  $\lambda$  to  $\mathbf{x}$  space.

$$E[\mathbf{x}] = \int_{-\infty}^{\infty} \mathbf{m}^{\lambda} G(\lambda, C) (d\lambda)_{\lambda}^{n} \approx \mu + \frac{1}{2} \sum_{i,j} \mathbf{H}_{ij}^{m} C_{ij}$$
$$E[\mathbf{x}\mathbf{x}^{T}] - E[\mathbf{x}] E[\mathbf{x}^{T}] \approx P + \mathbf{J}^{m} C(\mathbf{J}^{m})^{T} + \frac{1}{2} \sum_{i,j,k,l} \mathbf{H}_{ij}^{m} C_{ik} C_{jl} (\mathbf{H}_{kl}^{m})^{T}$$

These two equations are used when creating and destroying auxiliary variables with minimal changes to the mean and covariance in  $\mathbf{x}$ . This is done adaptively in response to the changes in uncertainty. When the uncertainty is low the distribution can be simplified by reducing the number of auxiliary dimensions and vice versa as shown schematically on the left in fig. (1).

## Destroying Auxiliary Dimensions

If we chose a basis in the  $\lambda$  space so that C = I then we can eliminate dimension q from  $\lambda$  by making these adjustments to the remaining parameters:

$$\mu \leftarrow \mu + \frac{1}{2} \mathbf{H}_{qq}^{m}, \quad P_{ij} \leftarrow P_{ij} + \Delta P_{ij} = P_{ij} + J_{iq}^{m} J_{jq}^{m} + \sum_{k} H_{iqk}^{m} H_{jqk}^{m} - \frac{1}{2} H_{iqq}^{m} H_{jqq}^{m}$$

In general, when the trace of  $\Delta P$  is below a threshold for some q we will eliminate that dimension by making the adjustments above and setting  $\lambda_q = 0.^2$  This  $\Delta P$  decreases during updates and increases during prediction so we normally do this check after updating with new observations.

 $<sup>^{2}</sup>$  We use the threshold .01 in our simulations.

Creating Auxiliary Dimensions

We can create new auxiliary dimensions by moving some of the uncertainty in unit direction  $\mathbf{u}$ , from P into a new  $\lambda$  dimension, q. For this we must chose the new parameters to meet these constraints with  $C_{qq} = 1$  and  $\mathbf{H}_{qk} = 0$ :

$$\mathbf{J}_q^m = \sigma \sqrt{1-\delta} \mathbf{u}, \quad P = P - \mathbf{u}(1-\delta)\sigma^2 \mathbf{u}^T, \quad P^{-1} = P^{-1} + \frac{P^{-1}\mathbf{u}(1-\delta)\sigma^2(P^{-1}\mathbf{u})^T}{\delta}$$

where  $\sigma^{-2} = \mathbf{u}^T P^{-1} \mathbf{u}$  and  $\delta$  is a small number to keep P positive definite. In practice we use the largest eigenvalue of P for  $\sigma^2$  and the eigen vector for  $\mathbf{u}$ . If the eigenvalue is above some threshold<sup>3</sup> then we create a new dimension q.

Parameterizing the distribution - the antiparticles

We will assume that the vector functions  $\mathbf{m}^{\lambda}$  can be parametrized in two ways. One is by some canonical parametrization,  $\pi$  which can be used to directly compute  $\mathbf{m}^{\lambda}$  for any lambda. The second is the values of  $\mathbf{m}^{\lambda}$  at a set of  $\lambda = \varphi_i$  from which the canonical parameters can be derived. This set of values with be notated by  $\mathbf{x}^i = m^{\varphi_i}$ . The  $\varphi_i$  are chosen distributed around the mean  $\lambda = 0$  value using the C in a manner reminiscent of how the regression (sigma) points are chosen in the LRKF (UKF).

We will call the set  $\{\mathbf{x}^i\}$  the antiparticles as they will function similarly to particles in the PF but are not random samples. We assume that we can go from the canonical parametrization to the antiparticles and back as required.

$$\{\mathbf{x}^i\} \leftrightarrow \{\pi\}$$
 (4)

The antiparticles will be sufficient for doing prediction and are the result of the update. We will find it necessary to use the canonical parameters for updates. They are also used when creating and destroying auxiliary dimensions.

#### 4 Estimation of the non-linear process

The starting point of all estimators is the process model:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \omega_k, \qquad \omega_k \sim G(0, Q_k)$$
 (5)

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \nu_k, \qquad \qquad \nu_k \sim G(0, R_k) \tag{6}$$

 $<sup>^3</sup>$  We use 1.0 for this threshold.

Here  $\mathbf{x}_k$  is the state,  $\mathbf{z}_k$  are the measurements,  $\mathbf{u}_k$  are the control signals, and Q/R are the white noise covariances. We will use subscripts k-1|k-1 and k|k-1 for the parameters before and after the  $k^{th}$  prediction respectively.

Prediction

Prediction is done by marginalizing over  $\mathbf{x}_{k-1}$ :

$$\int p(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{u}_k)p_{k-1|k-1}(\mathbf{x}_{k-1})d\mathbf{x}_{k-1} = p_{k|k-1}(\mathbf{x}_k)$$
(7)

For the above process and our assumed form for the distribution this becomes

$$\int G(\mathbf{x}_k - \mathbf{m}_{k|k-1}^{\lambda}, P_{k|k-1}) G(\lambda, C_{k|k-1}) (d\lambda)^{n_{\lambda}} \approx \infty$$

$$\int \int G(\mathbf{x}_k - f(\mathbf{x}_{k-1}, u_k), Q_k) G(\mathbf{x}_{k-1} - \mathbf{m}_{k-1|k-1}^{\lambda}, P_{k-1|k-1}) G(\lambda, C_{k-1|k-1}) d\mathbf{x}_{k-1} (d\lambda)^{n_{\lambda}}$$

We linearize around an estimated state  $\mathbf{x}_k = \mathbf{m}_{k-1|k-1}^{\lambda}$  just as in the EKF.

$$\begin{aligned} \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) &\approx \bar{\mathbf{x}}^{\lambda} + \mathbf{J}_{\lambda}^f(\mathbf{x}_{k-1} - \mathbf{m}_{k-1|k-1}^{\lambda}), & \mathbf{J}_{\lambda}^f &\equiv \frac{\partial \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k)}{\partial \mathbf{x}_{k-1}} \mid_{\mathbf{x}_{k-1} = \mathbf{m}_{k-1|k-1}^{\lambda}} \\ \bar{\mathbf{x}}^{\lambda} &\equiv \mathbf{f}(m_{k-1|k-1}^{\lambda}, \mathbf{u}_k), & P^{\lambda} &\equiv Q_k + \mathbf{J}_{\lambda}^f P_{k-1|k-1} (\mathbf{J}_{\lambda}^f)^T. \end{aligned}$$

Using these substitutions we can do the marginalization above and find:

$$\int G(\mathbf{x}_k - \mathbf{m}_{k|k-1}^{\lambda}, P_{k|k-1}) G(\lambda, C_{k|k-1}) (d\lambda)^{n_{\lambda}} \approx \int \sqrt{|P^{\lambda}|} G(\mathbf{x}_k - \bar{\mathbf{x}}^{\lambda}, P^{\lambda}) G(\lambda, C_{k-1|k-1}) (d\lambda)^{n_{\lambda}}$$

In light of this equation we make the following prediction rules:

$$P_{k|k-1} = P^{\lambda} \mid_{\lambda=0}, \quad C_{k|k-1} = C_{k-1|k-1},$$
 (8)

$$\mathbf{m}_{k|k-1}^{\lambda} = \bar{\mathbf{x}}^{\lambda} \Rightarrow \{\mathbf{x}_{k|k-1}^{i}\} = \{\mathbf{f}(\mathbf{x}_{k-1|k-1}^{i})\}$$
(9)

The approximations are that  $P^{\lambda}$  does not vary with  $\lambda$  and that by mapping the antiparticles through the nonlinear  $\mathbf{f}$ , the  $\mathbf{m}_{k|k-1}^{\lambda}$  will follow  $\bar{\mathbf{x}}^{\lambda}$  for all  $\lambda$ .

We end up with a simple predict formula that uses the EKF prediction for P, leaves C unchanged and maps the antiparticles through the dynamics in the same way that the mean is treated by the EKF, illustrated in fig. (1). If no update is to be done we can feed the antiparticles directly into the next predict step without converting to the canonical form.

Update

During update the observation measurements are used to form the posterior distribution. The most important part of the update is locating the maximum likelihood estimate, MLE, for both  $\mathbf{x}_k$  and  $\lambda$ . The canonical parametrization is used for this. Once this MLE is found a new set of antiparticles is found around this MLE. These then are ready for the next prediction step.

Setting  $\Delta \mathbf{z}(\mathbf{x}_k) = \mathbf{z}_k - \mathbf{h}(\mathbf{x}_k)$  and Bayes rule gives us the posterior:

$$G(\mathbf{x}_k - m_{k|k}^{\lambda}, P_{k|k})G(\lambda, C_{k|k}) \propto G(\mathbf{x}_k - m_{k|k-1}^{\lambda}, P_{k|k-1})G(\Delta \mathbf{z}(\mathbf{x}_k), R)G(\lambda, C_{k|k-1})$$

Taking the exponents on the right hand side above we see

$$(\mathbf{x}_{MLE}, \lambda_{MLE}) = \arg\min_{\mathbf{x}_k, \lambda} g(\mathbf{x}_k, \lambda)$$
 (10)

where g is given by:

$$g(\mathbf{x}_k, \lambda) = [(\mathbf{x}_k - \mathbf{m}_{k|k-1}^{\lambda})^T P_{k|k-1}^{-1} (\mathbf{x}_k - \mathbf{m}_{k|k-1}^{\lambda}) + \Delta \mathbf{z}(\mathbf{x}_k)^T R^{-1} \Delta \mathbf{z}(\mathbf{x}_k) + \lambda^T C_{k|k-1}^{-1} \lambda]/2.$$

We will repeatedly make use of linearization of the function h:

$$\Delta \mathbf{z}(\mathbf{x}_k) = \mathbf{z}_k - \mathbf{h}(\mathbf{x}_k) \approx \mathbf{z}_k - \mathbf{h}(\mathbf{x}) + J^h(\mathbf{x}_k - \mathbf{x})$$
(11)

We do a three phase update which is shown schematically in fig. (2). Phase 1 varies only  $\lambda$  while holding  $\mathbf{x}_k = \mathbf{m}_{k|k-1}^{\lambda}$ . This moves along the  $\mathbf{m}^{\lambda}$  curves to minimize eq. (10). Phase 2 varies both  $\mathbf{x}_k$  and  $\lambda$ . Phase 3 holds  $\lambda = \varphi_i$  while varying  $\mathbf{x}_k$  to find the updated antiparticle set.

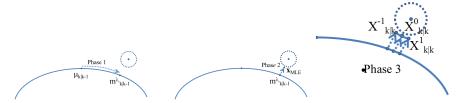


Fig. 2 Here we show the three update phases.

Update phase 1 - Getting near the MLE

We begin by converting the antiparticle set to the canonical parametrization. We seek a point  $\lambda_0$  on the  $\mathbf{m}_{k|k-1}^{\lambda}$  curves to best explain the measurements.

$$\lambda_0 = \arg\min_{\lambda} g(\mathbf{m}_{k|k-1}^{\lambda}, \lambda) \tag{12}$$

We use the Gauss-Newton method starting from  $\lambda_0 = 0$ ,

$$\nabla g = (C_{k|k-1}^{-1}\lambda_0 - (J^h \mathbf{J}_{k|k-1}^m)^T R_k^{-1} \Delta \mathbf{z}) \mid_{\mathbf{x} = \mathbf{m}_{k|k-1}^{\lambda_0}}$$
(13)

$$\Omega \equiv C_{k|k-1}^{-1} + (J^h \mathbf{J}_{k|k-1}^m)^T R_k^{-1} J^h \mathbf{J}_{k|k-1}^m \mid_{\mathbf{x} = \mathbf{m}_{k|k-1}^{\lambda_0}}$$
(14)

$$\Delta \lambda_0 = -\Omega^{-1} \nabla g \tag{15}$$

where  $\mathbf{J}_{k|k-1}^m$  denotes the Jacobian of  $\mathbf{m}_{k|k-1}^{\lambda}$  wrt.  $\lambda$ . We will use this  $\Delta\lambda_0$  to move iteratively towards a solution to eq. (12).

#### Update phase 2 - Finding the MLE

In the second phase we move to the solution to eq. (10) by varying both  $\mathbf{x}_k$  and  $\lambda$ . We use the Gauss-Newton method starting at  $\mathbf{x}_k = \mathbf{m}^{\lambda_0}$  and  $\lambda = \lambda_0$ .

$$w_{\lambda} \equiv (\mathbf{J}_{k|k-1}^{m})^{T} P_{k|k-1}^{-1} (\mathbf{x}_{k} - \mathbf{m}_{k|k-1}^{\lambda}) - C_{k|k-1}^{-1} \lambda$$
 (16)

$$\mathbf{w}_x \equiv (J^h)^T R_k^{-1} \Delta \mathbf{z}(\mathbf{x}_k) - P_{k|k-1}^{-1} (\mathbf{x}_k - \mathbf{m}_{k|k-1}^{\lambda})$$
(17)

$$W_{\lambda\lambda} \equiv (\mathbf{J}_{k|k-1}^m)^T P_{k|k-1}^{-1} \mathbf{J}_{k|k-1}^m + C_{k|k-1}^{-1}$$
(18)

$$W_{x\lambda} \equiv P_{k|k-1}^{-1} \mathbf{J}_{k|k-1}^m \tag{19}$$

$$W_{xx} \equiv (J^h)^T R_k^{-1} J^h + P_{k|k-1}^{-1}$$
 (20)

$$\begin{pmatrix} \Delta \lambda \\ \Delta \mathbf{x}_k \end{pmatrix} = W^{-1} \mathbf{w} = \begin{pmatrix} W_{\lambda \lambda} & W_{x \lambda}^T \\ W_{x \lambda} & W_{x x} \end{pmatrix}^{-1} \begin{pmatrix} w_{\lambda} \\ \mathbf{w}_x \end{pmatrix}$$
(21)

## Update C and P

We set  $P_{k|k} = (W_{xx})^{-1}|_{MLE}$  just as the IEKF. This gives the right shape for the distribution near the MLE. For the C update we note that  $C_{k|k}$  is the covariance of  $\lambda$  for the posterior distribution, eq. (2). By estimating this using the expansion around the MLE we find:

$$C_{k|k} \approx \int_{-\infty}^{\infty} (\lambda - \lambda_{MLE})^2 G((\lambda - \lambda_{MLE}, \mathbf{x} - \mathbf{x}_{MLE}), W^{-1}) (d\mathbf{x})^n (d\lambda)^n_{\lambda}$$
 (22)

$$C_{k|k}^{-1} = \left[ W_{\lambda\lambda} - W_{x\lambda}^T (W_{xx})^{-1} W_{x\lambda} \right] \mid_{\mathbf{x}_k, \lambda = MLE}$$
(23)

$$= C_{k|k-1}^{-1} + (J^h \mathbf{J}^m)^T (J^h P_{k|k-1} (J^h)^T + R)^{-1} J^h \mathbf{J}^m$$
(24)

Update phase 3 - Creating a new antiparticle set

For phase 3 we need to create a new antiparticle set that reflects the measurements. The  $\lambda_{MLE}$  and C are used to produce new values for  $\varphi_i$ . These then produce a new antiparticle set by minimizing g subject to  $\lambda = \varphi_i$ .

$$\mathbf{x}_{k|k}^{i} = \arg\min_{\mathbf{x}_{k}} g(\mathbf{x}_{k}, \varphi_{i}) \tag{25}$$

$$\Delta \mathbf{x}_{k|k}^{i} = (W_{xx})^{-1} \mathbf{w}_{x} \mid_{\mathbf{x}_{k} = \mathbf{x}_{k|k}^{i}, \lambda = \varphi_{i}}$$
(26)

We again use the Gauss-Newton method this time starting from  $\mathbf{x}^i = \mathbf{m}_{k|k-1}^{\varphi_i}$ .

Gauss-Newton Iterations

In each of the three update phases we compute a  $\Delta$  increment. This will in general have the wrong direction and the wrong magnitude although it will be approximately correct. We allow for the direction being wrong by applying the method iteratively stopping when the change in g is not significant. We allow for the magnitude being wrong by doing a line search in the direction of  $\Delta$  and finding the minimum g value along the line. This is done by computing g at N points along the line between 0 and  $2\Delta$ . We then interpolate to the minimum using a quadratic fit to the minimum and the points to either side.<sup>4</sup>

## 5 QAF and TAF

The quadratic antiparticle filter, QAF, uses this parametrization:

$$\mathbf{m}^{\lambda} = \mu + \mathbf{\Lambda}\lambda + \frac{1}{2}\lambda^{T}\mathbf{\Gamma}\lambda \tag{27}$$

Where  $\{\pi\} = \{\mu, \Lambda, \Gamma\}^5$ . We chose C to be diagonal then:

$$\varphi_0 = 0, \quad \varphi_i = (0, ..., \sqrt{C_{ii}}, ..., 0)^T, \quad \varphi_{-i} = -\varphi_i,$$
  
$$\varphi_{i,j} = (\varphi_i + \varphi_j)/\sqrt{2}, \quad i, j \in \{1, 2, ... n_{\lambda}\}, j < i.$$

This leads to a generalization of the linear correlations in the Gaussian distribution to parabolic ones. Parabolas can not model very larger circular nonlinearities. This lead us to try circular parametrizations. The trigonometric antiparticle filter, TAF, uses this parametrization:

 $<sup>^4</sup>$  We used N=21.

<sup>&</sup>lt;sup>5</sup>  $\Lambda$  is a matrix from the  $\lambda \to \mathbf{x}$  spaces and x vector  $\Gamma$  is a symmetric matrix wrt  $\lambda$ .

$$m_{\theta}^{\lambda} = \mu_{\theta} + \sum_{j} \alpha_{\theta j} \lambda_{j} \tag{28}$$

$$m_i^{\lambda} = \mu_i + \rho_i \sin \sum_j \alpha_{ij} \lambda_j + \kappa_i (\cos \sum_j \beta_{ij} \lambda_j - 1), \qquad i \neq \theta$$
 (29)

 $\{\pi\} = \{\mu, \rho, \alpha, \kappa, \beta\}^6$ . For the TAF we use an additional  $2n^{\lambda}$  values  $\varphi_{\pm 2i} = \varphi_{\pm i}/2$ , for  $i = 1, 2, ...n^{\lambda}$ .

## 6 Experiments

We did simulations of a robot moving in 2D.

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \Delta s \\ \Delta \theta \end{pmatrix}$$
(30)

With  $Q_k = J_u Q J_u^T + 10^{-10} I$  where  $J_u$  is the Jacobian of **f** wrt **u** and Q is a 2x2 diagonal matrix. The observations were range and bearing to point features with known data associations. Our UKF and PF<sup>7</sup> implementations were as in [33] and the IUKF was an iterative version.

For the first experiment we did 800 simulations for each of 3 values of Q, small, medium and large.<sup>8</sup>. The robot started at the center of a circle of features and moved with odometery reading steady increments  $(0.2, 0, 0)^T$  and the true path having noise added to that according to Q. The experiment was designed to have the robot first see one feature giving partial pose information and then later a second feature giving the exact pose. The consistency and accuracy was analyzed at the point before the first update, at the update, before the second feature update, at the second feature update, and then at selected times after the second feature was observed.

An example of one run is shown in fig. (3). One can see how the AF is able to approximately model the crescent shaped distribution. Fig. (4) shows the root mean square xy error as it evolved after the first update. We see that the QAF and the TAF converged more rapidly than the other methods. Notice that the PF tends to gets stuck at a fixed error depending on the density of particles while the other methods improve the pose more rapidly. Fig. (4) also shows the number of runs with errors outside a region of  $\pm 1, \pm 1, \pm 0.1$  in  $(x, y, \theta)$  error<sup>9</sup>. This indicates the divergent estimates as the measurement of two features should bring the pose into that region. In order to evaluate

<sup>&</sup>lt;sup>6</sup>  $\alpha$  and  $\beta$  are matricies from the  $\lambda \to \mathbf{x}$  spaces.

<sup>&</sup>lt;sup>7</sup> As in Thrun pages 220–228 (UKF), 250-252 (PF) and 110 (low variance sampling).

 $<sup>^{8}</sup>$  Q was  $10^{-5}$ ,  $10^{-4}$ , or  $10^{-3}$  along the diagonal.

 $<sup>^{9}</sup>$  The simulated distance units are arbitrary but can be thought of as meters while the angles are in radians.

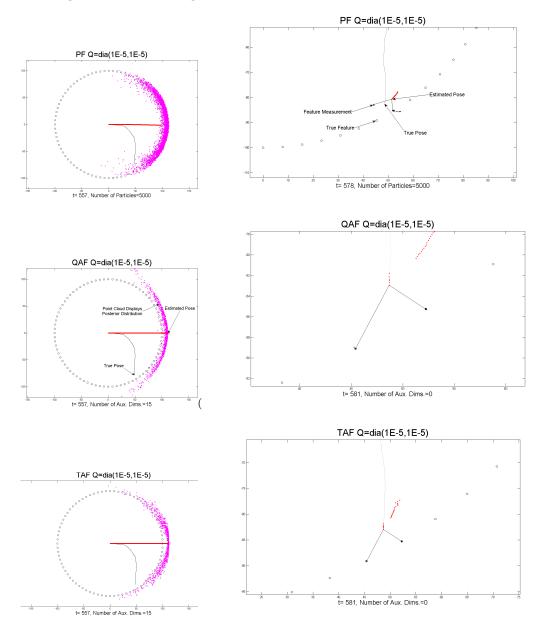


Fig. 3 Here we show the posterior distribution for the PF top (here we show 5,000 particles, while we later decided to use 2,000 and 20,000 for the experiments) QAF center and the TAF bottom. The left is just before observing the first feature and the right is 5 steps after observing the second features. The QAF and TAF posterior are displayed by sampling from it. The estimate is the thick (red) line while the thin (black) track is the true path. We see that for this outlier simulation on the left the PF is more consistent in its shape and the QAF is the least consistent. The better ends of the crescent for the TAF help it to update more accurately than the QAF as seen by the slight error in the first updates of the second feature shown as dots in the right column figures. In the bottom right figure these dots all lie on top of the true feature location (displayed as the ring of small circles) indicating that all the estimates explained the measurements. The QAF had the first of update with the second feature not quite line up the measurement of the feature with its true location. Both TAF and QAF could correct the pose accurately after seeing the second feature two times. The PF is hopelessly off as it had no particle near the correct pose. The same run for the 4 Gaussian estimators EKF, IEKF, UKF, and IUKF had the robot pose outside of the circle of features heading in. Clearly a Gaussian ellipse can not represent this situation.

consistency we computed a test statistic based on the mahalanobis distance

$$d^{2} = (\mathbf{x} - \mathbf{x}_{true})^{T} (Cov_{x})^{-1} (\mathbf{x} - \mathbf{x}_{true})$$
(31)

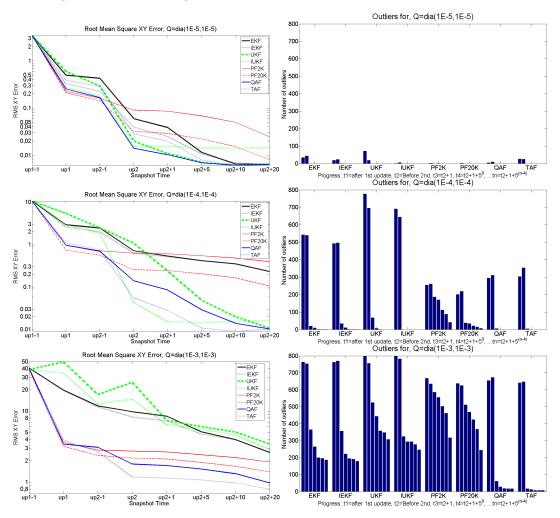
The cumulative  $\chi_3^2$  distribution of  $d^2$  was recorded at each time. This would be uniformly distributed if the error were Gaussian and the estimates were consistent. The errors should become approximately Gaussian after sufficient updates are performed. By sorting these values and plotting them we can compare the empirical distribution to the ideal straight line in the top left of fig. (5). It is not possible to show all these plots but we can summarize the consistency by plotting the Kolmogorov-Smirnov test statistic for all plots over time relative to the updates, fig. (5). This statistic is the maximum vertical distance between the empirical and ideal curves. The effect of particle depletion is seen by the very poor PF consistency shown here. The TAF has lower values in general and reaches a 'consistent' Guassian posterior sooner than the others.

Table 1 The mean computation times for the various filters for experiment 1 in sec.

Q(2,2)	EKF	IEKF	UKF	IUKF	PF2K	PF20K	QAF	TAF
1E-5	$0.65 \\ 0.65 \\ 0.57$	0.78	0.40	1.22	1.04	8.09	1.92	1.69
1E-4		0.80	0.40	1.22	1.03	7.91	2.17	2.11
1E-3		0.81	0.35	1.05	0.94	7.24	4.55	2.38

Table (1) summarizes the average computation times for experiment 1. This is not very informative as it depends on the matlab implementation details and the difficulty of the simulation but one can get some idea of the time trade-offs. We can see that the TAF seemed to be 2 times faster than the QAF when the nonlinearities were large (up to 13 auxiliary dimensions for the largest Q). This is most likely due to more easily moving to the MLE for the TAF parametrization. We can see a time factor of about 2-5 for the AF with moderate nonlinearities over the Gaussian estimators.

We then looked at how the errors behave after a series of updates each with insufficient information to fully localize the robot. We did simulations with a square 4 feature map and a true path that observed each feature in turn. We can see in fig. (6) that the AF were less sensitive to the build up of errors around the square.



**Fig. 4** Left is the root mean square error in x-y distance for small, medium and big Q. On the right are the number of simulations with pose error outside a box  $\pm 1, \pm 1, \pm 0.1$  for times: the first update, before second feature, the second feature update, then 1, 5, 10 and 20 after the second feature update. The AF are both seen to be more accurate than any of the other estimators.

#### 7 Conclusion

We have investigated a completely new type of nonlinear filter the AF. The AF methods can be significantly more accurate than the other popular recursive nonlinear estimators. The TAF was in some ways slightly better than the QAF on this problem for which it was tailored. The QAF is a more general filter. We also showed the new estimators were significantly more consistent

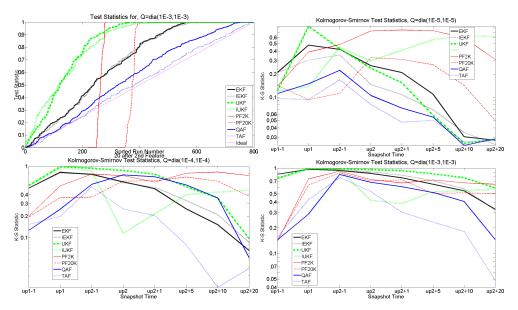


Fig. 5 The top left shows and example of the computation of the Kolmogrov-Smirnov, KS, statistic for the case of the big Q 20 updates after observing the second feature. The curves should lie on the straight line from 0 to 1 if the estimator was consistent. The KS-statistic is the maximum vertical distance between this line and the curve. In this case the AF estimators are much closer to consistent than the others. The evolution of the KS-statistic is shown in the other three plots for the three Q values. The TAF seems to generally be the least inconsistent estimator. The PF with 2,000 and 20,000 particles both become inconsistent.

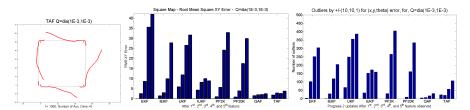


Fig. 6 The left shows and example from the 500 simulations in the second experiment. The TAF estimated path is shown as the thicker (red) curve while the true path is the thin black line. The four features are observed in turn starting and ending with the left bottom corner. We see that the estimate is able to mostly correct the angular error each time. Center we show the root mean square xy error 2 steps after each feature is used to correct the pose. The 2 steps is to let the estimate settle after the large change of first update. We see how the errors are amplified by all the estimators but the AF have much lower errors. The right shows the number of estimates outside a box  $\pm 10, \pm 10, \pm 1$  for  $(x, y, \theta)$  errors at these same times. The PF with 2,000 and 20,000 particles are especially poor at this test as the particle depletion gets worse each time. The IUKF was considerable better than any of the other Gaussian estimators. The QAF was the best, slightly better than the TAF. That is probably because the distance with no updates was half that of the experiment 1. This lead to smaller angular error and less difference in shape from parabolic.

than the EKF, IEKF, UKF, IUKF, and PF on a robot localization problem in 2D. In general the PF will be able to represent more shapes than the AF and so it is a more general solution. In particular it can estimate multimodal distributions. For problems that can be modeled approximatly by an AF, the AF has strengths making it a viable choice of estimator.

**Acknowledgements** This work was supported by the SSF through its Centre for Autonomous Systems (CAS).

## **Appendix**

#### References

- 1. R. Kalman, "A new approach filtering and prediction problems," *Trans. ASME, Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960.
- L. Ljung, "Asymptotic behavior of the extended kalman filter as a parameter estimator for linear systems," *IEEE Transaction on Automatic Control*, vol. 24, no. 1, pp. 36–51, Feb. 1979.
- 3. S. Huang and G. Dissanayake, "Convergence and consistency analysis for extended kalman filter based slam," *IEEE Transaction on Robotics and Automation*, vol. 23, no. 5, pp. 1036–1050, Oct. 2007.
- K. Reif, S. Gunther, E. Yaz, and R. Unbehauen, "Stochastic stability of the discrete-time extended kalman filter," *IEEE Transaction on Automatic Control*, vol. 44, no. 4, pp. 714–728, Apr. 1999.
- K. Xiong, H. Zhang, and L. Liu, "Adaptive robust extended kalman filter for nonlinear stochastic systems," Control Theory and Applications, IET, vol. 2, no. 3, pp. 239–250, Mar. 2008.
- 6. W. Zhang, B. S. Chen, and C. Tseng, "Robust  $H_{\infty}$  filtering for nonlinear stochastic systems," *IEEE Trans. on Signal Processing*, vol. 53, no. 2, pp. 589–598, Feb. 2005.
- L. Jetto, S. Longhi, and G. Venturini, "Development and experimental validation of an adaptive extended kalman filter for the localization of mobile robots," *IEEE Transaction on Robotics and Automation*, vol. 15, no. 2, pp. 219–229, Apr. 1999.
- 8. S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," in *Proc. of the IEEE*, vol. 92, 2004, pp. 401–422.
- E. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000, vol. 1, no. 1, pp. 153–158, Oct. 2000.
- 10. K. Xiong, H. Zhanga, and C. Chan, "Performance evaluation of UKF-based non-linear filtering," *Automatica*, vol. 42, no. 2, pp. 261–270, Feb. 2006.
- 11. K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Transaction on Automatic Control*, vol. 45, no. 5, pp. 910–927, May 2000.
- 12. T. Lefebvre, H. Bruyninck, and J. D. Schutter, "Kalman filters for non-linear systems: a comparison of performance," *Int. Journal of Control*, vol. 77, no. 7, pp. 639–653, May 2004.
- I. Arasaratnam, S. Haykin, and R. J. Elliott, "Discrete-time nonlinear filtering algorithms using gauss hermite quadrature," in *Proc. of the IEEE*, vol. 95, 2007, pp. 953–977.

 J. Clark, R. Vinter, and M. Yaqoob, "Shifted rayleigh filter: a new algorithm for bearings-only tracking," *IEEE Transaction on Aerospace and Electronic Systems*, vol. 43, no. 4, pp. 1373–1384, Oct. 2007.

- M. Norgaard, N. K. Poulsen, and O. Ravn, "New developments in state estimation for nonlinear systems," Automatica, vol. 36, pp. 1627–1638, 2000.
- B. Bell and F. Cathey, "The iterated kalman filter update as a gauss-newton method," *IEEE Transaction on Automatic Control*, vol. 38, no. 2, pp. 294–297, Feb. 1993.
- Y. Bar-Shalom and T. Fortmann, Tracking and Data Association. New York, NY.: Academic Press, 1987.
- 18. N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," in *Proc. of the IEEE for Radar and Signal Processing*, vol. 140, 1993, pp. 107–113.
- M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transaction* on Signal Processin, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- 20. J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filters for nonlinear problems," *IEEE Proceedings Radar, Sonar and Navigation*, vol. 146, no. 1, pp. 2–7, Feb. 1999.
- F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *Proc. of the IEEE Int. Conference on Robotics and Automation*, May 1999, pp. 1322–1328.
- 22. M. Montemerlo and et al., "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. of the Nat. Conf. on Artificial Intelligence (AAAI-02)*, Edmonton, Canada, 2002.
- G. Grisetti, C. Stachniss, and W. Burgard, "Improving grid-based slam with raoblackwellized particle filters by adaptive proposals and selective resampling," in Proc. of the IEEE Int. Conference on Robotics and Automation, Apr. 2005, pp. 2432–2437.
- 24. J. Kotecha and P. Djuric, "Gaussian particle filtering," *IEEE Transaction on Signal Processin*, vol. 51, no. 10, pp. 2592–2601, Oct. 2003.
- H. Durrant-Whyte, S. Majumder, S. Thrun, M. de Battista1, and S. Scheding, "A bayesian algorithm for simultaneous localisation and map building," Springer Tracts in Advanced Robotics, vol. 6, pp. 49–60, 2003.
- Y. Bresler and A. Macovski, "Exact maximum likelihood parameter estimation of superimposed exponential signals in noise," *IEEE Transaction on Acoustice,* Speech and Signal Processing, vol. 34, no. 5, pp. 1081–1089, 1986.
- K. R. Muske and J. B. Rawlings, Nonlinear moving horizon state estimation. Klewer, 1995.
- C. Rao, J. Rawlings, and D. Mayne, "Constrained state estimation for nonlinear discrete-time systems: stability and moving horizon approximations," *IEEE Transaction on Automatic Control*, vol. 48, no. 2, pp. 246–258, Feb. 2003.
- 29. J. Folkesson and H. I. Christensen, "Graphical SLAM a self-correcting map," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA04)*, vol. 1, 2004.
- F. Dellaert, "Square root sam: Simultaneous location and mapping via square root information smoothing," in Robotics: Science and Systems, 2005.
- 31. M. Briers, A. Doucet, and S. Singh, "Sequential auxiliary particle belief propagation," in *Proc. of the IEEE Introl. Conf. on Information Fusion*, vol. 1, 2005, pp. 1–8.
- 32. J. Folkesson, "Robustness of the quadratic antiparticle filter for robot localization," in *Proc. of the European Conference on Mobile Robots*, vol. 1, 2011.
- 33. S. Thrun, W. Burgard, and D. Fox, Probalistic robotics. MIT Press, 2005.