PGF 42: Progressive Gaussian Filtering with a Twist

Uwe D. Hanebeck
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
uwe.hanebeck@ieee.org

***Abstract*—A new Gaussian filter for estimating the state of nonlinear systems is derived that relies on two main ingredients: i) the progressive inclusion of the measurement information and ii) a tight coupling between a Gaussian density and its deterministic Dirac mixture approximation. No second Gaussian assumption for the joint density of state and measurement is required, so that the performance is much better than that of Linear Regression Kalman Filters (LRKFs), which heavily rely on this assumption. In addition, the new filter directly works with the generative system description. No Likelihood function is required. It can be used as a plug-in replacement for standard Gaussian filters such as the UKF.**

***Index Terms*—Gaussian filter, nonlinear Bayesian state estimation, nonlinear filtering, recursive estimation, progressive Bayesian estimation, homotopy, Dirac mixture approximation.**

I. Introduction

This paper is about Gaussian filters for high-dimensional nonlinear systems. The goal is to recursively estimate the hidden state of the system given noisy measurements arriving sequentially, a model for the relation of the state to measurements, and a model for the temporal state evolution. A filter is called a Gaussian filter when the estimated densities over the unknown state are represented by Gaussian densities.

Gaussian filters are attractive especially for high-dimensional systems, as they are usually simpler to implement than filters operating on more complex types of densities. Computational requirements are typically lower and the estimate can compactly be represented by mean vector and covariance matrix. On the downside, the simple density representation in some cases is not enough for representing the state estimate. In addition, various types of Gaussian filters exist that make further assumptions, such as a second Gaussian assumption between state and measurement. These filters, of course, do not achieve the performance of filters just imposing the state estimates to be Gaussian. However, without making additional assumptions, Gaussian filters are far more difficult to design.

In a nutshell, the proposed new Gaussian filter uses a deterministic Dirac mixture approximation (with an arbitrary number of optimally placed components) of a Gaussian density that is progressively updated. Instead of performing the filter step at once, the measurement information is introduced gradually. For that purpose, discrete information-preserving sub-update steps are defined that keep the updated density non-degenerate and close to Gaussian. For every sub-update, the Dirac mixture is re-weighted, but the weight change is immediately compensated by an appropriate modification of the component locations. By doing so, the components of the Dirac mixture approximation flow along the state space during the progression and are kept in the relevant regions of the state space. This allows i) to work with few components and ii) a direct high-quality update without further assumptions.

A. Problem Formulation

We consider a measurement equation of the form

$$y_k = h_k(\underline{x}_k, \underline{v}_k) \ , \tag{1}$$

with a time-varying measurement function $h_k(.,.)$, an N-dimensional state $\underline{x}_k$, and M-dimensional Gaussian noise $\underline{v}_k$ that enters in an arbitrary fashion. For simplicity, we assume a scalar measurement y_k with realization $\hat{y}_k$. In addition, we consider a system model describing the time evolution of the state according to

$$\underline{x}_{k+1} = \underline{a}_k(\underline{x}_k, \underline{w}_k) \ , \tag{2}$$

with time-varying system function $\underline{a}_k(.,.)$ and P-dimensional Gaussian noise $\underline{w}_k$. Possible inputs are already included in the mapping function.

For given noisy measurement realizations $\hat{y}_k$ sequentially received at time steps $k = 0, 1, 2, \ldots$, the goal is to recursively maintain the best possible Gaussian approximation of the true state density.

B. State of the Art

We will distinguish between two different types of Gaussian filters. The first type is the Gaussian assumed density filter (GADF). This filters directly approximates the true posterior resulting from multiplying the Gaussian prior and a Likelihood function derived from the given measurement equation and the noise description. The second type is the so called Linear Regression Kalman Filter (LRKF) [1]. This commonly used type of a Gaussian filter first approximates the (non-Gaussian) joint density of state and measurement corresponding to the prior density, the noise density, and the given nonlinear measurement equation by a jointly Gaussian density, which is independent of the actual measurement. In a second step, the posterior is calculated based on the given measurement.

Examples of LRKFs [2] are the Unscented Kalman Filter (UKF) [3] and its scaled version [4], its higher-order generalization [5], and a generalization to an arbitrary number of deterministic samples placed along the coordinate axes [6]. Filters performing an analytic or semi-analytic calculation of the required moments [7], [8] are based on a state decomposition into two parts, where one part can be calculated in closed form and the other part is calculated via a sample approximation [9]. Some filters calculate the required nonlinear moments of the prior Gaussian density based on numerical integration [10]. A progressive variant of an LRKF is proposed in [11].

GADFs are more complicated to implement in comparison to the LRKFs and there are various options for minimizing the deviation between the true posterior and its Gaussian approximation. One option is to employ moment matching, i.e., using the mean and covariance matrix of the true posterior as parameters for the desired Gaussian, as this is known to minimize the Kullback-Leibler distance between the two densities. Unfortunately, in the case of nonlinear measurement equations and the corresponding complicated Likelihood function, it is in general not a simple task to calculate mean and covariance matrix of the true posterior, as analytic solutions are a rare exception. In most cases, numerical integration over the true posterior, i.e., the product of the (Gaussian) prior density and the Likelihood function, is in order, such as Monte Carlo integration [12].

This paper generalizes the progressive Gaussian filter derived in [13] to the multi-dimensional case. In addition, it pursues a different progression approach close to the general idea presented in [14]. In [13], the progression is derived for a continuous progression parameter γ, so that we obtain a system of ordinary first-order differential equations (ODEs) for the parameters of the Gaussian density. Solving this system of ODEs gives the desired posterior. It is also what we call iterative, as the representation of the prior density is modified with increasing γ in order to keep the Dirac components in relevant regions. A general continuous and iterative progressive Bayesian approach has first been introduced [15] for state estimation with Gaussian mixture representations minimizing a squared-integral distance, which was generalized to the multi-dimensional case in [16]. Other distance measures are proposed in [17]. In contrast, the filtering method presented here and in [14] is directly derived with discrete progression steps and it is recursive in nature. Recursive here means that intermediate posteriors are calculated for each discrete progression step.

Besides state estimation, the progressive processing idea has been applied to moment calculation [18] and Gaussian mixture reduction [19]. A homotopy-based filtering method operating on a particle representation is given in [20].

Dirac mixture approximations for nonlinear estimation and filtering have been proposed for the case of scalar continuous densities in [21], [22] An algorithm for sequentially increasing the number of components is given in [23] and applied to recursive nonlinear prediction in [24]. Systematic Dirac mixture approximations of arbitrary multi-dimensional Gaussian densities are calculated in [25]. A more efficient method for the case of standard normal distributions with a subsequent transformation is given in [26]. Dirac mixtures calculated with this method are used throughout this paper.

C. Key Ideas

In this paper, a new type of Gaussian filter, called the Progressive Gaussian Filter (PGF 42), is derived that does not require a second Gaussian assumption for the joint density of state and measurement. Instead, the posterior Gaussian representation is calculated directly. For that purpose, a Dirac mixture representation of the Gaussian density is employed.

Without the second Gaussian assumption, directly performing the measurement update would lead to sample degeneration as mentioned above and many samples would be wasted. That would necessitate a very large number of samples for the approximation of the prior Gaussian density. Hence, the idea is to gradually include the measurement information instead of using all the information in one step. The total measurement update is decomposed into several sub-updates that are performed sequentially. A sub-update takes care of keeping all components alive. No component is down-weighted too much. This is achieved by adaptive sub-updates that keep the distortion of the given Dirac mixture caused by re-weighting within pre-specified bounds.

The result of a sub-update step is a re-weighted Dirac mixture with nonuniform weights. For the next sub-update, a uniformly weighted Dirac mixture is required. This re-sampling of deterministic Dirac mixtures could be achieved by a general optimization procedure as in [14]. This, however, would be too complex here, as we are just interested in the approximation of Gaussian densities. Hence, in this paper it is proposed to first approximate the nonuniformly weighted Dirac mixture by a continuous Gaussian that in a second step is re-approximated by a uniformly weighted Dirac mixture. Doing so not only avoids sample degeneration, but also causes the Dirac components to flow along the state space during the progression as weight changes are turned into location changes.

Although no second Gaussian assumption is imposed, an explicit use of the Likelihood representation of the given measurement equation is avoided. The desired backward inference (from measurement to state) for the measurement update is performed by a forward mapping, so that a generative model is sufficient. In the additive noise case, this is equivalent to the explicit use of the likelihood. For non-additive noise, artificial additive noise is introduced and the state is augmented with the original noise. After estimating the augmented state, the original state is recovered by marginalization.

In summary, the proposed new Gaussian filter requires the following ingredients:

i) An appropriate deterministic Dirac mixture representation of Gaussian densities,
ii) a progression schedule for performing discrete sub-updates,
iii) an adaptation mechanism for the "strengths" of the sub-updates in order to keep the distorted Dirac mixture non-degenerate and close to a Gaussian,
iv) and an efficient method for the re-approximation of a nonuniform Dirac mixture by a uniform one, done here via the corresponding continuous Gaussian density. This, in turn, requires the efficient conversion between Dirac mixtures and Gaussians.

Specific implementations for all these ingredients will be derived in the following.

II. Derivation of Progressive Gaussian Filtering

We will now derive the new Progressive Gaussian Filter (PGF 42) and all the details required for its efficient implementation. For that purpose, we will begin with a hypothetical treatment of the filtering problem in terms of continuous distributions. Then, Dirac mixture approximations are used for establishing the desired joint density by means of forward mappings, so that the method never actually requires the explicit use of the Likelihood function[1].

For the generative measurement equation in (1), a corresponding probabilistic model is given in the form of the conditional density

$$f(y_k|\underline{x}_k,\underline{v}_k) = \delta(y_k - h_k(\underline{x}_k,\underline{v}_k)) \ , \tag{3}$$

where $\delta(.)$ is the Dirac delta function. With Bayes' law, we obtain the posterior joint density of state and noise given the measurement sequence $y_{0:k}$ as

$$\begin{aligned} f(\underline{x}_k,\underline{v}_k \,|\, y_{0:k}) &= f(\underline{x}_k,\underline{v}_k \,|\, y_k, y_{0:k-1}) \\ &= \frac{f(y_k \,|\, \underline{x}_k,\underline{v}_k, y_{0:k-1}) \, f(\underline{x}_k,\underline{v}_k \,|\, y_{0:k-1})}{f(y_k \,|\, y_{0:k-1})} \ , \end{aligned}$$

where $f(y_k \,|\, \underline{x}_k,\underline{v}_k, y_{0:k-1}) = f(y_k \,|\, \underline{x}_k,\underline{v}_k)$ as the current measurement y_k is assumed to be conditionally independent of prior measurements given the current state of the system. The numerator on the right-hand-side is a constant according to $f(y_k \,|\, y_{0:k-1}) = 1/c_k$. The predicted state and noise given the measurements up to time step $k-1$ can be assembled from the prior state estimate $f(\underline{x}_k \,|\, y_{0:k-1})$ and the noise density $f(\underline{v}_k)$ according to

$$f(\underline{x}_k,\underline{v}_k \,|\, y_{0:k-1}) = f(\underline{x}_k \,|\, y_{0:k-1}) \, f(\underline{v}_k) \ ,$$

as the noise is considered to be independent of the state and of previous measurements. As a result, the desired posterior density of the state given the measurement sequence is given recursively as

$$\begin{aligned} &\underbrace{f(\underline{x}_k \,|\, y_k, y_{0:k-1})}_{f_k^e(\underline{x}_k)} = \int_{\mathbb{R}^M} f(\underline{x}_k,\underline{v}_k \,|\, y_k, y_{0:k-1}) \, d\,\underline{v}_k \\ &= c_k \int_{\mathbb{R}^M} \delta(y_k - h_k(\underline{x}_k,\underline{v}_k)) \, f(\underline{x}_k,\underline{v}_k \,|\, y_{0:k-1}) \, d\,\underline{v}_k \\ &= c_k \underbrace{\int_{\mathbb{R}^M} \delta(y_k - h_k(\underline{x}_k,\underline{v}_k)) \, f(\underline{v}_k) \, d\,\underline{v}_k}_{f_k^L(\underline{x}_k)} \ \underbrace{f(\underline{x}_k \,|\, y_{0:k-1})}_{f_k^p(\underline{x}_k)} \end{aligned}$$

or

$$f_k^e(\underline{x}_k) = c_k \, f_k^L(\underline{x}_k) \, f_k^p(\underline{x}_k) \ , \tag{4}$$

with posterior density $f_k^e(\underline{x}_k)$, Likelihood function $f_k^L(\underline{x}_k)$, and prior state density $f_k^p(\underline{x}_k)$ at time step k.

For a given Gaussian prior $f_k^p(\underline{x}_k)$, the posterior $f_k^e(\underline{x}_k)$ will not be Gaussian for arbitrary measurement functions — or corresponding Likelihood functions — and has to be approximated by a Gaussian density for the next recursion steps. As this is difficult for continuous densities, we directly work with discrete densities on a continuous domain, i.e., Dirac mixture densities, for representing prior and posterior densities.

Working with Dirac mixtures in (4) is still challenging as just multiplying the Dirac components with the Likelihood function leads to a starvation of components in regions of low Likelihood. In addition, Likelihood functions corresponding to measurement equations with non-additive noise such as (1) are either difficult to obtain or not available at all. Hence, it would be tempting to make the second Gaussian assumption that simplifies matters considerably as discussed above. However, this would limit the achievable estimation quality. In order to avoid this limitation, we will pursue a progressive filtering approach that neither requires the Likelihood function nor any other assumption besides prior and posterior being Gaussian distributed.

A. Progressive Filtering

We start by adding an artificial[2] noise source u_k to the right-hand-side of (1), which gives

$$y_k = h_k(\underline{x}_k,\underline{v}_k) + u_k \ . \tag{5}$$

We assume the noise u_k to be zero-mean Gaussian distributed with unnormalized density

$$f_k^u(u_k,\gamma) = \left(\exp\left(-\frac{1}{2} \frac{u_k^2}{\sigma_u^2} \right) \right)^{\gamma} \ , \tag{6}$$

where σ_u is a small standard deviation and γ is the progression parameter.

The resulting posterior for state and original noise is now given by

$$\begin{aligned} &f(\underline{x}_k,\underline{v}_k,\gamma \,|\, y_{0:k}) \\ &= c_k(\gamma) \int_{\mathbb{R}} f(\underline{x}_k,\underline{v}_k,u_k,\gamma \,|\, y_k, y_{0:k-1}) \, d\,\underline{u}_k \\ &= c_k(\gamma) \int_{\mathbb{R}} f(y_k \,|\, \underline{x}_k,\underline{v}_k,u_k) \, f_k^u(u_k,\gamma) \, d\,u_k \\ &\qquad \cdot f(\underline{x}_k,\underline{v}_k \,|\, y_{0:k-1}) \ , \end{aligned}$$

where $c_k(\gamma)$ is a normalization constant. Converting (5) into a probabilistic description gives

$$f(y_k \,|\, \underline{x}_k,\underline{v}_k,u_k) = \delta(y_k - h_k(\underline{x}_k,\underline{v}_k) - u_k)$$

and we finally obtain

$$\begin{aligned} \underbrace{f(\underline{x}_k,\underline{v}_k,\gamma \,|\, y_k, y_{0:k-1})}_{f_k^e(\underline{x}_k,\underline{v}_k,\gamma)} &= c_k(\gamma) \, f_k^u(y_k - h_k(\underline{x}_k,\underline{v}_k),\gamma) \\ &\quad \cdot \underbrace{f(\underline{x}_k,\underline{v}_k \,|\, y_{0:k-1})}_{f_k^p(\underline{x}_k,\underline{v}_k)} \ , \end{aligned} \tag{7}$$

with the modified prior and posterior densities over state *and* noise.

The progression parameter γ controls the inclusion of the original measurement information. Obviously, the case of $\gamma = 1$ corresponds to the original measurement equation when σ_u is small. For $\gamma = 0$, the modified measurement equation does not change the prior state estimate at all as we then have $f_k^u(u_k,\gamma) = 1$.

[1] It is usually more convenient to directly work with the generative measurement equation without deriving the Likelihood. However, when the Likelihood is available, the proposed filter can be used with little modification.

[2] In the case of purely additive noise, no artificial noise is required. The procedure detailed here is then used with the original noise description instead.

B. Dirac Mixture Representation

For further processing, the prior over state and noise is replaced by its Dirac mixture approximation

$$f_k^p(\underline{x}_k, \underline{v}_k) \approx \sum_{i=1}^{L} w_{k,i}^p \, \delta(\underline{z}_k - \hat{\underline{z}}_{k,i}^p) \tag{8}$$

with the augmented state $\underline{z}_k = \left[\underline{x}_k^T, \underline{v}_k^T\right]^T$, the Dirac components in the augmented state space $\hat{\underline{z}}_{k,i}^p = \left[(\hat{\underline{x}}_{k,i}^p)^T, (\hat{\underline{v}}_{k,i}^p)^T\right]^T$, and its weights $w_{k,i}^p$ for $i = 1, \ldots, L$.

For calculating this Dirac mixture approximation of a given Gaussian density, several options exist. Here, we use the approach derived in [26], where an optimal approximation of a standard normal distribution is efficiently performed offline in a first step. For a given arbitrary Gaussian density, the stored Dirac mixture approximation of the standard normal distribution is then online transformed in a second step. Of course, the resulting Dirac mixture approximation is not optimal anymore, but the online step is computationally cheap. An alternative would be to use the method for the direct and optimal Dirac mixture approximation of arbitrary Gaussians in [25]. This method, however, is computationally more expensive.

By inserting the Dirac mixture approximation into (7) we obtain

$$f_k^e(\underline{x}_k, \underline{v}_k, \gamma) \approx c_k(\gamma) \, f_k^u(\hat{y}_k - h_k(\underline{x}_k, \underline{v}_k), \gamma) \cdot \textstyle\sum_{i=1}^{L} w_{k,i}^p \, \delta(\underline{z}_k - \hat{\underline{z}}_{k,i}^p)$$

or

$$f_k^e(\underline{x}_k, \underline{v}_k, \gamma) \approx c_k(\gamma) \cdot \sum_{i=1}^{L} \underbrace{w_{k,i}^p \, f_k^u(\hat{y}_k - \bar{y}_{k,i}, \gamma)}_{w_{k,i}^e(\gamma)} \, \delta(\underline{z}_k - \hat{\underline{z}}_{k,i}^p)$$

with

$$\bar{y}_{k,i} = h_k(\hat{\underline{x}}_{k,i}^p, \hat{\underline{v}}_{k,i}^p) \ .$$

This means that the individual components do not change their locations, but are re-weighted depending on their distance from the actual measurement. The posterior Gaussian approximation could now be determined by calculating mean and covariance matrix of the density $f_k^e(\underline{x}_k, \underline{v}_k, \gamma)$ for $\gamma = 1$.

However, for increasing γ, we encounter the same effect as when performing the measurement update in one step. More and more components in (8) effectively are weighted to zero so that we loose a lot of components for representing the posterior Dirac mixture density $f_k^e(\underline{x}_k, \underline{v}_k, \gamma)$.

Hence, the key is to modify the given Dirac mixture approximation during the progression in such a way as to maintain its information content. This is generally achieved by compensating weight changes by location changes. One option is to use what we call an iterative progression, which replaces the samples used for representing $f_k^p(\underline{x}_k, \underline{v}_k)$ by more suitable ones based on the current intermediate posterior $f_k^e(\underline{x}_k, \underline{v}_k, \gamma)$. This is the approach taken in [13].

In contrast to iterative progression schemes, where the representation of the prior density is changed, we call changing the representation of the intermediate posterior a recursive progression scheme. This scheme is pursued in this paper.

It is important to note that both approaches, iterative and recursive progression, can either be derived in the form of an ordinary differential equation that describes the change of the parameters of the posterior Gaussian or in the form of discrete sub-updates. Of course, on a digital computer, the differential equation is solved in discrete steps anyway, so that both approaches result in similar implementations. In the next subsection, we will directly derive a recursive progression with discrete sub-update steps.

C. Recursive Update of Dirac Mixture Representation

In this subsection, we will now directly derive a discrete recursive progression method. After inserting the Dirac mixture representation of $f_k^p(\underline{x}_k, \underline{v}_k)$ into (7), we select an appropriate value of γ corresponding to a small distortion of the mixture so that it is still close to a Gaussian and far from being degenerate. Then, we replace the weighted Dirac mixture representing the intermediate posterior $f_k^e(\underline{x}_k, \underline{v}_k, \gamma)$ for the given γ by an unweighted one and continue the progression.

The intermediate posteriors $f_k^e(\underline{x}_k, \underline{v}_k, \gamma)$ can be recursively calculated over the progression by setting two consecutive values of γ, say γ_t and γ_{t+1}, into (7) and dividing the two expressions, which gives

$$\begin{aligned} & f_k^e(\underline{x}_k, \underline{v}_k, \gamma_{t+1}) \\ & = \underbrace{\frac{c_k(\gamma_{t+1})}{c_k(\gamma_t)}}_{d_k(\gamma_{t+1}, \gamma_t)} \, \frac{f_k^u(y_k - h_k(\underline{x}_k, \underline{v}_k), \gamma_{t+1})}{f_k^u(y_k - h_k(\underline{x}_k, \underline{v}_k), \gamma_t)} \, f_k^e(\underline{x}_k, \underline{v}_k, \gamma_t) \ , \end{aligned}$$

where $d_k(\gamma_{t+1}, \gamma_t)$ is a new normalization constant. From (6), we obtain

$$\begin{aligned} \frac{f_k^u(u_k, \gamma_{t+1})}{f_k^u(u_k, \gamma_t)} &= \frac{\left(\exp\left(-u_k^2 / \left(2\,\sigma_u^2\right)\right)\right)^{\gamma_{t+1}}}{\left(\exp\left(-u_k^2 / \left(2\,\sigma_u^2\right)\right)\right)^{\gamma_t}} \\ &= \left(\exp\left(-u_k^2 / \left(2\,\sigma_u^2\right)\right)\right)^{\Delta\gamma} = f_k^u(u_k, \Delta\gamma) \end{aligned}$$

with $\Delta\gamma_t = \gamma_{t+1} - \gamma_t$. This finally gives

$$\begin{aligned} f_k^e(\underline{x}_k, \underline{v}_k, \gamma_{t+1}) = d_k(\gamma_{t+1}, \gamma_t) \, & f_k^u(y_k - h_k(\underline{x}_k, \underline{v}_k), \Delta\gamma_t) \\ & \cdot f_k^e(\underline{x}_k, \underline{v}_k, \gamma_t) \ . \end{aligned} \tag{9}$$

In summary, (9) is a single discrete recursion step from a chain of discrete progression steps towards $\gamma = 1$. In this single step, we take the unweighted Dirac mixture approximation of $f_k^e(\underline{x}_k, \underline{v}_k, \gamma_t)$ and re-weight all of its components to obtain a representation for $f_k^e(\underline{x}_k, \underline{v}_k, \gamma_{t+1})$. The size of each step is selected in such a way that the weights are not down-weighted too much, i.e., remain non-degenerate and close to a Gaussian. Now, the challenge is that for the next recursion step, we again need an unweighted Dirac mixture approximation of $f_k^e(\underline{x}_k, \underline{v}_k, \gamma_{t+1})$. This problem will be solved in the next subsection.

D. Reapproximation of Nonuniformly Weighted Dirac Mixture by Uniformly Weighted One

After a certain sub-update step, we are given a Dirac mixture approximation of the underlying Gaussian density of $f_k^e(\underline{x}_k, \underline{v}_k, \gamma_t)$ as

$$f_k^e(\underline{x}_k, \underline{v}_k, \gamma_t) \approx \sum_{i=1}^{L} w_{k,i}^e(\gamma_t) \, \delta(\underline{z}_k - \hat{\underline{z}}_{k,i}^e(\gamma_t))$$

with nonuniform weights. The weights are stored in a weight vector

$$\underline{w}_k^e(\gamma_t) = \left[w_{k,1}^e(\gamma_t),\, w_{k,2}^e(\gamma_t),\, \ldots,\, w_{k,L}^e(\gamma_t)\right]^T$$

and the component locations are stored in a component matrix

$$\mathbf{Z}_k(\gamma_t) = \left[\underline{z}_{k,1}^e(\gamma_t),\, \underline{z}_{k,2}^e(\gamma_t),\, \ldots,\, \underline{z}_{k,L}^e(\gamma_t)\right] \quad .$$

Before performing the next progression step, this Dirac mixture approximation has to be re-approximated with a uniformly weighted one as the components cannot be down-weighted anymore without compromising its approximation quality. In order to obtain equal weights, the component locations have to be modified accordingly in order to compensate for the weight changes. For performing this re-approximation, the method proposed in [14] could be used. However, as we are eventually interested in a Gaussian representation of the true posterior, the solution proposed in this paper is to re-approximate the weighted Dirac mixture by an appropriate intermediate Gaussian and then to generate the new set of unweighted Dirac components from this Gaussian.

For converting the weighted Dirac mixture to a corresponding Gaussian representation of the intermediate posterior $f_k^e(\underline{x}_k, \underline{v}_k, \gamma_t)$, we calculate the sample mean $\hat{\underline{z}}_k^e$ and the sample covariance matrix $\mathbf{P}_k^e(\gamma_t)$ of the weighted Dirac mixture given by $\underline{w}_k^e(\gamma_t)$ and $\mathbf{Z}_k(\gamma_t)$.

Given the intermediate Gaussian posterior described by $\hat{\underline{z}}_k^e$ and $\mathbf{P}_k^e(\gamma_t)$, we can now obtain the desired uniformly weighted Dirac mixture approximation from it. Again, the method proposed in [26] is used.

E. Calculation of Progression Step Size

We now consider the maximum admissible step size during a sub-update that does not distort our state density representation too much.

First, we limit the distortion of the Dirac mixture approximation by defining a smallest ratio $R \in [0, 1]$ between the smallest component weight and the largest component weight of the Dirac mixture after a sub-update. This step size limit can be calculated before actually performing the next progression step. Given the hypothetical measurement $\bar{y}_{k,i} = h_k(\hat{\underline{z}}_{k,i}^e(\gamma_{t-1}))$ obtained by propagating the current sample set through the measurement function, the (unnormalized) posterior weights are given by

$$w_{k,i}^e(\gamma_t) = w_{k,i}^e(\gamma_{t-1}) \cdot \exp\left(-\frac{1}{2}\Delta\gamma \frac{(\hat{y}_k - \bar{y}_{k,i})^2}{\sigma_u^2}\right)$$

according to (9). For scalar measurements, we can now consider the posterior weights of the sub-update step according to the distance of the corresponding hypothetical measurement $\bar{y}_{k,i}$ from the true measurement $\hat{y}_k$. With the smallest distance d_{near} and the largest distance d_{far} and assuming equal prior weights, we obtain

$$R = \exp\left(-\frac{1}{2}\Delta\gamma \frac{d_{\text{far}}^2}{\sigma_u^2}\right) / \exp\left(-\frac{1}{2}\Delta\gamma \frac{d_{\text{near}}^2}{\sigma_u^2}\right)$$

or

$$\Delta\gamma = -2\,\sigma_u^2\, \log(R) \,/\, \left(d_{\text{far}}^2 - d_{\text{near}}^2\right) \quad . \tag{10}$$

Second, we control the information content by performing an update (forward update) with $\Delta\gamma$ followed by a reversed update (backward update) with $-\Delta\gamma$. When the backward update leads to an estimate that is close to the estimate before the forward update, the step size is acceptable (and can be increased). Otherwise, it is too large (and has to be decreased).

For the last progression step, $\Delta\gamma$ could lead to γ larger then one, so $\Delta\gamma$ has to be clipped accordingly.

III. Filtering based on Dirac Mixture

We will now put all the ingredients together and define a Gaussian filter comprising filter step and prediction step.

Filter Step: The complete filter step is summarized in Alg. 1 for a single measurement $\hat{y}_k$ taken at time step k (this index is omitted in the algorithm). First, in line 4, the state augmentation is performed by stacking state and measurement. The subsequent progression loop makes use of the **Update** subroutine given in Alg. 2. For a given step size $\Delta\gamma$, an update is performed (line 6) that is followed by a backward update with negative step size (line 7). When the backward update comes close to the previous estimate, this step size is accepted (line 9), used for incrementing γ (line 10), and increased (line 11). Otherwise, the step size is just decreased (line 14). For a successful progression step, the progression step counter is incremented[3] in line 12. From line 16 to line 18, $\Delta\gamma$ for the final step is limited in order to keep γ within the interval $[0, 1]$. Finally, when $\gamma = 1$ is reached, the desired posterior parameters are extracted from the augmented state representation in line 20.

Alg. 2 shows the **Update** subroutine that starts with converting a Gaussian given by mean and covariance matrix to a Dirac mixture approximation in line 1 by means of Gauss2DM(., .) that works according to Subsec. II-D. (Unweighted) Dirac components are mapped to hypothetical measurements in line 2. It is assumed that the measurement function $h(., .)$ from (1) accepts vectorized inputs (now written as $\underline{h}(.)$) and produces a corresponding output vector of hypothetical measurements, one for each Dirac component. From line 3 to line 9, the maximum allowable progression step size $\Delta\gamma$ is computed according to Subsec. II-E for the forward update. With the given $\Delta\gamma$, the actual re-weighting (also in vectorized form) is then performed in line 10, where $\odot$ denotes the element-by-element product. The conversion from a Dirac mixture approximation to a Gaussian given by mean and covariance matrix in line 11, called DM2Gauss(., .), is discussed in Subsec. II-B.

Prediction Step: The prediction step consists of i) assembling the mean and covariance matrix of the augmented state comprising the original state $\underline{x}_k$ and the noise term $\underline{w}_k$ in (2), ii) calculating its Dirac mixture approximation with Gauss2DM(., .), iii) propagating the Dirac components through the system model (2), and iv) converting the resulting (still uniformly weighted) Dirac mixture back to a Gaussian with DM2Gauss(., .).

IV. Evaluation

The proposed new Gaussian filter will now be compared to standard Gaussian filters. In order to give a quick insight into the achievable performance, a basic two-dimensional estimation

[3] Of course, it also makes sense to not only count the progression steps, but the actual number of updates performed.

Main Routine: PGF 42

Input : Prior mean $\hat{\underline{x}}^p$ and covariance matrix $\mathbf{C}^p$, current measurement $\hat{y}$, measurement mapping $h(.,.)$, noise covariance matrix $\mathbf{C}^v$, number of Dirac components L, and weight ratio R

Output : Posterior mean $\hat{\underline{x}}^e$ and covariance matrix $\mathbf{C}^e$, number of progression steps PC

```
// Initialize progression step counter
$\text{PC} = 0;$
// Initialize $\Delta\gamma$
$\Delta\gamma = 1;$
// Parameters for in-/decreasing $\Delta\gamma$
$\text{Down} = 0.5\,,\ \text{Up} = 1.5;$
// Initial sub-posterior Gaussian for $\gamma = 0$
$\hat{\underline{z}}^e = \begin{bmatrix} \hat{\underline{x}}^p \\ \underline{0}_M \end{bmatrix}\,,\ \mathbf{P}^e = \text{diag}(\mathbf{C}^p, \mathbf{C}^v);$
while $\gamma < 1$ do
    // Try forward update
    $[\hat{\underline{z}}^t, \mathbf{P}^t, \Delta\gamma] = \text{Update}(\hat{\underline{z}}^e, \mathbf{P}^e, \Delta\gamma);$
    // Corresponding backward update
    $[\hat{\underline{z}}^b, \mathbf{P}^b] = \text{Update}(\hat{\underline{z}}^t, \mathbf{P}^t, -\Delta\gamma);$
    if Backward update close to previous estimate? then
        // Make trial update the temporary estimate
        $\hat{\underline{z}}^e = \hat{\underline{z}}^t\,,\ \mathbf{P}^e = \mathbf{P}^t;$
        // Increment $\gamma$
        $\gamma = \gamma + \Delta\gamma;$
        // Increase step size
        $\Delta\gamma = \text{Up} * \Delta\gamma;$
        // Increment progression step counter PC
        $\text{PC} = \text{PC} + 1;$
    else
        // Decrease step size
        $\Delta\gamma = \text{Down} * \Delta\gamma;$
    end
    // Limit $\gamma$ to $[0, 1]$
    if $\gamma + \Delta\gamma > 1$ then
        $\Delta\gamma = 1 - \gamma;$
    end
end
// Extract posterior mean and covariance matrix
$\hat{\underline{x}}_k^e = \hat{\underline{z}}_{1:N}\,,\ \mathbf{C}_k^e = \mathbf{P}_{1:N,1:N}^e;$
```

Algorithm 1: Progressive Gaussian Filtering with PGF 42. The time index k has been omitted for the sake of clarity.

Subroutine: Update

Input : Prior mean $\hat{\underline{x}}^p$ and covariance matrix $\mathbf{C}^p$, current measurement $\hat{y}$, measurement mapping $h(.,.)$, noise covariance matrix $\mathbf{C}^v$, number of Dirac components L, weight ratio R, and number of progression steps PC

Output : Posterior mean $\hat{\underline{x}}^e$ and covariance matrix $\mathbf{C}^e$, number of progression steps PC

```
// Convert continuous to discrete representation
$[\underline{w}, \mathbf{Z}] = \text{Gauss2DM}(\hat{\underline{z}}^e, \mathbf{P}^e);$
// Map Dirac components to measurements
$\underline{\bar{y}} = \underline{h}(\mathbf{Z});$
// Forward update?
if $\Delta\gamma > 0$ then
    // Maximum allowable step size change
    $[d_{\text{near}}^2, d_{\text{far}}^2] = \text{min/max}\left((\hat{y} - \underline{\bar{y}})^2\right);$
    $\Delta\gamma_{\text{max}} = -2\,\sigma_u^2\,\log(R)\,/(d_{\text{far}}^2 - d_{\text{near}}^2);$
    // Cap to maximum allowable step size change
    if $\Delta\gamma > \Delta\gamma_{max}$ then
        $\Delta\gamma = \Delta\gamma_{\text{max}};$
    end
end
// Perform re-weighting and renormalization
$\underline{w} = \underline{w} \odot \exp\left(-\Delta\gamma\,\frac{1}{2}\,\frac{(\hat{y}-\underline{\bar{y}})^2}{\sigma_u^2}\right),\ \underline{w} = \underline{w}/(\underline{1}_L^T\,\underline{w});$
// Convert discrete to continuous representation
$[\hat{\underline{z}}^e, \mathbf{P}^e] = \text{DM2Gauss}(\underline{w}, \mathbf{Z});$
```

Algorithm 2: Update subroutine for Progressive Gaussian Filtering with PGF 42.

scenario is considered. The focus is on the filter step, for which new results have been derived in this paper.

We estimate the hidden state of a stochastic system based on scalar measurements $y_1, y_2 \in \mathbb{R}$ related to the system state $\underline{x} = [x_1, x_2]^T \in \mathbb{R}^2$ with the two measurement equations

$$y_i = h_i(\underline{x}) + v_i \ , \tag{11}$$

where v_i are zero-mean additive Gaussian noise terms with standard deviation σ_v. Two nonlinear measurement functions are considered, i.e.,

$$h_1(\underline{x}) = x_1^3 + x_2^3 \text{ and } h_2(\underline{x}) = x_1^3 - x_2^3 \ .$$

First, we select a true state $\tilde{\underline{x}}$ by drawing a random sample from a uniform distribution on the set $S = [-1, 1] \times [-1, 1]$. For an individual true state, two measurement realizations $\hat{y}_1$, $\hat{y}_2$ are produced based on (11) and the measurement functions $h_1(.)$, $h_2(.)$, respectively. A prior estimate $\hat{\underline{x}}^p$ is produced by adding a sample drawn from a zero-mean Gaussian prior density with covariance matrix

$$\mathbf{C}^p = \text{diag}(\sigma^p, \sigma^p)$$

to the true state. For estimating the true state $\tilde{\underline{x}}$, the filter first obtains the prior estimate $\hat{\underline{x}}^p$ with covariance matrix $\mathbf{C}^p$ and gets access to $\hat{y}_1$. An estimate $\underline{x}_1^e$ is calculated that comprises the expected value $\hat{\underline{x}}_1^e$ and the associated covariance matrix $\mathbf{C}_1^p$. Subsequently, a new estimate $\underline{x}_2^e$ comprising $\hat{\underline{x}}_2^e$ and $\mathbf{C}_2^p$ is calculated based on the previous estimate $\underline{x}_1^e$ and measurement $\hat{y}_2$. This is the final estimate.

This experiment is now performed 100 times for $\sigma_v = 0.1$ and 100 different prior noise levels adjusted by σ^p with $\sigma^p \in [0.1, 5]$. In general, the complexity of nonlinear estimation problems depends on how much the nonlinearity actually affects the filter. For low prior noise, the effect of smooth nonlinear measurement functions becomes negligible, so even simple filtering approaches achieve good performance. For increasingly uncertain prior information, however, the effective nonlinearity becomes more and more severe, calling for advanced filters when a decent estimation performance is expected.

The proposed new Progressive Gaussian Filter (PGF 42) is compared to the Unscented Kalman Filter (UKF) [3] and the Gaussian Particle Filter (GPF) [12] with the prior used as proposal density and 10^6 prior samples. For such a high number of prior samples, the GPF can be regarded as a reference in terms of the estimation quality achievable with a Gaussian filter, but is impractical in terms of computational complexity. For the PGF 42, only 15 components are used. The allowed weight ratio was selected to be $R = 0.1$. The choice of this value affects the number of progression steps required, but when selected within certain bounds, say 0.1 to 0.9, was found to be uncritical for the estimation quality.

The estimation results of the three filters — GPF, PGF 42, and UKF — are shown in Fig. 1. Each figure shows the Root Mean Square Error (RMSE) of the errors between the estimated and the true state for the respective filter. In addition, the minimum and maximum errors obtained for the simulations performed at each prior uncertainty level are shown.

GPF: As the number of samples is very large here, the GPF produces very good results and is used a a reference here. However, for smaller numbers of samples, we always encountered the problem of singular weight sets, so no estimate could be calculated. This is caused by sample sets placed in regions of low Likelihood, which of course happens with increasing frequency for larger prior uncertainties.

PGF 42: The PGF 42 produces results that are very close to the reference, the GPF with 10^6 samples. It is very fast as only 15 components are used and very reliable. The number of progression steps required by the PGF 42 is shown in Fig. 2, where for every prior uncertainty level, the average number of steps and the minimum/maximum numbers are shown.

UKF: The UKF degrades pretty quickly in this case and does not produce satisfying results for larger prior uncertainty. The RMSE quickly increases and a lot of estimates are really far from the truth.

V. Conclusions

A new Gaussian filter has been introduced that does not require a second Gaussian assumption between state and measurement and hence, is very accurate.

The key idea is to perform a progressive re-weighting and re-approximation of a deterministic Dirac mixture approximation by decomposing the measurement update into several sub-updates. By means of adapting the "strengths" of the sub-updates, degeneration of components is fully avoided. As a result, only a few components are required, which makes the filter very fast and efficient.

The new Gaussian filter offers a tradeoff between simplicity and performance that cannot easily be beaten. It is very elegant, easy to implement, and does not employ any sort of optimization method. The measurement equation can exhibit arbitrary non-additive noise structures and is used in generative form. No Likelihood function is required.

References

[1] T. Lefebvre, H. Bruyninckx, and J. De Schutter, "The linear regression kalman filter," in *Nonlinear Kalman Filtering for Force-Controlled Robot Tasks*, ser. Springer Tracts in Advanced Robotics, 2005, vol. 19.

[2] ——, "Kalman filters for non-linear systems: a comparison of performance," *International Journal of Control*, vol. 77, no. 7, pp. 639–653, 2004.

[3] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, Mar. 2000.

[4] S. J. Julier, "The scaled unscented transformation," in *American Control Conference, 2002. Proceedings of the 2002*, vol. 6. IEEE, 2002, pp. 4555– 4559 vol.6.

[5] D. Tenne and T. Singh, "The higher order unscented filter," in *American Control Conference, 2003. Proceedings of the 2003*, vol. 3. IEEE, Jun. 2003, pp. 2441– 2446 vol.3.

[6] M. F. Huber and U. D. Hanebeck, "Gaussian Filter based on Deterministic Sampling for High Quality Nonlinear Estimation," in *Proceedings of the 17th IFAC World Congress (IFAC 2008)*, vol. 17, no. 2, Seoul, Republic of Korea, Jul. 2008.

[7] M. F. Huber, F. Beutler, and U. D. Hanebeck, "Semi-Analytic Gaussian Assumed Density Filter," in *Proceedings of the 2011 American Control Conference (ACC 2011)*, San Francisco, California, USA, Jun. 2011.

[8] F. Beutler, M. F. Huber, and U. D. Hanebeck, "Semi-Analytic Stochastic Linearization for Range-Based Pose Tracking," in *Proceedings of the 2010 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2010)*, Salt Lake City, Utah, USA, Sep. 2010.

[9] ——, "Gaussian Filtering using State Decomposition Methods," in *Proceedings of the 12th International Conference on Information Fusion (Fusion 2009)*, Seattle, Washington, USA, Jul. 2009.

[10] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 910–927, May 2000.

[11] R. Zanetti, "Recursive update filtering for nonlinear estimation," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1481–1490, Jun. 2012.

[12] J. H. Kotecha and P. M. Djuric, "Gaussian particle filtering," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2592– 2601, Oct. 2003.

[13] U. D. Hanebeck and J. Steinbring, "Progressive Gaussian Filtering Based on Dirac Mixture Approximations," in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, Jul. 2012.

[14] P. Ruoff, P. Krauthausen, and U. D. Hanebeck, "Progressive Correction for Deterministic Dirac Mixture Approximations," in *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, Illinois, USA, Jul. 2011.

[15] U. D. Hanebeck, K. Briechle, and A. Rauh, "Progressive Bayes: A New Framework for Nonlinear State Estimation," in *Proceedings of SPIE, AeroSense Symposium*, vol. 5099, Orlando, Florida, USA, May 2003, pp. 256 – 267.

[16] U. D. Hanebeck and O. Feiermann, "Progressive Bayesian Estimation for Nonlinear Discrete-Time Systems: The Filter Step for Scalar Measurements and Multidimensional States," in *Proceedings of the 2003 IEEE Conference on Decision and Control (CDC 2003)*, Maui, Hawaii, USA, Dec. 2003, pp. 5366–5371.

[17] J. Hagmar, M. Jirstrand, L. Svensson, and M. Morelande, "Optimal parameterization of posterior densities using homotopy," in *2011 Proceedings of the 14th International Conference on Information Fusion (FUSION)*. IEEE, Jul. 2011, pp. 1–8.

[18] A. Rauh and U. D. Hanebeck, "Calculating Moments of Exponential Densities Using Differential Algebraic Equations," *IEEE Signal Processing Letters*, vol. 10, no. 5, pp. 144–147, May 2003.

[19] M. F. Huber and U. D. Hanebeck, "Progressive Gaussian Mixture Reduction," in *Proceedings of the 11th International Conference on Information Fusion (Fusion 2008)*, Cologne, Germany, Jul. 2008, pp. 1–8.

[20] F. Daum and J. Huang, "Nonlinear filters with particle flow induced by log-homotopy," *Proceedings of SPIE*, vol. 7336, no. 1, pp. 733 603–733 603–12, May 2009.

[21] O. C. Schrempf, D. Brunn, and U. D. Hanebeck, "Density Approximation Based on Dirac Mixtures with Regard to Nonlinear Estimation and Filtering," in *Proceedings of the 2006 IEEE Conference on Decision and Control (CDC 2006)*, San Diego, California, USA, Dec. 2006.

[22] ——, "Dirac Mixture Density Approximation Based on Minimization of the Weighted Cramér-von Mises Distance," in *Proceedings of the 2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006)*, Heidelberg, Germany, Sep. 2006, pp. 512–517.

[23] U. D. Hanebeck and O. C. Schrempf, "Greedy Algorithms for Dirac Mixture Approximation of Arbitrary Probability Density Functions," in *Proceedings of the 2007 IEEE Conference on Decision and Control (CDC 2007)*, New Orleans, Louisiana, USA, Dec. 2007, pp. 3065–3071.

[24] O. C. Schrempf and U. D. Hanebeck, "Recursive Prediction of Stochastic Nonlinear Systems Based on Optimal Dirac Mixture Approximations," in *Proceedings of the 2007 American Control Conference (ACC 2007)*, New York, New York, USA, Jul. 2007, pp. 1768–1774.

[25] U. D. Hanebeck, M. F. Huber, and V. Klumpp, "Dirac Mixture Approximation of Multivariate Gaussian Densities," in *Proceedings of the 2009 IEEE Conference on Decision and Control (CDC 2009)*, Shanghai, China, Dec. 2009.

[26] I. Gilitschenski and U. D. Hanebeck, "Efficient Deterministic Dirac Mixture Approximation (to appear)," in *Proceedings of the 2013 American Control Conference (ACC 2013)*, Washington D. C., USA, Jun. 2013.

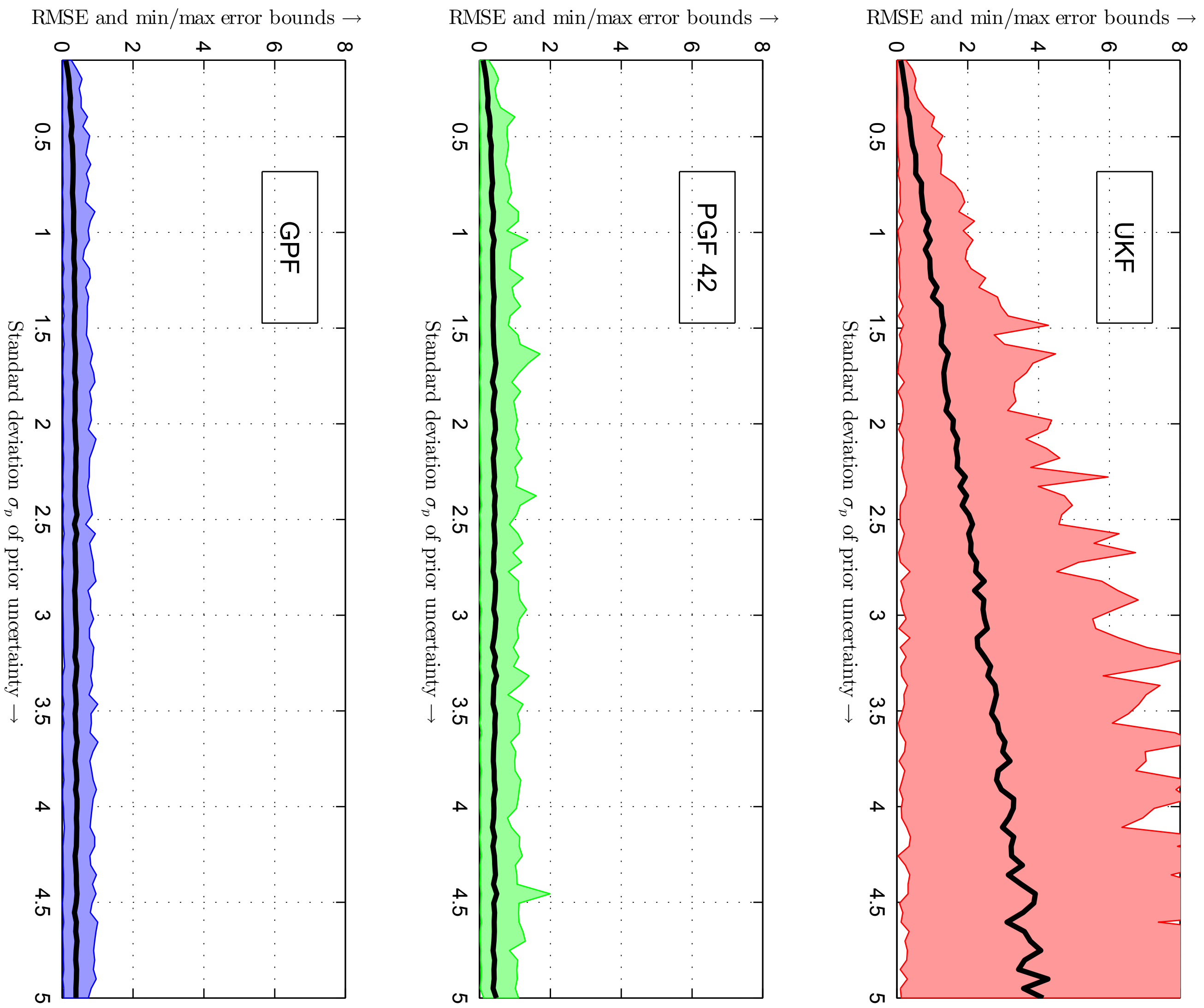

Figure 1. Estimation quality of three different Gaussian filters for different prior uncertainty levels. (Left) Gaussian Particle Filter (GPF). (Middle) The proposed new Progressive Gaussian Filter (PGF 42). (Right) The Unscented Kalman Filter (UKF).

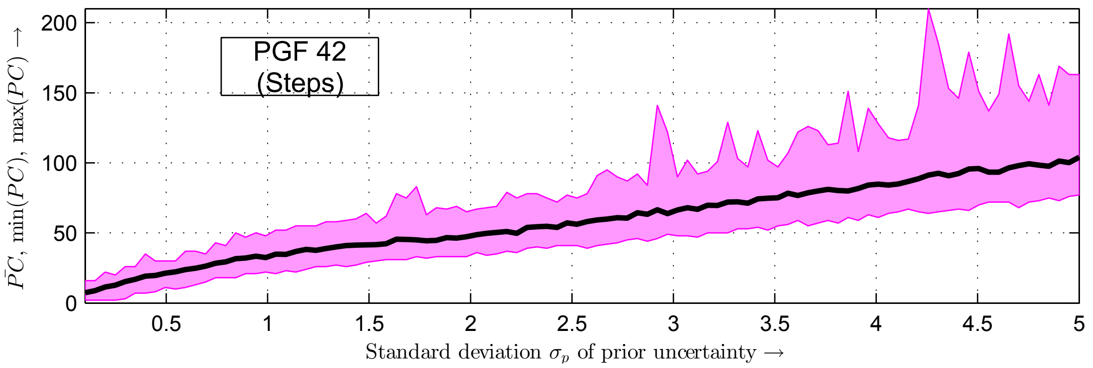

Figure 2. Progression steps used by the proposed new Progressive Gaussian Filter (PGF 42).