Semi-Analytic Progressive Gaussian Filtering

Jannik Steinbring, Antonio Zea, and Uwe D. Hanebeck

Abstract— **As an alternative to Kalman filters and particle filters, recently the progressive Gaussian filter (PGF) was proposed for estimating the state of discrete-time stochastic nonlinear dynamic systems. Like Kalman filters, the estimate of the PGF is a Gaussian distribution, but like particle filters, its measurement update works directly with the likelihood function in order to avoid the inherent linearization of the Kalman filters. However, compared to particle filters, the PGF allows for much faster state estimation and circumvents the severe problem of particle degeneracy by gradually transforming its prior Gaussian distribution into a posterior one. In this paper, we further enhance the estimation quality and runtime of the PGF by proposing a semi-analytic measurement update applicable to likelihood functions that only depend on a subspace of the system state. In fact, the proposed semi-analytic measurement update is not limited to the PGF and can be used by any nonlinear state estimator as long as its state estimate is Gaussian, e.g., the Gaussian particle filter.**

I. Introduction

When dealing with state estimation of discrete-time stochastic nonlinear dynamic systems, usually Kalman filters (KFs) are applied to the nonlinear models [1]–[8] or the more computational expensive particle filters (PFs) are used [9]–[13]. However, both filter classes have their individual drawbacks.

On the one hand, KFs inherently linearize the actual nonlinear relationship between state and measurement to perform the measurement update, that is, the joint density of state and measurement gets approximated as a Gaussian. This can be a very rough approximation and negatively effect the estimation quality.

On the other hand, PFs have the advantage of working directly with the likelihood function, and thus, do not perform a linearization for the measurement update. Additionally, PFs can represent multimodal densities in contrast to the unimodal KFs. An exception consists of the Gaussian particle filter (GPF) [10], which approximates the state distribution as a Gaussian as well. Nonetheless, the major problem of all PFs is particle degeneracy. This means that the particle reweighting during the measurement update can lead to a drastically reduced amount of particles with weights significantly greater zero or even situations where all particles get reweighted to zero. To circumvent this problem, a huge amount particles have to be used, especially for large state spaces, which in turn results in a heavy computational burden. Additionally, trying to utilize more sophisticated proposal densities can mitigate this problem, but this may require special treatment for different likelihood functions, or the proposal densities themselves can be computationally expensive. For example, the local linearization particle filter [12] uses a separate KF for each particle to obtain a better proposal density. Another way of reducing the amount of particles is to exploit linear/Gaussian substructures, called Rao–Blackwellization [14] or marginalized particle filters [15].

As an alternative to KFs and PFs, recently the progressive Gaussian filter (PGF) has emerged. First proposed in [16] and refined in [17], it relies on a Gaussian state distribution like KFs and the GPF, but avoids the linearization of the KF and instead also makes directly use of the likelihood function. However, in contrast to PFs, the PGF gradually, i.e., progressively, incorporates the information of a measurement into the current state estimate. For that, the prior Gaussian distribution is transformed into a posterior Gaussian distribution by computing multiple intermediate Gaussians. The PGF was already successfully applied to extended object tracking [18]–[20]. Additionally, a high-performance implementation on a graphics processing unit (GPU) was proposed in [19].

In this paper, we further improve estimation quality and runtime performance of the PGF by proposing a semi-analytic measurement update applicable to likelihood functions that do not depend on the entire system state, i.e., they only depend on a subspace of it, called the *observable state variables*. In such a case, the measurement update is twofold. First, we obtain an updated estimate only for the observable state variables by applying the standard measurement update of the PGF solely to the observable subspace of the system state. Second, the Gaussian state distribution allows for an analytic way to update the estimate of the *unobservable state variables* given the updated estimate for the observable state variables.

The analogous approach explicitly formulated for KFs exists [21], and is also called *state decomposition* [22]. In fact, the formulas for updating the estimate of the unobservable variables given the updated estimate of the observable variables are identical. The difference is how the updated estimate for the observable variables is computed. Furthermore, although marginalized particle filters also exploit special model structures, it is important to note that these filters perform a KF-like measurement update for the linear part of *each* particle. In contrast, for the PGF, a KF-like measurement update is performed only *once* in order to update the unobservable part of the PGF's state estimate. Additionally, these update formulas are different from those used by a marginalized particle filter, and are also used to update the correlation matrix of the observable and unobservable part

All authors are with the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Germany. E-mail: jannik.steinbring@kit.edu, antonio.zea@kit.edu, uwe.hanebeck@ieee.org

of the system state (that does not exist in a marginalized particle filter).

Our proposed semi-analytic PGF has several advantages. First, its estimation quality is improved due to the analytic treatment of the unobservable state variables. Second, the PGF measurement update only has to consider a subspace the system state, which reduces the amount of samples required for an update, and thus, reduces its runtime as well. Third, during the measurement update of the PGF, several Cholesky decompositions and computations of sample covariance matrices have to be conducted. Due to the respective complexities of $\mathcal{O}(N^3)$ and $\mathcal{O}(N^2)$, where N denotes the state space dimension, reducing the considered state space has a further runtime improvement. Finally, in some situations the PGF diverges when dealing with likelihood functions with unobservable state variables. The new semi-analytic measurement update can avoid this.

The remainder of this paper is structured as follows. First, in Section II, we describe the workflow of the progressive Gaussian filter. After that, we propose our semi-analytic version of the PGF in Section III. An evaluation of the new semi-analytic PGF by means of target tracking is performed in Section IV. Finally, the conclusions are given in Section V.

II. The Progressive Gaussian Filter

Like the popular Kalman filters, the progressive Gaussian filter estimates the system state $\underline{x} \in \mathbb{R}^N$ as a Gaussian distribution[1]. However, in contrast to Kalman filters, it does *not* approximate the joint distribution of state and measurement as a Gaussian. That is, the PGF directly tries to approximate the Bayesian measurement update

$$f(\underline{x} \,|\, \underline{\tilde{y}}) \approx \frac{f(\underline{\tilde{y}} \,|\, \underline{x}) f(\underline{x})}{\int_{\mathbb{R}^N} f(\underline{\tilde{y}} \,|\, \underline{x}) f(\underline{x}) \,\mathrm{d}\underline{x}} \ , \tag{1}$$

with received measurement $\underline{\tilde{y}}$, prior Gaussian state density

$$f(\underline{x}) = \mathcal{N}\left(\underline{x}; \underline{\mu}^{(x)}, \mathbf{\Sigma}^{(x)}\right) \ ,$$

arbitrary likelihood function $f(\underline{\tilde{y}} \,|\, \underline{x})$, and posterior Gaussian state density

$$f(\underline{x} \,|\, \underline{\tilde{y}}) = \mathcal{N}\left(\underline{x}; \underline{\mu}^{(x|\tilde{y})}, \mathbf{\Sigma}^{(x|\tilde{y})}\right) \ . \tag{2}$$

Note that, although the prior is assumed to be Gaussian, the true posterior (according to Bayes' rule) will not necessarily be Gaussian, and thus, we do not have equality in (1).

The key idea of the PGF to obtain the Gaussian posterior (2) is to use a so-called *progressive likelihood function* defined as

$$f(\underline{\tilde{y}} \,|\, \underline{x}, \gamma) := f(\underline{\tilde{y}} \,|\, \underline{x})^\gamma \ ,$$

with progression parameter $\gamma \in [0, 1]$. Based on this, the PGF considers the progressive Bayes update

$$f(\underline{x} \,|\, \underline{\tilde{y}}, \gamma) \approx \frac{f(\underline{\tilde{y}} \,|\, \underline{x}, \gamma) f(\underline{x})}{\int_{\mathbb{R}^N} f(\underline{\tilde{y}} \,|\, \underline{x}, \gamma) f(\underline{x}) \,\mathrm{d}\underline{x}} \ , \tag{3}$$

with progressive Gaussian posterior

$$f(\underline{x} \,|\, \underline{\tilde{y}}, \gamma) = \mathcal{N}\left(\underline{x}; \underline{\mu}^{(x|\tilde{y},\gamma)}, \mathbf{\Sigma}^{(x|\tilde{y},\gamma)}\right)$$

for $\gamma \in [0, 1]$. As with the Bayes update (1), the true posterior $f(\underline{x} \,|\, \underline{\tilde{y}}, \gamma)$ is not necessarily Gaussian, but it will be approximated as such throughout the measurement update of the PGF.

To get a better understanding of this progression approach, we consider two extreme cases. On the one hand, for $\gamma = 1$, we get the unmodified likelihood $f(\underline{\tilde{y}} \,|\, \underline{x})$, which gives

$$f(\underline{x} \,|\, \underline{\tilde{y}}, 1) \approx \frac{f(\underline{\tilde{y}} \,|\, \underline{x}) f(\underline{x})}{\int_{\mathbb{R}^N} f(\underline{\tilde{y}} \,|\, \underline{x}) f(\underline{x}) \,\mathrm{d}\underline{x}} \ ,$$

i.e., the measurement update (1). On the other hand, for $\gamma = 0$, we have

$$f(\underline{x} \,|\, \underline{\tilde{y}}, 0) = f(\underline{x}) \ ,$$

i.e., the progressive posterior equals the prior. In other words, for $\gamma = 0$, we do not incorporate any information of the received measurement $\underline{\tilde{y}}$ into the state estimate, whereas for $\gamma = 1$ all its information is processed.

Now, the progressive update (3) can be expressed in a recursive formula according to

$$f(\underline{x} \,|\, \underline{\tilde{y}}, \gamma + \Delta) \approx \frac{c(\gamma + \Delta)}{c(\gamma)} f(\underline{\tilde{y}} \,|\, \underline{x})^\Delta f(\underline{x} \,|\, \underline{\tilde{y}}, \gamma) \ , \tag{4}$$

for a given step size $\Delta > 0$ and $\gamma + \Delta \leq 1$. Here, the terms $c(\gamma)$ and $c(\gamma + \Delta)$ denote the normalization integrals of $f(\underline{x} \,|\, \underline{\tilde{y}}, \gamma)$ and $f(\underline{x} \,|\, \underline{\tilde{y}}, \gamma + \Delta)$, respectively. By starting the recursion with $\gamma = 0$, i.e., with the known Gaussian prior $f(\underline{x} \,|\, \underline{\tilde{y}}, 0) = f(\underline{x})$, and appropriate step sizes Δ, we can obtain multiple intermediate Gaussian distributions $f(\underline{x} \,|\, \underline{\tilde{y}}, \gamma)$ until we reach $\gamma = 1$, and thus, the desired Gaussian posterior $f(\underline{x}|\underline{\tilde{y}}, 1)$.

The complete procedure of the PGF is listed in Algorithm 1. A single recursion step is conducted with the aid of samples. More precisely, we approximate $f(\underline{x} \,|\, \underline{\tilde{y}}, \gamma)$ with a Dirac mixture, i.e., set of M *equally weighted* samples $\underline{x}^{(i)}$ according to

$$\frac{1}{M} \sum_{i=1}^{M} \delta(\underline{x} - \underline{x}^{(i)}) \ , \tag{5}$$

where $\delta(\cdot)$ denotes the Dirac delta distribution (line 3). The arbitrary number of samples M is the only user-defined parameter that controls the PGF. For any M, the samples (5) are generated with the optimal Gaussian sampling technique based on the localized cumulative distribution (LCD) [7], [8], [23]. This powerful sampling technique also builds the basis for the smart sampling Kalman filter (S^{2}KF) [7], [8]. An exemplary sampling of a 2D Gaussian distribution comprising $M = 31$ samples is depicted in Figure 1. Note that in order to obtain the samples $\{\underline{x}^{(i)}\}_{i=1}^M$ the Cholesky decomposition of the current intermediate covariance matrix $\mathbf{\Sigma}^{(x|\tilde{y},\gamma)}$ is required.

The determination of the step size Δ for the current recursion step is based on the log-likelihood evaluations of

[1] Vectors are underlined and matrices are printed bold face. For better readability, we omit a discrete time step k in this and the following section.

Algorithm 1 The Progressive Gaussian Filter

```
Input: prior state estimate $\underline{\mu}^{(x)}$ and $\mathbf{\Sigma}^{(x)}$, log-likelihood
    $\log(f(\underline{\tilde{y}}\,|\,\underline{x}))$, and number of samples $M$
 1: Set $\underline{\mu}^{(x|\tilde{y},\gamma)} \leftarrow \underline{\mu}^{(x)}, \mathbf{\Sigma}^{(x|\tilde{y},\gamma)} \leftarrow \mathbf{\Sigma}^{(x)}, \gamma \leftarrow 0$
 2: while $\gamma < 1$ do
 3:     $\{\underline{x}^{(i)}\}_{i=1}^{M}$ = LCD-Sampling($\underline{\mu}^{(x|\tilde{y},\gamma)}, \mathbf{\Sigma}^{(x|\tilde{y},\gamma)}, M$)
 4:     $l_i = \log(f(\underline{\tilde{y}}\,|\,\underline{x}^{(i)})) \quad \forall\, 1 \leq i \leq M$
 5:     $\mathcal{S} = \{l_i \mid \forall\, 1 \leq i \leq M \ \wedge \ l_i > -\infty\}$
 6:     $l_{\min} = \min(\mathcal{S}) \quad l_{\max} = \max(\mathcal{S})$
 7:     if $\mathcal{S} = \emptyset \ \vee \ l_{\min} = l_{\max}$ then
 8:         No progression possible $\Rightarrow$ abort update
 9:     end if
10:     $\Delta = -\log(M) \,/\, (l_{\min} - l_{\max})$
11:     if $\gamma + \Delta > 1$ then
12:         $\Delta = 1 - \gamma$
13:     end if
14:     $f(\underline{\tilde{y}}\,|\,\underline{x}^{(i)})^{\Delta} = \exp(\Delta(l_i - l_{\max})) \quad \forall\, 1 \leq i \leq M$
15:     $\alpha = \sum_{i=1}^{M} f(\underline{\tilde{y}}\,|\,\underline{x}^{(i)})^{\Delta}$
16:     $\alpha^{(i)} = f(\underline{\tilde{y}}\,|\,\underline{x}^{(i)})^{\Delta} \,/\, \alpha \quad \forall\, 1 \leq i \leq M$
17:     $\underline{\mu}^{(x|\tilde{y},\gamma)} \leftarrow \sum_{i=1}^{M} \alpha^{(i)} \underline{x}^{(i)}$
18:     $\mathbf{\Sigma}^{(x|\tilde{y},\gamma)} \leftarrow \sum_{i=1}^{M} \alpha^{(i)} (\underline{x}^{(i)} - \underline{\mu}^{(x|\tilde{y},\gamma)})(\underline{x}^{(i)} - \underline{\mu}^{(x|\tilde{y},\gamma)})^{\top}$
19:     $\gamma \leftarrow \gamma + \Delta$
20: end while
21: return posterior state estimate $\underline{\mu}^{(x|\tilde{y},\gamma)}$ and $\mathbf{\Sigma}^{(x|\tilde{y},\gamma)}$
```

the samples $\{\underline{x}^{(i)}\}_{i=1}^{M}$ (lines 4–13). That is, the step sizes Δ are not the same in each recursion step, and in turn yield an *adaptive* number of progression steps, specific to each measurement update.

Plugging (5) into (4) gives the reweighted Dirac mixture (line 14)

$$\sum_{i=1}^{M} \frac{c(\gamma + \Delta)}{c(\gamma) M} f(\underline{\tilde{y}}\,|\,\underline{x}^{(i)})^{\Delta} \delta(\underline{x} - \underline{x}^{(i)}) \ . \tag{6}$$

Normalizing the sample weights of (6) (lines 15–16) and subsequently computing sample mean and sample covariance matrix (lines 17–18) gives the intermediate Gaussian

$$f(\underline{x}\,|\,\underline{\tilde{y}}, \gamma + \Delta) = \mathcal{N}\left(\underline{x}; \underline{\mu}^{(x|\tilde{y},\gamma+\Delta)}, \mathbf{\Sigma}^{(x|\tilde{y},\gamma+\Delta)}\right)$$

for the next recursion step. In summary, in each recursion step a *resampling* with a fresh set of M equally weighted samples based on the current intermediate Gaussian $f(\underline{x}\,|\,\underline{\tilde{y}}, \gamma)$ is performed, and thus, sample degeneracy can be avoided. For a more detailed explanation of the PGF, please refer to [17].

Up to now, we have only considered the measurement update of the PGF. Concerning the state prediction, we distinguish between linear and nonlinear system models. In case of a linear system model, the prediction can be conducted in closed-form using the optimal Kalman filter prediction formulas. In case of a nonlinear system model, the PGF also utilizes the optimal Gaussian LCD sampling technique to generate a set of samples representing the current state estimate. By propagating these samples through the nonlinear system model and subsequently computing sample mean and sample covariance matrix of the propagated samples, we get the predicted state estimate. In fact, the nonlinear prediction of the PGF is identical to the nonlinear prediction of the S^{2}KF.

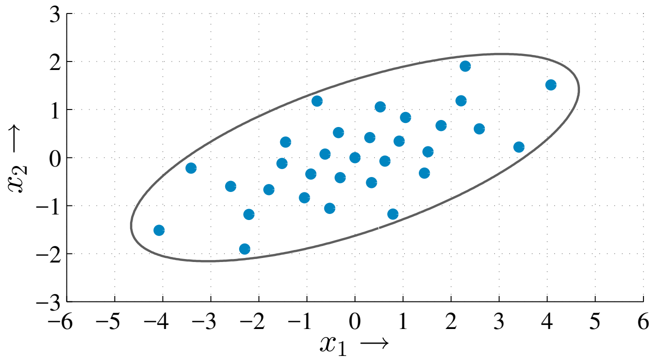


Fig. 1: Exemplary sampling of a 2D Gaussian obtained from the LCD sampling technique with 31 samples (blue dots) and 95% confidence interval of the Gaussian (gray ellipse).

Finally, an open-source implementation of the PGF (and other state-of-the-art state estimators like particle filters or the S^{2}KF) can be found in the nonlinear estimation toolbox [24].

III. Semi-Analytic Progressive Gaussian Filtering

After describing the workflow of the PGF for a general likelihood function $f(\underline{\tilde{y}}\,|\,\underline{x})$ in the previous section, we now consider a likelihood function that only depends on a subspace of $\underline{x}$. For that, we assume that the system state $\underline{x} \in \mathbb{R}^{A+B}$ is partitioned into two subspaces $\underline{a} \in \mathbb{R}^{A}$ and $\underline{b} \in \mathbb{R}^{B}$, that is,

$$\underline{x} = \begin{bmatrix} \underline{a} \\ \underline{b} \end{bmatrix} \ .$$

Like in Section II, the distribution of $\underline{x}$ is assumed to be Gaussian according to

$$f(\underline{x}) = f(\underline{a}, \underline{b}) = \mathcal{N}\left(\begin{bmatrix} \underline{a} \\ \underline{b} \end{bmatrix}; \begin{bmatrix} \underline{\mu}^{(a)} \\ \underline{\mu}^{(b)} \end{bmatrix}, \begin{bmatrix} \mathbf{\Sigma}^{(a)} & (\mathbf{\Sigma}^{(b,a)})^{\top} \\ \mathbf{\Sigma}^{(b,a)} & \mathbf{\Sigma}^{(b)} \end{bmatrix}\right) .$$

Next, we assume a nonlinear measurement model which only depends on the subspace $\underline{a}$, i.e.,

$$\underline{y} = \underline{h}(\underline{a}, \underline{v}) \ , \tag{7}$$

where $\underline{v} \in \mathbb{R}^{V}$ denotes measurement noise with density $f(\underline{v})$. Note that a received measurement $\underline{\tilde{y}}$ is a realization of the random vector $\underline{y}$. Due to (7), $\underline{a}$ contains the observable state variables and $\underline{b}$ the unobservable state variables. The corresponding likelihood function of (7) is given by

$$f(\underline{\tilde{y}}\,|\,\underline{x}) = f(\underline{\tilde{y}}\,|\,\underline{a}) = \int_{\mathbb{R}^{V}} \delta(\underline{\tilde{y}} - \underline{h}(\underline{a}, \underline{v})) f(\underline{v}) \,\mathrm{d}\underline{v} \ , \tag{8}$$

and of course solely depends on $\underline{a}$ as well.

With the likelihood (8), the posterior state density of the entire system state $\underline{x}$ is proportional to

$$\begin{aligned} f(\underline{x}\,|\,\underline{\tilde{y}}) = f(\underline{a},\underline{b}\,|\,\underline{\tilde{y}}) &\propto f(\underline{\tilde{y}}\,|\,\underline{x}) f(\underline{x}) \\ &= f(\underline{\tilde{y}}\,|\,\underline{a}) f(\underline{a},\underline{b}) \\ &= f(\underline{b}\,|\,\underline{a}) f(\underline{\tilde{y}}\,|\,\underline{a}) f(\underline{a}) \ . \end{aligned}$$

Note that $f(\underline{b}\,|\,\underline{a})$ is a conditionally Gaussian distribution with mean

$$\underline{\mu}^{(b|a)} = \underline{\mu}^{(b)} + \mathbf{K}(\underline{a} - \underline{\mu}^{(a)})$$

and covariance matrix

$$\mathbf{\Sigma}^{(b|a)} = \mathbf{\Sigma}^{(b)} - \mathbf{K}(\mathbf{\Sigma}^{(b,a)})^\top \ ,$$

where

$$\mathbf{K} = \mathbf{\Sigma}^{(b,a)} \mathbf{\Sigma}^{(a)^{-1}} \ .$$

Furthermore, we can obtain a posterior density only for the subspace $\underline{a}$ by again using Bayes' rule, i.e.,

$$f(\underline{a}\,|\,\underline{\tilde{y}}) = \frac{f(\underline{\tilde{y}}\,|\,\underline{a}) f(\underline{a})}{\int_{\mathbb{R}^A} f(\underline{\tilde{y}}\,|\,\underline{a}) f(\underline{a})\,\mathrm{d}\underline{a}} \ . \qquad (9)$$

Although $f(\underline{a})$ is Gaussian, this, in general, does not hold for $f(\underline{a}\,|\,\underline{\tilde{y}})$ due to the nonlinear measurement model $\underline{h}(\underline{a},\underline{v})$ and possible non-Gaussian measurement noise $\underline{v}$.

The key idea is now to get an Gaussian approximation of the posterior (9), i.e.,

$$f(\underline{a}\,|\,\underline{\tilde{y}}) \approx \mathcal{N}\left(\underline{a}; \underline{\mu}^{(a|\tilde{y})}, \mathbf{\Sigma}^{(a|\tilde{y})}\right) \ , \qquad (10)$$

by applying the PGF only to the subspace $\underline{a}$ and its prior Gaussian density $f(\underline{a})$. Based on this, the posterior density of the entire system state $\underline{x}$ can also be approximated as Gaussian distribution according to

$$\begin{aligned} f(\underline{a},\underline{b}\,|\,\underline{\tilde{y}}) &\propto f(\underline{b}\,|\,\underline{a}) f(\underline{a}\,|\,\underline{\tilde{y}}) \\ &\approx f(\underline{b}\,|\,\underline{a}) \mathcal{N}\left(\underline{a}; \underline{\mu}^{(a|\tilde{y})}, \mathbf{\Sigma}^{(a|\tilde{y})}\right) \\ &= \mathcal{N}\left(\begin{bmatrix}\underline{a}\\ \underline{b}\end{bmatrix}; \begin{bmatrix}\underline{\mu}^{(a|\tilde{y})}\\ \underline{\mu}^{(b|\tilde{y})}\end{bmatrix}, \begin{bmatrix}\mathbf{\Sigma}^{(a|\tilde{y})} & (\mathbf{\Sigma}^{(b,a|\tilde{y})})^\top \\ \mathbf{\Sigma}^{(b,a|\tilde{y})} & \mathbf{\Sigma}^{(b|\tilde{y})}\end{bmatrix}\right) , \end{aligned}$$

with

$$\begin{aligned} \underline{\mu}^{(b|\tilde{y})} &= \underline{\mu}^{(b)} + \mathbf{K}(\underline{\mu}^{(a|\tilde{y})} - \underline{\mu}^{(a)}) \ , \\ \mathbf{\Sigma}^{(b|\tilde{y})} &= \mathbf{\Sigma}^{(b)} + \mathbf{K}(\mathbf{\Sigma}^{(a|\tilde{y})} - \mathbf{\Sigma}^{(a)})\mathbf{K}^\top \ , \\ \mathbf{\Sigma}^{(b,a|\tilde{y})} &= \mathbf{K}\mathbf{\Sigma}^{(a|\tilde{y})} \ , \\ \mathbf{K} &= \mathbf{\Sigma}^{(b,a)} \mathbf{\Sigma}^{(a)^{-1}} \ . \end{aligned}$$

A proof is given in the Appendix. Thus, the posterior of the subspace $\underline{b}$ (and its correlation with the subspace $\underline{a}$) can be obtained in an analytic way given the updated, i.e., posterior, estimate of the subspace $\underline{a}$. As can be seen, the distribution of the subspace $\underline{b}$, and its correlation with the subspace $\underline{a}$, will only change if (i) prior correlations $\mathbf{\Sigma}^{(b,a)} \neq \mathbf{0}$ exists and (ii) the distribution of the subspace $\underline{a}$ changed, i.e., $\underline{\mu}^{(a|\tilde{y})} \neq \underline{\mu}^{(a)}$ and $\mathbf{\Sigma}^{(a|\tilde{y})} \neq \mathbf{\Sigma}^{(a)}$.

Note that our proposed semi-analytic approach can also directly applied to any other nonlinear estimator that maintains a Gaussian state estimate, e.g., the GPF. In such a case, (10) will be computed by using the measurement update of the GPF, not the measurement update of the PGF. The analytic part remains the same, though.

IV. Evaluation

In this section, we evaluate the proposed semi-analytic PGF by means of target tracking. We consider the target's system state $\underline{x}_k = \left[\underline{a}_k^\top, \underline{b}_k^\top\right]^\top$ with target position

$$\underline{a}_k = [p_k^{(\mathsf{x})}, p_k^{(\mathsf{y})}]^\top \ ,$$

in Cartesian coordinates and their respective velocities and accelerations

$$\underline{b}_k = [\dot{p}_k^{(\mathsf{x})}, \dot{p}_k^{(\mathsf{y})}, \ddot{p}_k^{(\mathsf{x})}, \ddot{p}_k^{(\mathsf{y})}]^\top \ .$$

Furthermore, we model the temporal evolution of the target with an constant acceleration model according to [25]

$$\underline{x}_k = \mathbf{A}\underline{x}_{k-1} + \mathbf{B}\underline{w} \ , \qquad (11)$$

with matrices

$$\mathbf{A} = \begin{bmatrix}\mathbf{I}_2 & T\mathbf{I}_2 & \frac{1}{2}T^2\mathbf{I}_2 \\ \mathbf{0} & \mathbf{I}_2 & T\mathbf{I}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_2\end{bmatrix} \ , \ \mathbf{B} = \begin{bmatrix}\frac{1}{2}T^2\mathbf{I}_2 \\ T\mathbf{I}_2 \\ \mathbf{I}_2\end{bmatrix} \ ,$$

respectively, $T = 0.01$, and zero-mean state-independent white Gaussian system noise $\underline{w}$ with covariance matrix $\mathbf{Q} = \mathrm{diag}(10^{-1}, 10^{-1})$.

At each time step k, we receive a noisy position measurement $\underline{\tilde{y}}_k$ in polar coordinates according to

$$\underline{y}_k = \underline{h}(\underline{a}_k) + \underline{v} = \begin{bmatrix}\|\underline{a}_k\|_2 \\ \mathrm{atan2}(p_k^{(\mathsf{y})}, p_k^{(\mathsf{x})})\end{bmatrix} + \underline{v} \ , \qquad (12)$$

where $\mathrm{atan2}(\cdot,\cdot)$ denotes the four-quadrant inverse tangent and $\underline{v}$ zero-mean state-independent white Gaussian noise with covariance matrix $\mathbf{R} = \mathrm{diag}(10^{-2}, 10^{-4})$. The corresponding likelihood function is given by

$$f(\underline{\tilde{y}}_k\,|\,\underline{a}_k) = \mathcal{N}\left(\underline{\tilde{y}}_k - \underline{h}(\underline{a}_k); \underline{0}, \mathbf{R}\right) \ .$$

As we can apply our proposed semi-analytic measurement update to any nonlinear filter that estimates the system state as a Gaussian distribution, we evaluate the following estimators:

- the PGF with $M = 101$ samples,
- the semi-analytic PGF with $M = 51$ samples,
- the GPF with 10^5 particles, and
- the semi-analytic GPF also with 10^5 particles.

Due to the linear system model (11), the state prediction for all investigated filters can be optimally computed in closed-form using the Kalman filter prediction formulas.

We perform 100 Monte Carlo runs. In each run, we initialize all filters with the mean $\underline{\mu}_0^{(x)} = [1, 1, 0, 0, 0, 0]^\top$ and the covariance matrix $\mathbf{\Sigma}_0^{(x)} = 10^2\mathbf{I}_6$. Then, an initial system state is created by randomly drawing a sample from this Gaussian distribution. Based on this, for 200 time steps we simulate the temporal behavior of the system by propagating the current system state, together with a random realization for $\underline{w}$, through the system model (11). Furthermore, in each time step k we simulate a noisy measurement $\tilde{\underline{y}}_k$ by evaluating (12) together with a random realization of $\underline{v}$.

We assess the estimation quality by computing the root mean square error (RMSE) over all runs for the target position $[p_k^{(\mathrm{x})}, p_k^{(\mathrm{y})}]^\top$, the target velocity $[\dot{p}_k^{(\mathrm{x})}, \dot{p}_k^{(\mathrm{y})}]^\top$, and the target acceleration $[\ddot{p}_k^{(\mathrm{x})}, \ddot{p}_k^{(\mathrm{y})}]^\top$, respectively. The results are depicted in Figures 2a–2c. First, for the position error, it can be seen that both PGF versions are nearly identical. However, keep in mind that the semi-analytic PGF only uses half of the samples used by the standard PGF. The semi-analytic GPF, although equipped with several particles, is not as good as the PGFs. The by far worst estimator is the standard GPF. Second, for the velocity error, the semi-analytic PGF is slightly better than the standard PGF. Like for the position error, the semi-analytic GPF is worse than the PGFs. The standard GPF again is the worst estimator. Finally, the acceleration errors reveal a significant difference in the estimation quality of the PGF and its semi-analytic version. At the beginning, the semi-analytic PGF is much better than the standard PGF. Additionally, the semi-analytic GPF has a much larger error than the semi-analytic PGF between the time steps 20 and 100. As with the position and velocity errors, the standard GPF delivers the worst results of all filters.

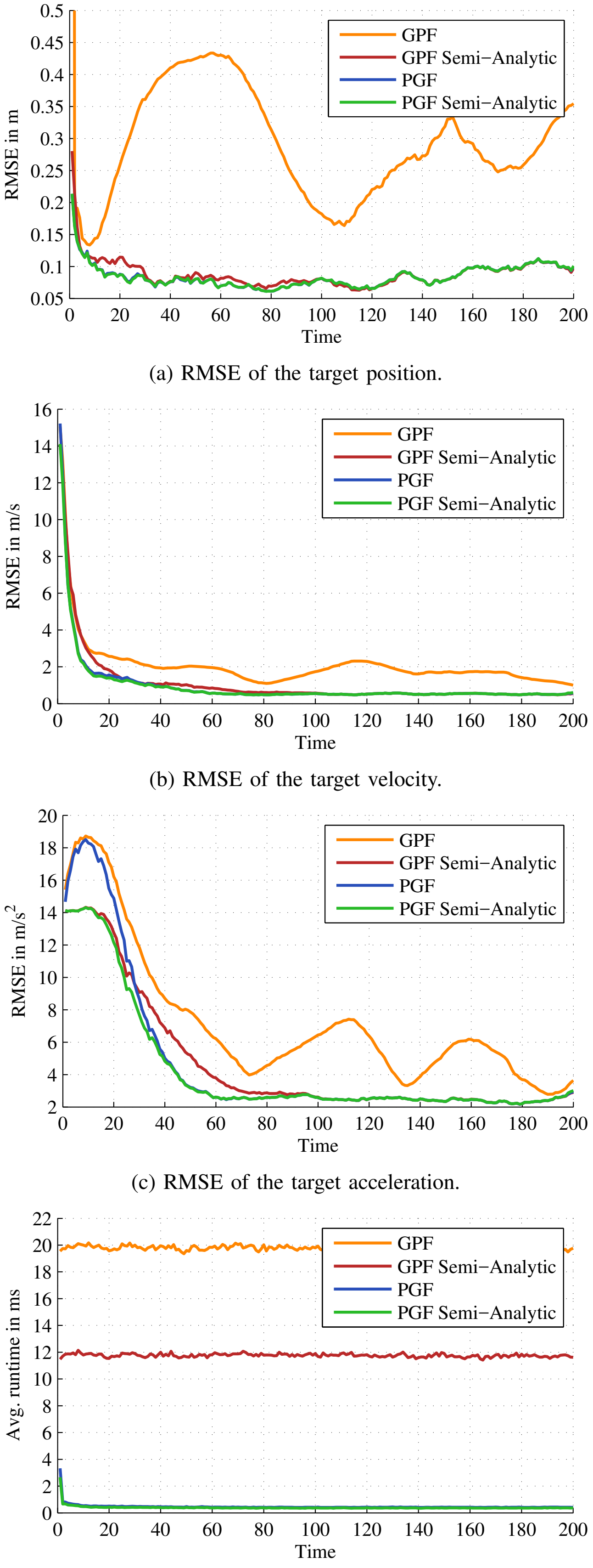
Now, when looking at the averaged runtimes of the measurement updates from the respective filters in Figure 2d, we see that the Gaussian particle filters have a much higher runtime than the progressive Gaussian filters. However, for the GPF there is a significant improvement when using the semi-analytic measurement update. This can be explained with the reduced overhead for computing sample mean and sample covariance to obtain the posterior state estimate. For the PGF, the semi-analytic version is only slightly faster. Hence, for this scenario the semi-analytic version of the PGF yields no significant runtime improvement due to the rather small system state $\underline{x}_k$ and small number of employed samples.

V. Conclusions

In this paper, we presented a semi-analytic measurement update for the progressive Gaussian filter. It can be applied if the likelihood function only depends on a subspace of the system state, and exploits the fact that the state estimate of the PGF is a Gaussian distribution. In doing so, the PGF's estimation quality can be improved, its runtime reduced, and potential filter divergence avoided. In addition, the proposed semi-analytic measurement update is not limited to the PGF. In fact, it can be used by any nonlinear state estimator as long as its state estimate is Gaussian, e.g., the Gaussian particle filter. The target tracking evaluation showed that our proposed semi-analytic measurement update can improve the estimation quality of the PGF. Furthermore, the Gaussian particle filter can also profit from the semi-analytic measurement update. In summary, the semi-analytic PGF was the best estimator regarding estimation quality and runtime.

Appendix

We prove that the product of the conditionally Gaussian distribution $f(\underline{b}\,|\,\underline{a})$ and the Gaussian approximation



(a) RMSE of the target position.

(b) RMSE of the target velocity.

(c) RMSE of the target acceleration.

(d) Averaged runtime for the measurement update.

Fig. 2: Target tracking results.

$\mathcal{N}\left(\underline{a}; \underline{\mu}^{(a|\underline{\tilde{y}})}, \mathbf{\Sigma}^{(a|\underline{\tilde{y}})}\right)$ yields an unnormalized Gaussian distribution. We have

$$f(\underline{b}\,|\,\underline{a})\,\mathcal{N}\left(\underline{a}; \underline{\mu}^{(a|\tilde{y})}, \mathbf{\Sigma}^{(a|\tilde{y})}\right) \propto \exp\left(-\frac{1}{2}z\right) \ ,$$

with

$$\begin{aligned} z &= (\underline{b} - \underline{\mu}^{(b|a)})^\top {\mathbf{\Sigma}^{(b|a)}}^{-1} (\underline{b} - \underline{\mu}^{(b|a)}) + \\ &\quad (\underline{a} - \underline{\mu}^{(a|\tilde{y})})^\top {\mathbf{\Sigma}^{(a|\tilde{y})}}^{-1} (\underline{a} - \underline{\mu}^{(a|\tilde{y})}) \\ &= (\underline{b} - \underline{\mu}^{(b)} - \mathbf{K}(\underline{a} - \underline{\mu}^{(a)}))^\top {\mathbf{\Sigma}^{(b|a)}}^{-1} \cdot \\ &\quad (\underline{b} - \underline{\mu}^{(b)} - \mathbf{K}(\underline{a} - \underline{\mu}^{(a)})) + \\ &\quad (\underline{a} - \underline{\mu}^{(a|\tilde{y})})^\top {\mathbf{\Sigma}^{(a|\tilde{y})}}^{-1} (\underline{a} - \underline{\mu}^{(a|\tilde{y})}) \ . \end{aligned}$$

First, extending this by adding twice the term $\underline{\mu}^{(a|\tilde{y})} - \underline{\mu}^{(a|\tilde{y})}$ and define

$$\underline{\mu}^{(b|\tilde{y})} := \underline{\mu}^{(b)} + \mathbf{K}(\underline{\mu}^{(a|\tilde{y})} - \underline{\mu}^{(a)}) \ ,$$

we get

$$\begin{aligned} z &= (\underline{b} - \underline{\mu}^{(b)} - \mathbf{K}(\underline{\mu}^{(a|\tilde{y})} - \underline{\mu}^{(a|\tilde{y})} + \underline{a} - \underline{\mu}^{(a)}))^\top {\mathbf{\Sigma}^{(b|a)}}^{-1} \cdot \\ &\quad (\underline{b} - \underline{\mu}^{(b)} - \mathbf{K}(\underline{\mu}^{(a|\tilde{y})} - \underline{\mu}^{(a|\tilde{y})} + \underline{a} - \underline{\mu}^{(a)})) + \\ &\quad (\underline{a} - \underline{\mu}^{(a|\tilde{y})})^\top {\mathbf{\Sigma}^{(a|\tilde{y})}}^{-1} (\underline{a} - \underline{\mu}^{(a|\tilde{y})}) \\ &= (\underline{a} - \underline{\mu}^{(a|\tilde{y})})^\top \left({\mathbf{\Sigma}^{(a|\tilde{y})}}^{-1} + \mathbf{K}^\top {\mathbf{\Sigma}^{(b|a)}}^{-1} \mathbf{K}\right) \cdot \\ &\quad (\underline{a} - \underline{\mu}^{(a|\tilde{y})}) + \\ &\quad (\underline{b} - \underline{\mu}^{(b|\tilde{y})})^\top {\mathbf{\Sigma}^{(b|a)}}^{-1} (\underline{b} - \underline{\mu}^{(b|\tilde{y})}) - \\ &\quad (\underline{b} - \underline{\mu}^{(b|\tilde{y})})^\top {\mathbf{\Sigma}^{(b|a)}}^{-1} \mathbf{K} (\underline{a} - \underline{\mu}^{(a|\tilde{y})}) - \\ &\quad (\underline{a} - \underline{\mu}^{(a|\tilde{y})})^\top \mathbf{K}^\top {\mathbf{\Sigma}^{(b|a)}}^{-1} (\underline{b} - \underline{\mu}^{(b|\tilde{y})}) \\ &= \left(\begin{bmatrix} \underline{a} \\ \underline{b} \end{bmatrix} - \begin{bmatrix} \underline{\mu}^{(a|\tilde{y})} \\ \underline{\mu}^{(b|\tilde{y})} \end{bmatrix}\right)^\top \mathbf{\Sigma}^{-1} \left(\begin{bmatrix} \underline{a} \\ \underline{b} \end{bmatrix} - \begin{bmatrix} \underline{\mu}^{(a|\tilde{y})} \\ \underline{\mu}^{(b|\tilde{y})} \end{bmatrix}\right) \ , \end{aligned}$$

with block matrix

$$\mathbf{\Sigma}^{-1} = \begin{bmatrix} {\mathbf{\Sigma}^{(a|\tilde{y})}}^{-1} + \mathbf{K}^\top {\mathbf{\Sigma}^{(b|a)}}^{-1} \mathbf{K} & -\mathbf{K}^\top {\mathbf{\Sigma}^{(b|a)}}^{-1} \\ -{\mathbf{\Sigma}^{(b|a)}}^{-1} \mathbf{K} & {\mathbf{\Sigma}^{(b|a)}}^{-1} \end{bmatrix} \ .$$

Second, using the formulas for inverting a 2×2 block matrix and equating the coefficients, we get the inverse of $\mathbf{\Sigma}^{-1}$ according to

$$\begin{aligned} \mathbf{\Sigma} &= \begin{bmatrix} \mathbf{\Sigma}^{(a|\tilde{y})} & \mathbf{\Sigma}^{(a|\tilde{y})} \mathbf{K}^\top \\ \mathbf{K} \mathbf{\Sigma}^{(a|\tilde{y})} & \mathbf{\Sigma}^{(b)} + \mathbf{K}(\mathbf{\Sigma}^{(a|\tilde{y})} - \mathbf{\Sigma}^{(a)}) \mathbf{K}^\top \end{bmatrix} \\ &=: \begin{bmatrix} \mathbf{\Sigma}^{(a|\tilde{y})} & (\mathbf{\Sigma}^{(b,a|\tilde{y})})^\top \\ \mathbf{\Sigma}^{(b,a|\tilde{y})} & \mathbf{\Sigma}^{(b|\tilde{y})} \end{bmatrix} \ . \end{aligned}$$

Finally, this yields

$$\begin{aligned} & f(\underline{b}\,|\,\underline{a})\,\mathcal{N}\left(\underline{a}; \underline{\mu}^{(a|\tilde{y})}, \mathbf{\Sigma}^{(a|\tilde{y})}\right) \propto \\ & \qquad \mathcal{N}\left(\begin{bmatrix} \underline{a} \\ \underline{b} \end{bmatrix}; \begin{bmatrix} \underline{\mu}^{(a|\tilde{y})} \\ \underline{\mu}^{(b|\tilde{y})} \end{bmatrix}, \begin{bmatrix} \mathbf{\Sigma}^{(a|\tilde{y})} & (\mathbf{\Sigma}^{(b,a|\tilde{y})})^\top \\ \mathbf{\Sigma}^{(b,a|\tilde{y})} & \mathbf{\Sigma}^{(b|\tilde{y})} \end{bmatrix}\right) \ . \end{aligned}$$

References

[1] Kazufumi Ito and Kaiqi Xiong, “Gaussian Filters for Nonlinear Filtering Problems,” *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 910–927, May 2000.

[2] Magnus Nørgaard, Niels K. Poulsen, and Ole Ravn, “New Developments in State Estimation for Nonlinear Systems,” *Automatica*, vol. 36, no. 11, pp. 1627–1638, 2000.

[3] Simon J. Julier and Jeffrey K. Uhlmann, “Unscented Filtering and Nonlinear Estimation,” in *Proceedings of the IEEE*, vol. 92, Mar. 2004, pp. 401–422.

[4] Dan Simon, *Optimal State Estimation*, 1st ed. Wiley & Sons, 2006.

[5] Jindřich Duník, Ondřej Straka, and Miroslav Šimandl, “The Development of a Randomised Unscented Kalman Filter,” in *Proceedings of the 18th IFAC World Congress*, Milano, Italy, Aug. 2011, pp. 8–13.

[6] Bin Jia, Ming Xin, and Yang Cheng, “High-Degree Cubature Kalman Filter,” *Automatica*, vol. 49, no. 2, pp. 510–518, Feb. 2013.

[7] Jannik Steinbring and Uwe D. Hanebeck, “LRKF Revisited: The Smart Sampling Kalman Filter (S^{2}KF),” *Journal of Advances in Information Fusion*, vol. 9, no. 2, pp. 106–123, Dec. 2014.

[8] Jannik Steinbring, Martin Pander, and Uwe D. Hanebeck, “The Smart Sampling Kalman Filter with Symmetric Samples,” *Journal of Advances in Information Fusion*, vol. 11, no. 1, pp. 71–90, Jun. 2016.

[9] Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp, “A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[10] Jayesh H. Kotecha and Petar M. Djuric, “Gaussian Particle Filtering,” *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2592–2601, Oct. 2003.

[11] ——, “Gaussian Sum Particle Filtering,” *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2602–2612, Oct. 2003.

[12] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House Publishers, 2004.

[13] Arnaud Doucet and Adam M. Johansen, “A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later,” in *Oxford Handbook of Nonlinear Filtering*, 2011, pp. 656–704.

[14] George Casella and Christian P. Robert, “Rao-Blackwellisation of Sampling Schemes,” *Biometrika*, vol. 81, no. 1, pp. 81–94, 1996.

[15] Thomas Schön, Fredrik Gustafsson, and Per-Johan Nordlund, “Marginalized Particle Filters for Mixed Linear/Nonlinear State-Space Models,” *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2279–2289, Jul. 2005.

[16] Uwe D. Hanebeck, “PGF 42: Progressive Gaussian Filtering with a Twist,” in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013.

[17] Jannik Steinbring and Uwe D. Hanebeck, “Progressive Gaussian Filtering Using Explicit Likelihoods,” in *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*, Salamanca, Spain, Jul. 2014.

[18] Antonio Zea, Florian Faion, and Uwe D. Hanebeck, “Exploiting Clutter: Negative Information for Enhanced Extended Object Tracking,” in *Proceedings of the 18th International Conference on Information Fusion (Fusion 2015)*, Washington D. C., USA, Jul. 2015.

[19] Jannik Steinbring and Uwe D. Hanebeck, “GPU-Accelerated Progressive Gaussian Filtering with Applications to Extended Object Tracking,” in *Proceedings of the 18th International Conference on Information Fusion (Fusion 2015)*, Washington D. C., USA, Jul. 2015, pp. 1038–1045.

[20] Florian Faion, Antonio Zea, Jannik Steinbring, Marcus Baum, and Uwe D. Hanebeck, “Recursive Bayesian Pose and Shape Estimation of 3d Objects Using Transformed Plane Curves,” in *Proceedings of the IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF 2015)*, Bonn, Germany, Oct. 2015.

[21] Tine Lefebvre, Herman Bruyninckx, and Joris De Schutter, *Nonlinear Kalman Filtering for Force-Controlled Robot Tasks*, ser. Springer Tracts in Advanced Robotics. Berlin Heidelberg: Springer, 2005, vol. 19.

[22] Frederik Beutler, Marco F. Huber, and Uwe D. Hanebeck, “Gaussian Filtering using State Decomposition Methods,” in *Proceedings of the 12th International Conference on Information Fusion (Fusion 2009)*, Seattle, USA, Jul. 2009.

[23] Uwe D. Hanebeck, Marco F. Huber, and Vesa 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.

[24] Jannik Steinbring, “Nonlinear Estimation Toolbox.” [Online]. Available: https://bitbucket.org/nonlinearestimation/toolbox

[25] Yaakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York Chichester Weinheim Brisbane Singapore Toronto: Wiley-Interscience, 2001.