Low-rank Tensor Integration for Gaussian Filtering of Continuous Time Nonlinear Systems

Alex A. Gorodetsky, Sertac Karaman, Youssef M. Marzouk

Abstract— **Integration-based Gaussian filters such as unscented, cubature, and Gauss-Hermite filters are effective ways to assimilate data and models within nonlinear systems. Traditionally, these filters have only been applicable for systems with a handful of states due to stability and scalability issues. In this paper, we present a new integration method for scaling quadrature-based filters to higher dimensions. Our approach begins by decomposing the dynamics and observation models into separated, low-rank tensor formats. Once in low-rank tensor format, adaptive integration techniques may be used to efficiently propagate the mean and covariance of the distribution of the system state with computational complexity that is polynomial in dimension and rank. Simulation results are shown on nonlinear chaotic systems with 20 state variables.**

I. INTRODUCTION

In this paper we consider the problem of filtering nonlinear continuous-time dynamics using discrete-time observations. A solution to the filtering problem is critical to many scientific and engineering tasks which evolve in time and for which, often real-time, decisions must be made. Tasks ranging from weather prediction [29] and ocean monitoring [9], [21], [4] to reentry vehicle tracking [6] all rely on assimilating nonlinear models and indirect data.

Computing the optimal solution to the nonlinear filtering problem can be viewed as a Bayesian inference task; however, pursuing this exact solution is intractable for general problem types. Exceptions include the Kalman and Kalman-Bucy filters [16], [17] that are valid for linear dynamical systems and linear observation operators, and the Benes filter [2] for problems in which the nonlinear drift term satisfies certain structures. Numerical solutions that target the full Bayesian posterior by solving Kushner's [20], [19] or Zakai's equation [30] quickly encounter the curse of dimensionality; their expense grows exponentially with dimension. Particle filters [10], [25], [8], [27], [5] offer an alternative approach; however they suffer particle degeneracy in high dimensions. Furthermore, while heuristic methods exist to enable particle filters to scale, they still only provide a Monte-Carlo estimate of the uncertainty.

In this paper we consider a class of filters that *approximate* the solution of the Bayesian inference problem using Gaussian assumptions and attempt to accurate capture both the expectation and covariance. The most well known approach of this type is the extended Kalman Filter, and it works by linearizing dynamics. However, linearization is often

A. Gorodetsky is with Sandia National Laboratories, Albuquerque, NM. alex@alexgorodetsky.com. S. Karaman and Y. Marzouk are with Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. {sertac,ymarz}@mit.edu.

inaccurate for highly nonlinear problems. An alternative class of approaches based on "statistical linearization" has shown better success for nonlinear problems. These methods propagate the mean and covariance of the filtered distributions by evaluating certain multidimensional integrals with respect to Gaussian densities. Examples of these integration-based Gaussian filters include the unscented Kalman filter (UKF) [15] the Gauss-Hermite (GH) Kalman filter [14], the cubature Kalman filter [1], and others [28]. In [28], a unified view of these filters showed how these methods can be viewed as multidimensional integration rules.

As shown in [28], most of the existing integration-based filters use quadrature rules that either become unstable in high dimensions (unscented and cubature filters) or use quadrature rules that grow exponentially with dimension (GH filter). In this paper, we show how to enable a GH-type filter without incurring exponentially growing cost. Our algorithm exploits *low-rank* structure of the dynamical systems and observation operators of interest. Essentially, low-rank functions, those that can be represented as a small sum of products of univariate functions, do not need to be evaluated on full tensor-product quadrature grids to be accurately approximated. Examples of utilizing low-rank structure for multilinear algebraic operations such as integration and computing inner products can be found in the tensor decomposition literature, see e.g. [24], [23], where tensor product grids are compressed using the tensor-train (TT) decomposition.

In this paper, we use a functional version of the tensor-train decomposition [12] to adaptively approximate the dynamical systems and to quickly computing integrals and inner products that are needed for Gaussian filtering. The resulting algorithm scales polynomially with dimension and rank. Our contributions include

- demonstrating low-rank representations of dynamical systems for Gaussian filtering,
- enabling adaptive integration through low-rank function approximation [12] in polynomial time, and
- validating and assessing our formulation on a chaotic system.

Our results indicate a promising new direction for this class of filtering problems.

II. PROBLEM DEFINITION

In this section, we describe continuous-time dynamical systems and Gaussian filtering.

A. Dynamical Systems

The sets of integers and reals are denoted by $\mathbb{Z}$ and $\mathbb{R}$. Their non-negative counterparts are denoted by $\mathbb{Z}_+$ and $\mathbb{R}_+$. The expectation operator is denoted by $\mathbb{E}[\cdot]$, and the covariance operator is denoted by $\mathbb{C}\text{ov}[\cdot, \cdot]$.

We consider a data assimilation problem in which that state $x(t) \in \mathscr{X} \subset \mathbb{R}^d$ evolves in continuous time according to a stochastic differential equation, and observations $y \in \mathscr{Y} \subset \mathbb{R}^{d_y}$ occur at discrete time intervals $t_k \in \mathscr{T} \subset [0, \infty)$. The evolution of this joint system is given by

$$dx = b(t, x)dt + \Sigma dw(t), \tag{1}$$
$$y_k = h(t_k, x(t_k)) + \xi_k, \quad \xi_k \sim \mathcal{N}(0, V), \tag{2}$$

where $w(t) \in \mathbb{R}^{d_w}$ denotes Brownian motion, the drift $b : \mathscr{T} \times \mathscr{X} \to \mathbb{R}^d$ denotes the evolution of the system according to some "known" or baseline dynamics, the diffusion $\Sigma \in \mathbb{R}^{d \times d_w}$ accounts for model error or uncertainty in the state evolution, the observation operator $h : \mathscr{T} \times \mathscr{X} \to \mathbb{R}^{d_y}$ maps the system state to observations, and $\xi_k \in \mathbb{R}^{d_y}$ is Gaussian observation noise with zero mean and covariance V. Note that, for simplicity of presentation, we consider the case where the diffusion is constant across time and space; however, our algorithmic approaches are applicable to the general case as well.

B. Gaussian filtering

We now describe equations for integration-based Gaussian filtering. Gaussian filtering makes the following assumption.

Assumption 1 (Gaussian filtering distributions): Let $0 \leq t \leq T$, and let the dynamics of the state and observations evolve according to (1) and (2). Let $m : [0, T] \to \mathbb{R}^d$ represent the mean and $C : [0, T] \to \mathbb{R}^{d \times d}$ represent the covariance of the filtered distribution of $x(t)$. Gaussian filtering assumes that the initial distribution $x(0) \sim \mathcal{N}(m(0), C(0))$, the forecast distributions for $k = 0, 1, \ldots$

$$x(t)|\boldsymbol{y}_{t_{k+1}} \sim \mathcal{N}(m(t), C(t)), \quad t \in (t_k, t_{k+1}),$$

and the analyzed distributions

$$x(t_{k+1})|\boldsymbol{y}_{t_{k+2}} \sim \mathcal{N}(m(t_{k+1}), C(t_{k+1})),$$

are all Gaussian.

1) Forecast: Under Assumption 1, one can derive the following equations for the propagation of the mean and covariance [26]. Let μ_t denote the measure of a Gaussian random variable with mean $m(t)$ and $C(t)$. The forecast equations for the evolution of the mean and covariance are ordinary differential equations (ODEs)

$$\frac{dm}{dt} = \int b(t, x)\mu_t(dx), \tag{3}$$
$$\frac{dC}{dt} = \Sigma \Sigma^T + \mathbb{C}\text{ov}[x, b(t, x)] + \mathbb{C}\text{ov}[b(t, x), x], \tag{4}$$

for $t_k \leq t \leq t_{k+1}$ where

$$\mathbb{C}\text{ov}[x, b(t, x)] = \int \left(b(t, x) - \frac{dm}{dt} \right) (x - m(t)) \, \mu_t(dx).$$

Let m_{k+1}^- and C_{k+1}^- denote the mean and covariance at t_{k+1}, i.e.,

$$m_{k+1}^- = \int_{t_k}^{t_{k+1}} \frac{dm}{dt} dt \quad \text{and} \quad C_{k+1}^- = \int_{t_k}^{t_{k+1}} \frac{dC}{dt} dt.$$

2) Analysis: The analysis stage updates these means and covariance with a new observation y_{k+1} at time t_{k+1}. Let μ denote the measure of a Gaussian random variable with mean m_{k+1}^- and covariance C_{k+1}^-. The update follows the form of the standard Kalman filter

$$m(t_{k+1}) = m_{k+1}^- + K(y_{k+1} - \eta_{k+1}) \tag{5}$$
$$C(t_{k+1}) = C_{k+1}^- - KSK \tag{6}$$

where

$$S = \mathbb{C}\text{ov}[h(t_{k+1}, x), h(t_{k+1}, x)] + V,$$
$$K = \mathbb{C}\text{ov}[x, h(t_{k+1}, x)] S^{-1},$$

and

$$\eta_{k+1} = \int h(t_k, x)\mu(dx),$$

$$\mathbb{C}\text{ov}[x, h(t_{k+1}, x)] = \int (x - m_{k+1}^-)(h(t_{k+1}, x) - \eta_{k+1}) \mu(dx),$$

$$\mathbb{C}\text{ov}[h(t_{k+1}, x), h(t_{k+1}, x)] =$$
$$\int (h(t_{k+1}, x) - \eta_{k+1})(h(t_{k+1}, x) - \eta_{k+1}) \mu(dx).$$

The primary computational difficulty of using these equations is evaluating the high dimensional integrals with respect to a Gaussian density. As described in the introduction, most of the algorithms for solving these questions can be interpreted choosing a numerical integration schemes [28] for this evaluation. Next, we describe low-rank structure of functions and how such structure allows us to incorporate high-dimensional integration techniques.

III. LOW-RANK REPRESENTATIONS OF FUNCTIONS

In this section, we describe a low-rank representation format that enables adaptive and high-dimensional integration. Let $\mathscr{X}$ be a tensor product space $\mathscr{X} := \mathscr{X}_1 \times \mathscr{X}_2 \times \cdots \times \mathscr{X}_d$, with $\mathscr{X}_i \subset \mathbb{R}$ for $i = 1, \ldots, d$. A *low-rank* approximation of a function $f : \mathscr{X} \to \mathbb{R}$ represents the function as a sum of a few products of univariate functions $f_i^{(j)} : \mathscr{X}_i \to \mathbb{R}$, e.g.,

$$f(x_1, \ldots, x_d) \approx \sum_{j=1}^{R} f_1^{(j)}(x_1) \ldots f_d^{(j)}(x_d).$$

This approximation is called the canonical polyadic (CP) decomposition [3]. The storage complexity of the CP format is linear with dimension and depends on the storage complexity of each $f_j^{(i)}$. For instance, if each input dimension is discretized into n grid points, so that each univariate function is represented by n values (alternatively, if each univariate function is parameterized with n parameters), then the storage complexity of the CP format is $\mathcal{O}(dnR)$, in other words it is linear with dimension, linear with discretization level, and linear with rank R. Thus, for a class of functions whose ranks grow polynomially with dimension, i.e., $R(d) =$

$\mathcal{O}(d^p)$ for some $p \in \mathbb{N}$, polynomial storage complexity is attained.

Contrast this representation of $f(x_1, \ldots, x_d)$ with a lookup table obtained by its evaluation on a tensor-product grid. The storage requirement is $O(n^d)$, it grows exponentially with dimension. Unfortunately such approaches are typically used for accurate integration. For example, tensor product quadrature and sparse-grid quadrature both generate lookup tables that scale exponentially with dimension.

The CP format, however, can be problematic in practice because the problem of determining the canonical rank of a discretized function, or tensor, is NP complete [18], [13], and the problem of finding the best approximation in Frobenius norm for a given rank can be ill-posed [7].

Instead, we propose using a functional variant of the *tensor-train* (TT) decomposition [24], [23], for which the best fixed-rank approximation problem is well posed. In this format, a multivariate function $f : \mathscr{X} \to \mathbb{R}$ is represented as

$$f(x_1, \ldots, x_d) = \sum_{\alpha_0=1}^{r_0} \cdots \sum_{\alpha_d=1}^{r_d} f_1^{(\alpha_0, \alpha_1)}(x_1) \ldots f_d^{(\alpha_{d-1}, \alpha_d)}(x_d),$$
(7)

where $r_i \in \mathbb{Z}_+$ are the *ranks*, with $r_0 = r_d = 1$. For each input coordinate $i = 1, \ldots, d$, it is convenient to group the set of univariate functions $\left\{ f_1^{(\alpha_{i-1}, \alpha_i)} \right\}$ into matrix-valued functions called *cores*:

$$\mathcal{F}_i(x_i) = \begin{bmatrix} f_i^{(1,1)}(x_i) & \cdots & f_i^{(1,r_i)}(x_i) \\ \vdots & \ddots & \vdots \\ f_i^{(r_{i-1},r_i)}(x_i) & \cdots & f_i^{(r_{i-1},r_i)}(x_i) \end{bmatrix}.$$

Thus the evaluation of a function in FT format may equivalently be posed as a sequence of $d-1$ vector-matrix products:

$$f(x_1, \ldots, x_d) = \mathcal{F}_1(x_1) \ldots \mathcal{F}_d(x_d).$$
(8)

A. High-dimensional integration in low-rank tensor formats

Fast integration schemes are critical for filtering. In this section, we describe how the flexibility of the low-rank representation can be exploited to create a fast and adaptive integration scheme with respect to Gaussian measure.

Let $\mu = \mu_1 \times \mu_2 \times \cdots \times \mu_d$ be a product measure with each μ_i denoting a probability measure on $\mathscr{X}_i$. Integration involves integrating each core and performing matrix-vector multiplication $d - 1$ times

$$\int f(x)\mu(dx)$$
(9)

$$= \int \mathcal{F}_1(x_1)\mathcal{F}_2(x_2)\ldots\mathcal{F}_d(x_d)\mu_1(dx_1)\ldots\mu_d(dx_d)$$

$$= \left(\int \mathcal{F}_1(x_1)\mu_1(dx_1) \right) \left(\int \mathcal{F}_2(x_2)\mu_2(dx_2) \right) \times$$

$$\ldots \left(\int \mathcal{F}_d(x_d)\mu_d(dx_d) \right)$$

$$= \Gamma_1 \Gamma_2 \ldots \Gamma_d$$
(10)

where $\Gamma_i = \int \mathcal{F}_i(x_i)\mu_i(dx_i)$ contains entries

$$[\Gamma_i]_{\alpha_{i-1}, \alpha_i} = \int f_i^{(\alpha_{i-1}, \alpha_i)}(x_i)\mu_i(dx_i),$$

for $\alpha_{i-1} \in [1, \ldots, r_{i-1}]$, $\alpha_i \in [1, \ldots, r_i]$.

At this point, we have converted a multidimensional integration problem into the problem of integrating $\mathcal{O}(dr^2)$ univariate functions. Suppose that each univariate function is represented as an expansion of Hermite polynomials

$$f_i^{(\alpha_{i-1}, \alpha_i)}(x_i) = \sum_{\ell=1}^{P} a_{i,\ell}^{(\alpha_{i-1}, \alpha_i)} \operatorname{He}_\ell(x_i),$$

for $\alpha_{i-1} = 1, \ldots, r_{i-1}$, $\alpha_i = 1, \ldots, r_i$, and $a_{i,\ell}^{(\alpha_{i-1}, \alpha_i)} \in \mathbb{R}$, where the Hermite polynomials satisfy the following orthonormality condition

$$\int_{\mathbb{R}} \operatorname{He}_\ell(s) \operatorname{He}_j(s) \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}s^2} ds = \delta_{\ell,j}.$$

Once in this format, the integration of this univariate function is simply the evaluation of its first coefficient

$$\int f_i^{(\alpha_{i-1}, \alpha_i)}(x_i) \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_i^2} dx_i = a_{i,1}^{(\alpha_{i-1}, \alpha_i)}.$$

In other words, when each univariate function of each core is represented as an expansion of Hermite polynomials, integration of each function requires $\mathcal{O}(1)$ operations, and integration of the full multivariate function requires $\mathcal{O}(dr^2)$ operations. Relatedly, the inner product of two functions weighted with a Gaussian measure can be computed using $\mathcal{O}(dr^3)$ operations using the algorithm from [11].

IV. LOW-RANK MODELING OF DYNAMICAL SYSTEMS

In this section, we describe how to model dynamical systems in low-rank format in order to effectively evaluate the integrals described in Section II-B.

A. Overview

To use the low-rank representation and algorithms from Section III the function inputs must be distributed with respect to a tensor product measure. However the filtering equations involve integrals with respect to correlated Gaussian random variables. To this end, we first create a set of decorrelated random variables ν, and then map these variables to x. The new variable $\nu = (\nu_1, \ldots, \nu_d) \in \mathbb{R}^d$ has a standard multivariate normal distribution, i.e., $\nu \sim \mathcal{N}(0, I)$. A mapping from ν to x is linear and given by

$$x = m + C^{1/2}\nu.$$
(11)

Using this mapping, we define a new drift term $\hat{b}(t, \nu)$ as $\hat{b}(t, \nu) = b(t, m(t) + C(t)^{1/2}\nu)$, and a new observation term $\hat{h}(t_k, \nu(t_k)) = h(t, m(t_k) + C(t_k)^{1/2}\nu)$.

Under these transformations, the filtering equations (3)-(6) are all transformed to those with right hand sides containing integrals with respect to ν. For example, the right hand side of (3) becomes

$$\frac{dm}{dt} = (2\pi)^{-d/2} \int \hat{b}(t, \nu) e^{-\frac{1}{2}\nu^T \nu} d\nu.$$
(12)

Evaluating these integrals requires integrating the transformed drift and observation operators. To use low-rank integration schemes, we must approximate these operators

in a low-rank format each time we need to evaluate the RHS of the forecast equations and each time a Bayesian update must be performed. We propose three primary strategies for building these approximations

1) *analytically* derive low-rank representations of the transformed dynamics,
2) build an adaptive approximation of the dynamics by treating them as black-box models, e.g., using the fiber-adaptive cross approximation scheme of [12]., and
3) separate the dynamics into portions to be analytically represented and those to be approximated.

The advantage of the first strategy is computational speed and accuracy since adaptive approximation of functions can take time and many evaluations. However, the advantage of the second strategy is a broad applicability to general forms of dynamics and observation operators. Therefore, the combination of the two schemes is preferable to maintain high accuracy and mitigate the computational expense.

In summary, during every time step of the propagation equations or the Bayesian update equations we perform three steps:

1) decorrelate the state variables through Equation (11),
2) build a low-rank representation of the drift and observation operators, and
3) use low-rank schemes for integration and inner products for equations (3)–(6).

B. Examples

In this section, we demonstrate how several benchmark problems can be modeled in low-rank format.

1) Lorenz 1963: Our first example is the chaotic Lorenz 1963 dynamics with a radar-type nonlinear observation

$$dx_1 = r_1(-x_1 + x_2) + \sigma dw_1(t)$$
$$dx_2 = r_2 x_1 - x_1 x_3 + \sigma dw_2(t)$$
$$dx_3 = -r_3 x_3 + x_1 x_2 + \sigma dw_3(t)$$
$$y_k = \sqrt{\left(x_1 - \frac{1}{2}\right)^2 + x_2^2 + x_3^2} + \xi_k, \quad \xi_k \sim \mathcal{N}(0,1).$$

Low-rank drift: Consider a fixed time t, current filter mean m, and current filter covariance square root P. The decorrelated random variables are ν. Substituting $x = m + P\nu$ into the right-hand-side for the first state we obtain

$$\hat{b}_1(\nu) = r_1 \left[-\left(m_1 + \sum_{j=1}^{3} P_{1j}\nu_j \right) + \left(m_2 + \sum_{j=1}^{3} P_{2j}\nu_j \right) \right],$$
$$= r_1 \left[m_2 m_1 + (m_1 + m_2) \sum_{j=1}^{3} (P_{2j} - P_{1j})\nu_j \right].$$

This linear function has a TT rank-2 representation. Furthermore, this linear function can be captured exactly using Hermite polynomials, and therefore, no adaptive approximation is necessary. Next we consider the equation for the evolution of x_2. This term is quadratic so it can also

be analytically represented in low-rank format. Following a similar derivation, we obtain

$$\hat{b}_2(\nu) = r_2 \left[m_1(1 - m_3) + \sum_{j=1}^{3} ((1 - m_3)P_{1j} - m_1 P_{3j})\nu_j - \right.$$
$$\left. \sum_{i,j=1}^{3} P_{1i}P_{3j}\nu_i\nu_j \right].$$

A quadratic function in TT format has rank $\mathcal{O}(d)$, can be analytically represented in low-rank format, and integrated quickly. The same procedure can be followed for $\hat{b}_3(\nu)$.

Low-rank observation: The observation operator is the square root of a quadratic term, and as a result, there is no exact low-rank representation. In this case, a black-box approximation procedure can be used to approximate it in low-rank form. In this paper, we use the fiber-adaptive cross-approximation procedure described in [12]. This procedure requires $\mathcal{O}(dpr^2)$ evaluations of $\hat{h}$ to build the approximation. In this case p denotes the maximum number of Hermite polynomials used within any individual $f_i^{(\alpha_{i-1}, \alpha_i)}$. This number is equivalent to the order of a tensor-product quadrature with p nodes for each dimension.

2) Lorenz 1996: Our next example is the chaotic Lorenz 1996 [22] dynamics frequently used to benchmark filtering problems. This model represents an atmospheric variable at equally spaced poitns around a circle of constant latitude.

$$dx_i = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F + \sigma dw_i(t),$$
$$x_0 = x_d \quad x_{-1} = x_{d-1} \quad x_{d+1} = x_1$$

for $i = 1, \ldots, d$. This equation has an external forcing term F, and different values can lead to chaotic or fully turbulent behaviors. Observations are made of every other state

$$y_k^j = x_{2j-1}(t_k) + \xi_k, \quad \xi_k \sim \mathcal{N}(0,1), \quad \text{for } j = 1, \ldots d/2$$

Note that the right-hand-side of the dynamics contain only quadratic terms and therefore the dynamics of the uncorrelated system are also quadratic with rank bounded by d. The observation operator is linear both in the correlated variables and in the uncorrelated variables. Overall, no adaptive approximation is needed and the low-rank formulation of the dynamics can be obtained analytically.

C. Comparison with alternate Gaussian filters

In this section, we discuss the advantages and disadvantages of the proposed scheme as compared with other numerical integration based filters. As discussed in [28], a majority of Gaussian filters, aside from the GH filter, become unstable as the dimension increases, and the disadvantage of the stable GH filter is its exponential growth in computational complexity with dimension. Here, we show that the computational cost of the low-rank approximation scheme is moderately larger, but still less than exponential, than for unscented or cubature filters. For reference, consider that an unscented Kalman filter of third order accuracy in both the mean and the covariance requires $2d + 1$ evaluations of the drift and observation operators.

Generating a low-rank approximation of a black-box function in FT format requires $\mathcal{O}(dmr^2)$ operations [11], where m is the number of evaluations of f needed to generate a univariate approximation of $f_i^{(\alpha_{i-1},\alpha_i)}$. For third order accuracy we need to at least $m = 4$ nodes in our quadrature rule since an m-point Gaussian quadrature rule is accurate for polynomial orders up to $2m - 1$, and the function approximation algorithm we use utilizes a pseudospectral approximation algorithm that needs to compute the weighted inner product $\langle f(x)\,\mathrm{He}\,l\rangle$ for certain univariate fibers through f. If f is third order, then we need to integrate a sixth order integrand accurately, requiring $m = 4$ nodes. The computational complexity is therefore $\mathcal{O}(4dr^2)$, or $\mathcal{O}(r^2)$ larger than the unscented filter. Thus, we pay a computational price for stability that scales linearly with dimension.

V. Numerical Results

We now compare the stability of our filter with the UKF and demonstrate it's performance for the Lorenz 96 system with $d = 20$ and $F = 8$ to stay within the chaotic regime. We fix $\sigma = 0.25$ and a measurement noise variance of one.

A. Stability comparison

The stability results in [28] are based on discrete-time filtering. In continuous-time filtering, additional care must be taken within the time-stepping scheme for the filter dynamics; large step sizes often impose additional stability issues. Empirically, we notice that the instability in our proposed filter is significantly delayed compared with the continuous-discrete UKF (CDUKF) [26]. Suppose we use a time step of 0.01s and Runge-Kutta-4 for filter propagation, and make observations every $\Delta t = 0.25$s. Figure 1 shows that the UKF diverges before even the first data-point is taken. The low-rank filter takes five times longer to diverge. This problem is particular susceptible to instability because we do not initialize the filter on the attractor.

B. Lorenz 1996

Next, we filter the Lorenz 1996 dynamics using an adaptive RKF45 scheme to enhance stability. This problem would require 3^{20} quadrature nodes, or roughly 3.5 billion samples, for a fifth-order-accurate GH filter. Observations of every other variable are taken every $\Delta t = 0.1$ seconds. The traces for a subset of states are provided in Figure 2.

Figure 2 shows that the mean tracks the system state for both observed and unobserved variables, but the uncertainty is smaller for the observed variables. Furthermore, tracking and uncertainty are noticeably worse when $t < 2$ for many of the variables due to the fact that no burn-in was used before filtering is started.

VI. Conclusion

We have demonstrated an approach that leverages low-rank structure of nonlinear differential equations to perform efficient and adaptive Gaussian filtering. The resulting algorithm enables integration-based Gaussian filtering to be useful for higher dimensional models than the stable Gauss-Hermite filter. As a result, our algorithm provides a promising alternative to traditional particle and ensemble type filters for problems with tens of dimensions.

Future work will further analyze how approximations of the RHS of the mean and covariance propagation ODEs and the numerical time integration scheme affects filter stability. Another direction will seek to combine dimension reduction methods with Gaussian filtering. Since our methodology is now applicable to problems with tens of dimensions, this combination can potentially enable large-scale filtering for a wider number of applications.

References

[1] I. Arasaratnam and S. Haykin. Cubature Kalman filters. *IEEE Transactions on Automatic Control*, 54(6):1254–1269, June 2009.

[2] V. E. Beneš. Exact finite-dimensional filters for certain diffusions with nonlinear drift. *Stochastics*, 5(1-2):65–92, June 1981.

[3] J. Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319, September 1970.

[4] James A Carton and Benjamin S Giese. A reanalysis of ocean climate using Simple Ocean Data Assimilation (soda). *Monthly Weather Review*, 136(8):2999–3017, 2008.

[5] Alexandre Chorin, Matthias Morzfeld, and Xuemin Tu. Implicit particle filters for data assimilation. *Communications in Applied Mathematics and Computational Science CAMCoS*, 5(2):221–240, November 2010.

[6] Peter J. Costa. Adaptive model architecture and extended Kalman-Bucy filters. *IEEE Transactions on Aerospace and Electronic Systems*, 30(2):525–533, 1994.

[7] V. De Silva and L. Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, January 2008.

[8] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[9] Michael Ghil and Paola Malanotte-Rizzoli. Data assimilation in meteorology and oceanography. *Advances in Geophysics*, 33:141–266, 1991.

[10] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, Institution of Engineering and Technology (IET), 1993.

[11] A.A. Gorodetsky, S. Karaman, and Y.M. Marzouk. Function-Train: A continuous analogue of the Tensor-Train decomposition. *arXiv preprint arXiv:1510.09088*, 2015.

[12] Alex Gorodetsky, Sertac Karaman, and Youssef Marzouk. Efficient high-dimensional stochastic optimal motion control using Tensor-Train decomposition. In *Robotics: Science and Systems XI*, Rome, Italy, July 2015. Robotics: Science and Systems Foundation.

[13] Johan Håstad. Tensor rank is NP-complete. *Journal of Algorithms*, 11(4):644–654, December 1990.
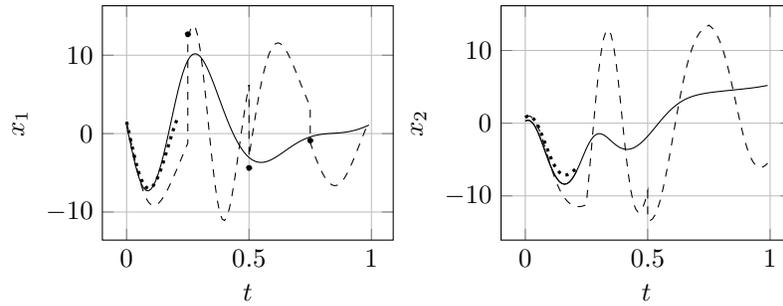
Fig. 1: Filtering instability comparison between the CDUKF and the proposed low-rank filter. True trajectory (solid black), the low-rank filter mean (dashed line), and the CDUKF mean (dotted), are shown for the first two state variables. Data is shown with black dots. The CDUKF breaks down after 0.2 seconds, the low-rank filter breaks down after 1 second.
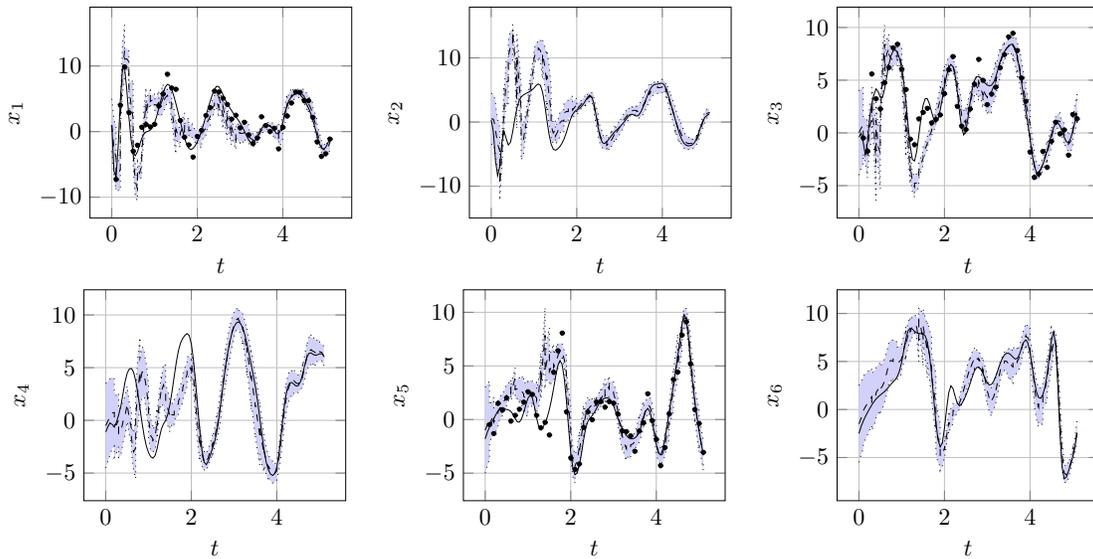

Fig. 2: Traces of the mean (dashed line), two times standard deviation (shaded region), observations (black-dots) and system trajectory (solid line) for specific system states for the Lorenz96 chaotic system.

[14] K. Ito and K. Xiong. Gaussian filters for nonlinear filtering problems. *IEEE Transactions on Automatic Control*, 45(5):910–927, May 2000.

[15] Simon J. Julier and Jeffrey K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In Ivan Kadar, editor, *Signal Processing, Sensor Fusion, and Target Recognition VI*. SPIE-Intl Soc Optical Eng, July 1997.

[16] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, March 1960.

[17] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83(1):95–108, March 1961.

[18] J. B. Kruskal. *Multiway data analyis*, chapter Rank, decomposition, and uniqueness for 3-way and N-way arrays. Amsterdam, North-Holland, 1989.

[19] H. J. Kushner. Dynamical equations for optimal nonlinear filtering. *Journal of Differential Equations*, 3(2):179–190, April 1967.

[20] Harold J. Kushner. On the differential equations satisfied by conditional probablitity densities of Markov processes, with applications. *Journal of the Society for Industrial and Applied Mathematics Series A Control*, 2(1):106–119, January 1964.

[21] P. F. J Lermusiaux. Estimation and study of mesoscale variability in the Strait of Sicily. *Dynamics of Atmospheres and Oceans*, 29(2):255–303, 1999.

[22] Edward N Lorenz. Predictability: A problem partly solved. In *Proc.*

[23] I. V. Oseledets. Tensor-Train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, January 2011.

Seminar on predictability, volume 1, 1996.

[24] Ivan V. Oseledets and Eugene Tyrtyshnikov. TT-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*, 432(1):70–88, January 2010.

[25] Michael K. Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, June 1999.

[26] Simo Sarkka. On unscented Kalman filtering for state estimation of continuous-time nonlinear systems. *IEEE Transactions on Automatic Control*, 52(9):1631–1641, September 2007.

[27] Peter Jan van Leeuwen. Particle filtering in geophysical systems. *Monthly Weather Review*, 137(12):4089–4114, December 2009.

[28] Y. Wu, D. Hu, M. Wu, and X. Hu. A numerical-integration perspective on Gaussian filters. *IEEE Transactions o Signal Processing*, 54(8):2910–2921, 2006.

[29] Ming Xue, Donghai Wang, Jidong Gao, Keith Brewster, and Kelvin K Droegemeier. The Advanced Regional Prediction System (ARPS), storm-scale numerical weather prediction and data assimilation. *Meteorology and Atmospheric Physics*, 82(1):139–170, 2003.

[30] Moshe Zakai. On the optimal filtering of diffusion processes. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 11(3):230–243, 1969.