A Serial Approach to Handling High-Dimensional Measurements in the Sigma-Point Kalman Filter

Colin McManus and Timothy D. Barfoot
Institute for Aerospace Studies
University of Toronto
Toronto, Canada
Email: {colin.mcmanus,tim.barfoot}@utoronto.ca

Abstract—Pose estimation is a critical skill in mobile robotics and is often accomplished using onboard sensors and a Kalman filter estimation technique. For systems to run online, computational efficiency of the filter design is crucial, especially when faced with limited computing resources. In this paper, we present a novel approach to serially process high-dimensional measurements in the Sigma-Point Kalman Filter (SPKF), in order to achieve a low computational cost that is linear is the measurement dimension. Although the concept of serially processing measurements has been around for quite some time in the context of the Extended Kalman Filter (EKF), few have considered this approach with the SPKF. At first glance, it may be tempting to apply the SPKF update step serially. However, we prove that without re-drawing sigma points, this 'naive' approach cannot guarantee the positive-definiteness of the state covariance matrix (not the case for the EKF). We then introduce a novel method for the Sigma-Point Kalman Filter to process high-dimensional, uncorrelated measurements serially that is algebraically equivalent to processing the measurements in parallel, but still achieves a computational cost linear in the measurement dimension.

I. INTRODUCTION

For nonlinear estimation, the Extended Kalman Filter (EKF) is perhaps the most common technique that is used in practice [8]. However, it is widely understood that the first-order linearization used in the EKF can lead to issues in both accuracy and stability [9, 25].

Derivative-free Kalman filters, such as the Divided Difference Filter (DDF) [16], the Central Difference Filter (CDF) [4], the Unscented Kalman Filter (UKF) [9], and the Iterated Sigma-Point Kalman Filter (ISPKF) [20] offer an attractive alternative to the EKF as they are theoretically more accurate and easier to implement due to their derivative-free nature. Collectively, these filters have been referred to as Sigma-Point Kalman Filters (SPKF) [21], or Linear Regression Kalman Filters (LRKF) [12], because they linearize a function by way of a statistical linear regression through a specific sampling of points [12]. This paper focuses on the most common of the four types of SPKF, namely, the UKF. As an aside, although SPKF is technically an umbrella term, we prefer to use the term SPKF synonymously with UKF, as we feel it is less colloquial.

For many practical field robotic systems, filter efficiency is a critical requirement due to limited computational resources. As an example, we refer to the autonomous underground tram-

Fig. 1. Autonomous underground tramming vehicle used by Marshall et al. [14]; equipped with a SICK laser rangefinder and wheel/shaft encoders. We consider the same localization scenario as Marshall et al., which is a vehicle using a planar rangefinder to localize against a known map. As was the case for this system, we consider localization using the SPKF. Image credit: Atlas Copco.

ming system developed by Marshall et al. [14] (see Figure 1), which uses a SICK laser rangefinder and the SPKF to localize against a known map. Since this system functioned in an underground, infrastructureless mine environment, the vehicle could only use its onboard sensors for motion estimation. However, due to computational constraints, they were only able to process a small portion of the incoming data-stream from their SICK laser.

With the ever-increasing data acquisition rates of light detection and ranging (lidar) sensors, such as a Velodyne lidar[1], developing efficient estimation techniques are crucial for online performance. Motivated by this need, we have been investigating serially processing high-dimensional measurements as opposed to processing the measurements in parallel, which requires a large matrix inversion in the measurement-update step. Although the concept of serial processing has been around for quite some time[2], few have considered serial processing with the SPKF.

For the EKF, it is well known that parallel processing and serial processing will produce an algebraically equivalent state covariance matrix, so long as the state estimate is not being re-

---

[1]High Definition Lidar HDL-64E S2, *Data Sheet*, Veldoyne Lidar Inc.
[2]According to Gelb et al. [3], the idea of serial processing instead of parallel processing was recognized as early as 1962.

linearized after each measurement update [3][3]. Intuitively, one would expect the same properties to hold true for the SPKF. However, as we will show, 'naively' applying the standard SPKF update step for serial processing cannot guarantee a positive-definite state covariance and will not result in the same estimate. In response to this rather counterintuitive result, we derive a new form of the measurement-update step in the SPKF, in order to process high-dimensional, uncorrelated measurements serially. This new form is algebraically identical to processing the measurements in parallel, but maintains a cost that is linear in the measurement dimension.

We present simulations comparing the computational time of our serial SPKF (S-SPKF) with the canonical parallel processing form (P-SPKF) and demonstrate the significant efficiency improvements with our method. In addition, to verify the algebraic equivalency between the S-SPKF and P-SPKF, we implemented a simulation representing the scenario described in Marshall et al. [14], which is a robot localizing against a known map using a 2D planar laser as the primary sensor. It is worth noting that this particular problem is not well suited for the EKF because it involves a generative sensor model that does not have a closed-form analytical expression, and thus requires numerical Jacobians. Nonetheless, as the EKF is such a common estimator, we have included it in our comparison.

II. REVIEW

Julier et al. [9] introduced the SPKF as an alternative to the EKF, demonstrating its superior performance and ease of implementation for nonlinear estimation problems. Initially, the SPKF was presented in a form in which the noise associated with the transition and observation models was added outside the nonlinearity, allowing one to simply add the noise covariance matrices to the state covariance updates. Later, Julier and Uhlmann [6] presented a more general form where the process and observation noises are stacked with the state, requiring the computation of a larger number of sigma-points; obviously this is not ideal for applications with large measurement dimensions. To address this issue, Julier and Uhlmann [7] introduced a smaller set of sigma-points, called *simplex sigma-points*, which are chosen to minimize the third order moments of the probability density function (PDF). They showed that the dimensionality of the problem can be reduced from $2N+1$ to $N+2$ sigma-points. A more stable version of this algorithm was later presented in [5], which constrains the simplex sigma-points to be equidistant from the origin. Quine [17] derived a similar reduced SPKF formulation, but one that used the minimal set of $N+1$ sigma-points, which can only capture the mean and covariance of the PDF.

In [22], van der Merwe and Wan presented the Square Root Unscented Kalman Filter (SR-UKF), which uses a series of linear algebra techniques such as QR decomposition, Cholesky factor updating, and efficient least squares, in order to avoid the need to refactor the state covariance matrix at each time step. In addition to these efficiency gains, the SR-UKF also presents numerical stability benefits as it ensures that the state covariance is always positive-definite.

The aforementioned methods aim to improve efficiency by either reducing the number of sigma-points or by using efficient algebraic simplifications. However, as for processing measurements serially or in parallel, there is little mentioned in the literature pertaining to the SPKF. Duan et al. [2] and later Lei and Han [13], introduced a SPKF serial processing approach for the problem of radar target tracking with range measurements. Their method was analogous to serial processing with the EKF and re-linearizing after incorporating each measurement. In the case of the SPKF, this means re-drawing sigma-points and estimating a new mean, covariance and associated statistical Jacobians. They argue in favour of re-drawing sigma-points because of the success with the EKF counterpart in target tracking applications [15, 18]. However, we note that improved accuracy resulting from re-linearizing is not guaranteed for all systems, as was the case for our simulations (this is discussed in more detail later). Asl and Pourtakdoust [1] also considered the problem of radar target tracking and developed a serial processing approach that uses a UD covariance factorization for additional efficiency gains. Unlike Duan et al., the sigma-points were not re-drawn after each measurement update. Our work shares similarities to Asl and Pourtakdoust [1] as we do not re-draw sigma-points after each measurement update and we also use a factorization technique to re-write the canonical SPKF equations in a more efficient form. However, their method is not algebraically equivalent to the parallel case and uses a UD factorization whereas we chose to use a statistical Jacobian factorization.

In the common case of uncorrelated measurements, the information filter is well suited for serial processing, as its form naturally results in a linear summation of the innovation terms. Consequently, the Unscented Information Filters (UIF) by Vercauteren et al. [24] and Lee et al. [10, 11] can be considered other serial processing methods for the SPKF. However, instead of maintaining an estimate of the information vector, we derive an alternative serial form of the measurement-update step that directly computes the mean, making our form more efficient. This is accomplished by expressing the measurement-update step in terms of its statistical Jacobian factors and performing a series of manipulations to decouple the state and measurement terms. Loosely speaking, we have developed a hybrid of the canonical SPKF measurement-update step and the UIF measurement-update step that is particularly helpful for the case of processing measurements serially.

III. THE SIGMA-POINT KALMAN FILTER

This section presents the canonical SPKF algorithm in its entirety. Consider the following nonlinear, dynamic, discrete-

---

[3]This is only true for one measurement update step, as the mean would not be algebraically identical.

time transition model, $\mathbf{h}(\cdot)$, and measurement model, $\mathbf{g}(\cdot)$,

$$
\begin{aligned}
\mathbf{x}_k &= \mathbf{h}\left(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k\right), & \mathbf{w}_k &\sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}_k\right) \\
\mathbf{y}_k &= \mathbf{g}\left(\mathbf{x}_k\right) + \mathbf{n}_k, & \mathbf{n}_k &\sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}_k\right)
\end{aligned}
$$

where $\mathbf{x}_k$ is the state, $\mathbf{y}_k$ is our sensor measurement, $\mathbf{u}_k$ is a control input, $\mathbf{w}_k$ is process noise, $\mathbf{Q}_k$ is the process noise covariance matrix, $\mathbf{n}_k$ is measurement noise, $\mathbf{R}_k$ is the measurement noise covariance matrix, and k denotes the time step. The goal is to take a prior belief of the state and covariance, $\left\{\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{P}}_{k-1}\right\}$, and estimate the posterior belief $\left\{\hat{\mathbf{x}}_k, \hat{\mathbf{P}}_k\right\}$, according to a two stage process: the *prediction step* and the *correction step*.

A. Prediction $\left\{\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{P}}_{k-1}\right\} \to \left\{\hat{\mathbf{x}}_k^-, \hat{\mathbf{P}}_k^-\right\}$

1) Augment our state with the motion noise[4]:

$$
\mathbf{z} := \begin{bmatrix} \hat{\mathbf{x}}_{k-1} \\ \mathbf{0} \end{bmatrix}, \qquad \mathbf{Y} := \begin{bmatrix} \hat{\mathbf{P}}_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \end{bmatrix} \qquad (1)
$$

We let $L = \dim \mathbf{z} = N + P$, where N is the state dimension and P is the process noise dimension.

2) Convert $\{\mathbf{z}, \mathbf{Y}\}$ to a sigma-point representation:

$$
\begin{aligned}
\mathbf{SS}^T &:= \mathbf{Y} && \text{(Cholesky decomposition)} \\
\mathcal{Z}_0 &:= \mathbf{z} \\
\mathcal{Z}_i &:= \mathbf{z} + \sqrt{L+\kappa}\,\mathrm{col}_i\mathbf{S} \\
\mathcal{Z}_{i+L} &:= \mathbf{z} - \sqrt{L+\kappa}\,\mathrm{col}_i\mathbf{S}
\end{aligned} \qquad i = 1 \ldots L.
$$

where κ is a scaling parameter that affects fourth and higher-order moments of the PDF.

3) Unstack each sigma-point into state and motion noise,

$$
\mathcal{Z}_i =: \begin{bmatrix} \mathcal{X}_{i,k-1} \\ \mathcal{W}_{i,k} \end{bmatrix},
$$

and then pass through the nonlinear motion model:

$$
\mathcal{X}_{i,k}^- := \mathbf{h}\left(\mathcal{X}_{i,k-1}, \mathbf{u}_k, \mathcal{W}_{i,k}\right) \qquad i = 0 \ldots 2L.
$$

4) Recombine the transformed sigma-points into the predicted belief, $\left\{\hat{\mathbf{x}}_k^-, \hat{\mathbf{P}}_k^-\right\}$, according to

$$
\begin{aligned}
\hat{\mathbf{x}}_k^- &:= \sum_{i=0}^{2L} \beta_i \mathcal{X}_{i,k}^-, \\
\hat{\mathbf{P}}_k^- &:= \sum_{i=0}^{2L} \beta_i \left(\mathcal{X}_{i,k}^- - \hat{\mathbf{x}}_k^-\right)\left(\mathcal{X}_{i,k}^- - \hat{\mathbf{x}}_k^-\right)^T,
\end{aligned}
$$

where $\beta_i = \begin{cases} \frac{\kappa}{L+\kappa} & i = 0 \\ \frac{1}{2}\frac{\kappa}{L+\kappa} & \text{otherwise} \end{cases}$

---

[4]For notational convenience, we use $\mathbf{0}$ rather loosely, since the dimensions are not the same for each matrix.

B. Correction $\left\{\hat{\mathbf{x}}_k^-, \hat{\mathbf{P}}_k^-\right\} \to \left\{\hat{\mathbf{x}}_k, \hat{\mathbf{P}}_k\right\}$

1) Pass each sigma-point through the nonlinear observation model exactly:

$$
\mathcal{Y}_{i,k} := \mathbf{g}\left(\mathcal{X}_{i,k}^-\right) \qquad i = 0 \ldots 2L.
$$

2) Compute the predicted measurement and innovation covariance, according to

$$
\begin{aligned}
\hat{\mathbf{y}}_k &:= \sum_{i=0}^{2L} \beta_i \mathcal{Y}_{i,k}, \\
\mathbf{V}_k &:= \sum_{i=0}^{2L} \beta_i \left(\mathcal{Y}_{i,k} - \hat{\mathbf{y}}_k\right)\left(\mathcal{Y}_{i,k} - \hat{\mathbf{y}}_k\right)^T + \mathbf{R}_k.
\end{aligned}
$$

3) Build the state-measurement covariance and Kalman gain according to

$$
\begin{aligned}
\mathbf{U}_k &:= \sum_{i=0}^{2L} \beta_i \left(\mathcal{X}_{i,k}^- - \hat{\mathbf{x}}_k^-\right)\left(\mathcal{Y}_{i,k} - \hat{\mathbf{y}}_k\right)^T, \\
\mathbf{K}_k &:= \mathbf{U}_k \mathbf{V}_k^{-1}.
\end{aligned}
$$

4) Compute the posterior belief, $\left\{\hat{\mathbf{x}}_k, \hat{\mathbf{P}}_k\right\}$, according to

$$
\hat{\mathbf{x}}_k := \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{y}_k - \hat{\mathbf{y}}_k\right), \quad \hat{\mathbf{P}}_k := \hat{\mathbf{P}}_k^- - \mathbf{K}_k \mathbf{U}_k^T.
$$

IV. SERIAL PROCESSING

In this section we will present our new serial processing approach for the SPKF. But first, we begin by proving that naive serial processing using the standard SPKF measurement-update step cannot guarantee the positive-definiteness of the state covariance matrix (not the case for the EKF). Since the remaining section is concerned with the correction step at a single time-step, we will omit the subscript k for convenience.

In the ensuing discussion, we assume that we acquire an uncorrelated high-dimensional measurement with a block-diagonal noise covariance matrix:

$$
\mathbf{y} = \begin{bmatrix} \mathbf{y}_1^T & \mathbf{y}_2^T & \cdots & \mathbf{y}_M^T \end{bmatrix}^T, \qquad \mathbf{R} = \mathrm{diag}\left(\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_M\right),
$$

where $\mathbf{y}_i \in \mathbb{R}^J$ and $\mathbf{R}_i \in \mathbb{R}^{J \times J}$.

A. Parallel Processing Approach

Following the correction step shown in section III, the innovation covariance is given by the following

$$
\mathbf{V} = \sum_{i=0}^{2L} \beta_i \left(\mathcal{Y}_i - \hat{\mathbf{y}}\right)\left(\mathcal{Y}_i - \hat{\mathbf{y}}\right)^T + \mathbf{R} = \mathcal{Y}\mathcal{Y}^T + \mathbf{R},
$$

where we have conveniently defined

$$
\mathcal{Y} := \begin{bmatrix} \sqrt{\beta_0}\left(\mathcal{Y}_0 - \hat{\mathbf{y}}\right) \ldots \sqrt{\beta_{2L}}\left(\mathcal{Y}_{2L} - \hat{\mathbf{y}}\right) \end{bmatrix} =: \begin{bmatrix} \mathcal{Y}_1 \\ \vdots \\ \mathcal{Y}_M \end{bmatrix}.
$$

We can then see that the innovation covariance matrix will be dense with these predicted measurement error terms:

$$
\mathbf{V} = \begin{bmatrix} \mathcal{Y}_1\mathcal{Y}_1^T + \mathbf{R}_1 & \cdots & \mathcal{Y}_1\mathcal{Y}_M^T \\ \vdots & \ddots & \vdots \\ \mathcal{Y}_M\mathcal{Y}_1^T & \cdots & \mathcal{Y}_M\mathcal{Y}_M^T + \mathbf{R}_M \end{bmatrix} \qquad (2)
$$

The 'state-measurement' covariance is given by

$$\mathbf{U} = \boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}^T,$$

where,

$$\boldsymbol{\mathcal{X}} := \left[\sqrt{\beta_0} \left(\mathcal{X}_0^- - \hat{\mathbf{x}}^- \right) \ldots \sqrt{\beta_{2L}} \left(\mathcal{X}_{2L}^- - \hat{\mathbf{x}}^- \right) \right]. \quad (3)$$

The resulting posterior is then

$$\begin{aligned}
\hat{\mathbf{x}} &= \hat{\mathbf{x}}^- + \boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}^T \mathbf{V}^{-1} \left(\mathbf{y} - \hat{\mathbf{y}} \right), \\
\hat{\mathbf{P}} &= \boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{X}}^T - \boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}^T \mathbf{V}^{-1} \boldsymbol{\mathcal{Y}}\boldsymbol{\mathcal{X}}^T,
\end{aligned} \quad (4)$$

where $\mathbf{V}$ is given by (2) and we note that $\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{X}}^T = \hat{\mathbf{P}}^-$. So far we have merely rewritten the SPKF update step using its statistical Jacobian form, which will prove useful in the remainder of this section.

B. 'Naive' Serial Processing Approach

By assumption, the $\mathbf{y}_i$ measurements are uncorrelated; we can therefore attempt to process them serially. The first measurement is processed as follows, where we use the notation $(\cdot)^i$ to denote an estimate at iteration i:

$$\begin{aligned}
\mathbf{V}^1 &= \boldsymbol{\mathcal{Y}}_1\boldsymbol{\mathcal{Y}}_1^T + \mathbf{R}_1 \\
\mathbf{K}^1 &= \boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}_1^T \left(\boldsymbol{\mathcal{Y}}_1\boldsymbol{\mathcal{Y}}_1^T + \mathbf{R}_1 \right)^{-1} \\
\hat{\mathbf{x}}^1 &= \hat{\mathbf{x}}^- + \boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}_1^T \left(\boldsymbol{\mathcal{Y}}_1\boldsymbol{\mathcal{Y}}_1^T + \mathbf{R}_1 \right)^{-1} \left(\mathbf{y}_1 - \hat{\mathbf{y}}_1 \right) \\
\hat{\mathbf{P}}^1 &= \boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{X}}^T - \boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}_1^T \left(\boldsymbol{\mathcal{Y}}_1\boldsymbol{\mathcal{Y}}_1^T + \mathbf{R}_1 \right)^{-1} \boldsymbol{\mathcal{Y}}_1\boldsymbol{\mathcal{X}}^T
\end{aligned}$$

For the second measurement, we have

$$\begin{aligned}
\hat{\mathbf{x}}^2 &= \hat{\mathbf{x}}^1 + \boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}_2^T \left(\boldsymbol{\mathcal{Y}}_2\boldsymbol{\mathcal{Y}}_2^T + \mathbf{R}_2 \right)^{-1} \left(\mathbf{y}_2 - \hat{\mathbf{y}}_2 \right), \\
\hat{\mathbf{P}}^2 &= \hat{\mathbf{P}}^1 - \boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}_2^T \left(\boldsymbol{\mathcal{Y}}_2\boldsymbol{\mathcal{Y}}_2^T + \mathbf{R}_2 \right)^{-1} \boldsymbol{\mathcal{Y}}_2\boldsymbol{\mathcal{X}}^T.
\end{aligned}$$

Continuing this process and stacking everything in matrix form, we see that the posterior is given by

$$\begin{aligned}
\hat{\mathbf{x}} &= \hat{\mathbf{x}}^- + \boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}^T \mathbf{V}^{-1} \left(\mathbf{y} - \hat{\mathbf{y}} \right), \\
\hat{\mathbf{P}} &= \boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{X}}^T - \boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}^T \mathbf{V}^{-1} \boldsymbol{\mathcal{Y}}\boldsymbol{\mathcal{X}}^T,
\end{aligned} \quad (5)$$

where

$$\mathbf{V} = \text{diag} \left(\boldsymbol{\mathcal{Y}}_1\boldsymbol{\mathcal{Y}}_1^T + \mathbf{R}_1, \ldots, \boldsymbol{\mathcal{Y}}_M\boldsymbol{\mathcal{Y}}_M^T + \mathbf{R}_M \right). \quad (6)$$

Comparing (6) with (2), we immediately see that 'naively' processing the measurements serially results in the loss of all the off-diagonal terms in the innovation covariance, yielding a different estimate. In addition, it turns out that this method of serial processing cannot guarantee a positive-definite state covariance matrix. We now introduce the following theorems that will be useful in analyzing the stability of the parallel and serial processing methods.

Theorem 1. *Let* $\boldsymbol{\mathcal{X}} := [\mathbf{0}, \mathbf{B}, -\mathbf{B}]$, *such that* $\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{X}}^T := \hat{\mathbf{P}}$, *where* $\hat{\mathbf{P}} \in \mathbb{R}^{N \times N}$, *and* $\hat{\mathbf{P}} > 0$. *If* $\mathbf{A} \in \mathbb{R}^{M \times M}$ *and* $\mathbf{A} > 0$ *then* $\boldsymbol{\mathcal{X}}\mathbf{A}\boldsymbol{\mathcal{X}}^T > 0$, *else if* $\mathbf{A} \leq 0$ *then* $\boldsymbol{\mathcal{X}}\mathbf{A}\boldsymbol{\mathcal{X}}^T \leq 0$.

Proof: Let

$$\mathbf{v} := \begin{bmatrix} \mathbf{0} \\ \mathbf{u} \\ -\mathbf{u} \end{bmatrix}, \qquad \mathbf{A} := \begin{bmatrix} a_{11} & \mathbf{a}_{21}^T & \mathbf{a}_{31}^T \\ \mathbf{a}_{21} & \mathbf{A}_{22} & \mathbf{A}_{32}^T \\ \mathbf{a}_{31} & \mathbf{A}_{32} & \mathbf{A}_{33} \end{bmatrix},$$

where $\mathbf{u} \neq \mathbf{0}$ and $\mathbf{A}$ is an $M \times M$ real, symmetric matrix. Recalling the definition of a positive definite matrix, we note that $\mathbf{A}$ is positive definite if $\mathbf{v}^T \mathbf{A} \mathbf{v} > 0$, $\forall \ \mathbf{v} \in \mathbb{R}^M, \mathbf{v} \neq \mathbf{0}$. Using the above defined quantities, we have $\mathbf{v}^T \mathbf{A} \mathbf{v} = \mathbf{u}^T \left(\mathbf{A}_{22} - \mathbf{A}_{32}^T - \mathbf{A}_{32} + \mathbf{A}_{33} \right) \mathbf{u}$, where we can see that

$$\begin{aligned}
\mathbf{A} > 0 &\implies \mathbf{A}_{22} - \mathbf{A}_{32}^T - \mathbf{A}_{32} + \mathbf{A}_{33} > 0, \quad (7) \\
\mathbf{A} \leq 0 &\implies \mathbf{A}_{22} - \mathbf{A}_{32}^T - \mathbf{A}_{32} + \mathbf{A}_{33} \leq 0.
\end{aligned}$$

Now we define a new quantity, $\boldsymbol{\mathcal{X}} := [\mathbf{0}, \mathbf{B}, -\mathbf{B}]$, where $\mathbf{B}$ is an invertible matrix that is proportional to the Cholesky factor of $\hat{\mathbf{P}}$. Taking note of the fact that

$$\boldsymbol{\mathcal{X}}\mathbf{A}\boldsymbol{\mathcal{X}}^T = \mathbf{B} \left(\mathbf{A}_{22} - \mathbf{A}_{32}^T - \mathbf{A}_{32} + \mathbf{A}_{33} \right) \mathbf{B}^T,$$

we see that

$$\mathbf{w}^T \left(\boldsymbol{\mathcal{X}}\mathbf{A}\boldsymbol{\mathcal{X}}^T \right) \mathbf{w} = \mathbf{w}^T \mathbf{B} \left(\mathbf{A}_{22} - \mathbf{A}_{32}^T - \mathbf{A}_{32} + \mathbf{A}_{33} \right) \mathbf{B}^T \mathbf{w}.$$

Since $\mathbf{B}$ is invertible, we know that $\mathbf{w} \neq \mathbf{0} \Rightarrow \mathbf{B}^T \mathbf{w} \neq \mathbf{0}$. Thus, using (7), we can conclude that

$$\begin{aligned}
\boldsymbol{\mathcal{X}}\mathbf{A}\boldsymbol{\mathcal{X}}^T > 0 &\quad \text{if } \mathbf{A} > 0 \\
\boldsymbol{\mathcal{X}}\mathbf{A}\boldsymbol{\mathcal{X}}^T \leq 0 &\quad \text{if } \mathbf{A} \leq 0
\end{aligned}$$

$\blacksquare$

Theorem 2. *If* $\mathbf{A} > 0$, $\mathbf{B} \geq 0$ *and* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N \times N}$, *then* $\mathbf{A} + \mathbf{B} > 0$.

Proof: Trivially, we have that
$$\mathbf{x}^T (\mathbf{A} + \mathbf{B}) \mathbf{x} = \underbrace{\mathbf{x}^T \mathbf{A} \mathbf{x}}_{>0} + \underbrace{\mathbf{x}^T \mathbf{B} \mathbf{x}}_{\geq 0} > 0.$$
$\blacksquare$

1) Parallel Processing Stability: For parallel processing, the state covariance update, given by equation (4), has the form

$$\begin{aligned}
\hat{\mathbf{P}} &= \boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{X}}^T - \boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}}^T \left(\boldsymbol{\mathcal{Y}}\boldsymbol{\mathcal{Y}}^T + \mathbf{R} \right)^{-1} \boldsymbol{\mathcal{Y}}\boldsymbol{\mathcal{X}}^T \\
&= \boldsymbol{\mathcal{X}} \left(\mathbf{1} - \boldsymbol{\mathcal{Y}}^T \left(\boldsymbol{\mathcal{Y}}\boldsymbol{\mathcal{Y}}^T + \mathbf{R} \right)^{-1} \boldsymbol{\mathcal{Y}} \right) \boldsymbol{\mathcal{X}}^T.
\end{aligned}$$

We will now introduce two forms of the well known Sherman-Morrison-Woodbury (SMW) identity [19], which will play an important role in our analysis:

$$\mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B} \left(\mathbf{C}^{-1} + \mathbf{D}\mathbf{A}^{-1}\mathbf{B} \right)^{-1} \mathbf{D}\mathbf{A}^{-1} = (\mathbf{A} + \mathbf{B}\mathbf{C}\mathbf{D})^{-1} \quad (8)$$

$$\mathbf{A}^{-1}\mathbf{B} \left(\mathbf{C}^{-1} + \mathbf{D}\mathbf{A}^{-1}\mathbf{B} \right)^{-1} = (\mathbf{A} + \mathbf{B}\mathbf{C}\mathbf{D})^{-1} \mathbf{B}\mathbf{C} \quad (9)$$

Using (8), we can re-write the state covariance as

$$\hat{\mathbf{P}} = \boldsymbol{\mathcal{X}} \left(\mathbf{1} + \boldsymbol{\mathcal{Y}}^T \mathbf{R}^{-1} \boldsymbol{\mathcal{Y}} \right)^{-1} \boldsymbol{\mathcal{X}}^T.$$

Given that $\mathbf{R}^{-1}$ is positive definite, if we decompose it into Cholesky factors, $\mathbf{S}\mathbf{S}^T := \mathbf{R}$, we see that $\boldsymbol{\mathcal{Y}}^T \mathbf{R}^{-1} \boldsymbol{\mathcal{Y}} =$

$(\boldsymbol{\mathcal{Y}}^T \mathbf{S})(\boldsymbol{\mathcal{Y}}^T \mathbf{S})^T$, which is at least semi-positive definite. Since $\mathbf{1}$ is the identity matrix and therefore positive definite, by Theorem 2, we know that $\mathbf{1} + \boldsymbol{\mathcal{Y}}^T \mathbf{R}^{-1} \boldsymbol{\mathcal{Y}}$ is positive definite and, therefore, so is its inverse. By Theorem 1, we therefore have that $\hat{\mathbf{P}} > 0$ for the case of parallel processing.

2) Serial Processing Stability: For 'naive serial' processing, the state covariance update, given by equation (5), has the form

$$\hat{\mathbf{P}} = \boldsymbol{\mathcal{X}} \left(\mathbf{1} - \sum_{i=1}^{M} \boldsymbol{\mathcal{Y}}_i^T \left(\boldsymbol{\mathcal{Y}}_i \boldsymbol{\mathcal{Y}}_i^T + \mathbf{R}_i \right)^{-1} \boldsymbol{\mathcal{Y}}_i \right) \boldsymbol{\mathcal{X}}^T.$$

Clearly, one cannot ensure that the matrix $\mathbf{1} - \sum_{i=1}^{M} \boldsymbol{\mathcal{Y}}_i^T \left(\boldsymbol{\mathcal{Y}}_i \boldsymbol{\mathcal{Y}}_i^T + \mathbf{R}_i \right)^{-1} \boldsymbol{\mathcal{Y}}_i > 0$, especially for large M. By Theorem 1, if $\mathbf{1} - \sum_{i=1}^{M} \boldsymbol{\mathcal{Y}}_i^T \left(\boldsymbol{\mathcal{Y}}_i \boldsymbol{\mathcal{Y}}_i^T + \mathbf{R}_i \right)^{-1} \boldsymbol{\mathcal{Y}}_i \leq 0$, then the state covariance matrix will not be positive definite. Note, however, that if the sigma-points are re-drawn after each measurement update, then the state covariance would be given by the following, where $(\cdot)^i$ denotes iteration i:

$$\hat{\mathbf{P}}^i = \boldsymbol{\mathcal{X}}^i \left(\mathbf{1} + \boldsymbol{\mathcal{Y}}^{i^T} \mathbf{R}^{-1} \boldsymbol{\mathcal{Y}}^i \right)^{-1} \boldsymbol{\mathcal{X}}^{i^T},$$

where $\boldsymbol{\mathcal{X}}^i$ is given by equation (3) and constructed with sigma-points from $\{\hat{\mathbf{x}}^{i-1}, \hat{\mathbf{P}}^{i-1}\}$. By the same reasoning as in the parallel case, we have that $\hat{\mathbf{P}}^i > 0$ for $i = 1 \ldots M$. Of course, this approach does not yield the same result as parallel processing and for our localization simulations, re-drawing resulted in worse performance than parallel processing.

C. New Serial Processing Approach

We now introduce our new serial processing approach. One of the important aspects of this method has already been introduced, which is to express the standard covariance update equation, $\hat{\mathbf{P}} = \hat{\mathbf{P}}^- - \mathbf{K}\mathbf{U}^T$, as the following

$$\hat{\mathbf{P}} = \boldsymbol{\mathcal{X}} \left(\mathbf{1} + \boldsymbol{\mathcal{Y}}^T \mathbf{R}^{-1} \boldsymbol{\mathcal{Y}} \right)^{-1} \boldsymbol{\mathcal{X}}^T,$$

which we obtain from the first form of the SMW identity. Using the fact that we are considering uncorrelated measurements, we define the following quantity:

$$\mathbf{C} := \left(\mathbf{1} + \sum_{i=1}^{M} \boldsymbol{\mathcal{Y}}_i^T \mathbf{R}_i^{-1} \boldsymbol{\mathcal{Y}}_i \right)^{-1}$$

Note that $\mathbf{C}^{-1}$ can be computed serially and has dimension $2L + 1 \times 2L + 1$, which is relatively inexpensive to invert, since we are interested in the case where $M \gg L$. After serially computing $\mathbf{C}^{-1}$, the state covariance update is given by $\hat{\mathbf{P}} = \boldsymbol{\mathcal{X}} \mathbf{C} \boldsymbol{\mathcal{X}}^T$. Thus, we have introduced a method that is based on the parallel processing equations, but serially computes the quantity $\mathbf{C}^{-1}$ to obtain the estimated state covariance.

In a similar manner, we can develop a serial approach to compute the mean of the state estimate, which is given by

$$\hat{\mathbf{x}} = \hat{\mathbf{x}}^- + \boldsymbol{\mathcal{X}} \boldsymbol{\mathcal{Y}}^T \left(\boldsymbol{\mathcal{Y}} \boldsymbol{\mathcal{Y}}^T + \mathbf{R} \right)^{-1} (\mathbf{y} - \hat{\mathbf{y}}).$$

We now invoke the second form of the SMW identity (9) to give us

$$\hat{\mathbf{x}} = \hat{\mathbf{x}}^- + \boldsymbol{\mathcal{X}} \left(\mathbf{1} + \boldsymbol{\mathcal{Y}}^T \mathbf{R}^{-1} \boldsymbol{\mathcal{Y}} \right)^{-1} \boldsymbol{\mathcal{Y}}^T \mathbf{R}^{-1} (\mathbf{y} - \hat{\mathbf{y}}).$$

Using our newly defined quantity, $\mathbf{C}$, and the fact that the measurements are uncorrelated, we can simplify the above to

$$\hat{\mathbf{x}} = \hat{\mathbf{x}}^- + \boldsymbol{\mathcal{X}} \mathbf{C} \underbrace{\left(\sum_{i=1}^{M} \boldsymbol{\mathcal{Y}}_i^T \mathbf{R}_i^{-1} (\mathbf{y}_i - \hat{\mathbf{y}}_i) \right)}_{=:\mathbf{d}}.$$

Again, $\mathbf{C}^{-1}$ and $\mathbf{d}$ can be computed serially, after which they are pre-multiplied by $\boldsymbol{\mathcal{X}}$. In essence, we have decoupled the state from the measurements, in order to compute the measurement quantities in a serial manner. It is interesting to note the similarity with the Unscented Information Filter [10, 11], however, there are several key differences. Firstly, we directly compute the mean and not the information vector. Secondly, we do not have an extra Jacobian term in the innovation. Thirdly, this form is only attainable if one makes explicit use of the statistical Jacobian factors (i.e., $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$).

It is worth mentioning that under this approach, regardless of the order in which the measurements are processed, we obtain an algebraically identical result to the parallel version. Our serial processing method, which will be referred to as the S-SPKF, is summarized below, where we have re-introduced the time-step, k, to make things comparable to the canonical SPKF. The predictive step has been omitted since it is identical to the canonical SPKF predictive step.

1) Serial Correction Step $\left\{ \hat{\mathbf{x}}_k^-, \hat{\mathbf{P}}_k^- \right\} \to \left\{ \hat{\mathbf{x}}_k, \hat{\mathbf{P}}_k \right\}$:
1) Initialize our serial processing variables:

$$\mathbf{C}_{0,k}^{-1} := \mathbf{1}, \qquad \mathbf{d}_{0,k} := \mathbf{0}$$

2) For all measurements, $n = 1 \ldots M$, do the following:
   a) Pass each sigma-point through the nonlinear observation model:

   $$\boldsymbol{\mathcal{Y}}_{i,k} := \mathbf{g} \left(\boldsymbol{\mathcal{X}}_{i,k}^- \right) \qquad i = 0 \ldots 2L.$$

   b) Compute the predicted measurement and $\boldsymbol{\mathcal{Y}}_{n,k}$

   $$\hat{\mathbf{y}}_{n,k} := \sum_{i=0}^{2L} \beta_i \boldsymbol{\mathcal{Y}}_{i,k},$$

   $$\boldsymbol{\mathcal{Y}}_{n,k} := \left[\ldots \sqrt{\beta_i} \left(\boldsymbol{\mathcal{Y}}_{i,k} - \hat{\mathbf{y}}_{n,k} \right) \ldots \right],$$

   for $i = 0 \ldots 2L$.
   c) Compute $\mathbf{C}_{n,k}^{-1}$ and $\mathbf{d}_{n,k}$

   $$\mathbf{C}_{n,k}^{-1} := \mathbf{C}_{n-1,k}^{-1} + \boldsymbol{\mathcal{Y}}_{n,k}^T \mathbf{R}_{n,k}^{-1} \boldsymbol{\mathcal{Y}}_{n,k},$$

   $$\mathbf{d}_{n,k} := \mathbf{d}_{n-1,k} + \boldsymbol{\mathcal{Y}}_{n,k}^T \mathbf{R}_{n,k}^{-1} (\mathbf{y}_{n,k} - \hat{\mathbf{y}}_{n,k}).$$

3) Compute $\boldsymbol{\mathcal{X}}_k$

$$\boldsymbol{\mathcal{X}}_k := \left[\ldots \sqrt{\beta_i} \left(\boldsymbol{\mathcal{X}}_{i,k}^- - \hat{\mathbf{x}}_k^- \right) \ldots \right], \ i = 0 \ldots 2L$$

4) Compute the posterior belief, $\left\{ \hat{\mathbf{x}}_k, \hat{\mathbf{P}}_k \right\}$

$$\hat{\mathbf{x}}_k := \hat{\mathbf{x}}_k^- + \boldsymbol{\mathcal{X}}_k \mathbf{C}_k \mathbf{d}_k, \quad \hat{\mathbf{P}}_k := \boldsymbol{\mathcal{X}}_k \mathbf{C}_k \boldsymbol{\mathcal{X}}_k^T.$$
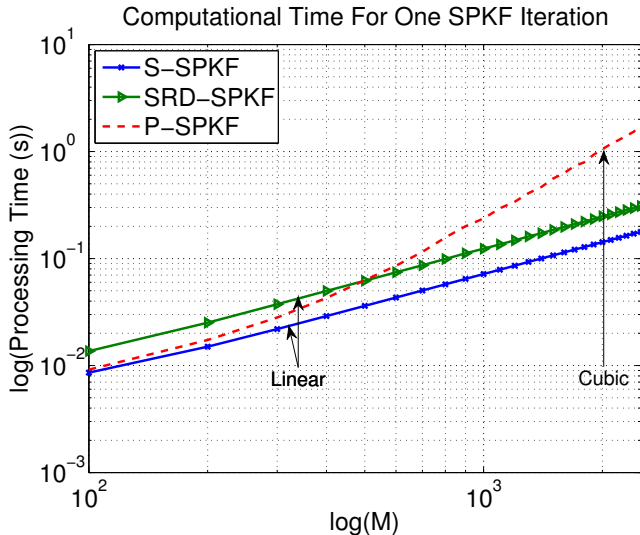
Fig. 2. Processing time (s) versus measurement dimension (M) for various processing implementations. Note that the line slopes indicate the computational complexities. In this case, the serial approaches have a linear slope with respect to the measurement dimension and the parallel approach has a cubic cost, as expected. The SRD-SPKF has a larger vertical offset than the S-SPKF because it has to re-draw sigma points each iteration.

V. RESULTS

The results section has been divided into two subsections: the first presents our experimental efficiency results and the second presents our accuracy analysis. For our simulations, we ran Matlab v7.7 on a 2.66GHz Intel Core i7 CPU.

A. Efficiency Results

We compared our serial processing approach (the S-SPKF) against the following two SPKF implementations:

1) Serial Processing - Re-drawing Sigma-points: We consider a serial processing approach that re-draws sigma-points after each iteration in the measurement update step. This method will be denoted as the SRD-SPKF, where 'RD' stands for 're-draw'.

2) Parallel Processing - Triangular Decomposition: Instead of directly inverting the large $M \times M$ innovation covariance matrix, we solve the system $\mathbf{K}_k \mathbf{V}_k = \mathbf{U}_k$ for $\mathbf{K}_k$ using a triangular decomposition scheme (i.e., we compute the triangular Cholesky factors, $\mathbf{SS}^T = \mathbf{V}_k$, and then solve $\mathbf{K}_k = \left(\mathbf{U}_k / \mathbf{S}^T \right) / \mathbf{S}$. using forward/backward substitutions). This method will be denoted by P-SPKF, since we are processing the measurements in parallel.

To test the efficiency of each filter implementation, we examined the processing time required for one SPKF function call versus a range of measurement dimensions ($M = 100 \ldots 2500$). Figure 2 shows the results averaged over 10 trials on a log-log plot, where we have plotted the processing time versus the measurement dimension.

Using the log-log plot, we experimentally estimated the cost of each method by calculating the respective line slopes. Although this approximation does not take into account additional terms, the aim was to provide a rough estimate of
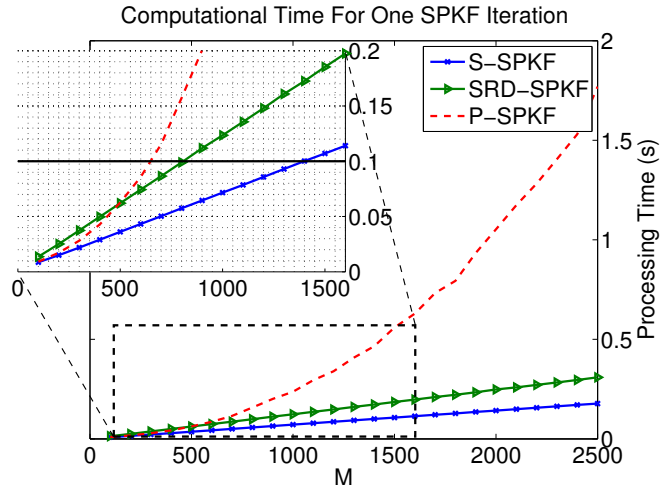

Fig. 3. Processing time (s) versus measurement dimension (M) for various processing implementations (Note: this is the linear plot of Figure 2). We have provided an example cutoff time in the inset figure near the top left, which indicates the maximum number of measurements that can be processed in one control cycle operating at 10Hz.

TABLE I
EXPERIMENTAL ORDERS OF COMPLEXITY FOR METHODS SHOWN IN
FIGURE 3 (BASED ON LINE SLOPES)

Method	Experimental	Theoretical
S-SPKF	$\mathcal{O}(M^{1.0})$	$\mathcal{O}(M)$
SRD-SPKF	$\mathcal{O}(M^{1.0})$	$\mathcal{O}(M)$
P-SPKF	$\mathcal{O}(M^{2.1})$	$\mathcal{O}(M^3)$

the computational complexities associated with each method. Table I shows our experimentally estimated results, along with the theoretical complexities for comparison. Note that since the state dimension and process noise dimension, L, is held constant, only the dependence on the measurement dimension can be evaluated.

As expected, the Cholesky decomposition method was the most costly[5], followed by the serial processing methods, where we see that our method is actually less expensive than the SRD-SPKF. This is because the SRD-SPKF has to re-draw sigma-points and recompute statistical Jacobians for each measurement update step, which is not required in the S-SPKF.

To illustrate the importance of this serial processing approach for mobile robots, we wish to emphasize the fact that for online systems, processing constraints can limit the number of measurements used in the SPKF. For instance, a typical mobile robot may have a control loop that operates at a frequency of 10Hz, meaning that the SPKF must provide the controller with state estimates at least once every 0.1s. For this particular case, we have included the cutoff time of 0.1s in Figure 3, which indicates the maximum number of sensor measurements the SPKF could afford to process per control cycle. The results indicate that the S-SPKF can process approximately 2.3 times as many measurements as the P-SPKF

---

[5]The experimental order of complexity for the P-SPKF is lower than the theoretical complexity due to the fact that Matlab's built-in Cholesky function uses an efficient Fortran subroutine, which reduces the processing time.

and approximately 1.8 times more than the SRD-SPKF.

Although we have restricted our attention to measurement dimensions between 100 and 2500, data acquisition rates for 3D lidar sensors can be considerably higher and amount to much larger measurement dimensions. For instance, the Velodyne HDL-64E S2 lidar can acquire approximately 1.8 million points per second[6], which, depending on the frequency at which the estimator is running, would amount to hundreds of thousands of measurements per SPKF iteration. In this case, the efficiency gains from serial processing versus parallel processing would be critical for systems that wish to incorporate such data-dense sensors.
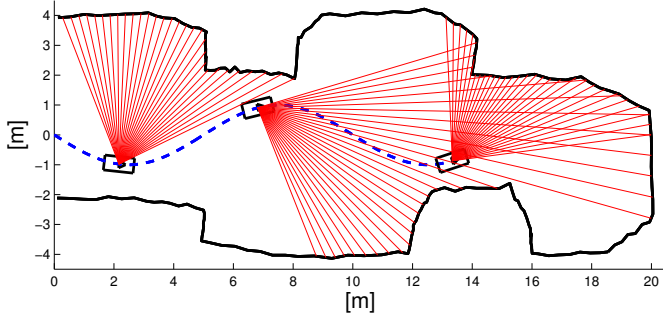

Fig. 4. Snapshots of simulated robot using a 2D lidar sensor to localize against a known map. In this example, only 30 laser rays were plotted for illustration; however the measurement dimension was 200 (i.e., 200 range readings).

B. Accuracy Results

As a consistency check, we ran each SPKF processing method through the simulation shown in Figure 4, which is a mobile robot localizing against a known map and using the SPKF to track a path along a corridor. This is similar to the scenario described by Marshall et al. [14]. Furthermore, we compared the accuracy of the SPKF against the EKF to demonstrate the superior performance of the SPKF (for details on the EKF, the reader is referred to [3]). It should be noted that we used numerical Jacobians in the EKF as we only have a generative (i.e., black box) sensor model due to the unstructured occupancy grid map of the corridor.

The estimation error was defined as the root-mean-squared (RMS) error with respect to groundtruth:

$$E_{\hat{\mathbf{x}}} := \sqrt{\frac{\sum_{k=0}^{N} |\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_{k_{true}}|^2}{N+1}},$$

and the relative error between covariance estimates of the P-SPKF and S-SPKF was measured using the Frobenious norm:

$$E_{\hat{\mathbf{P}}} := \sum_{k=0}^{N} \sqrt{\text{tr}\left[\left(\hat{\mathbf{P}}_{k,\text{serial}} - \hat{\mathbf{P}}_{k,\text{parallel}}\right)^T \left(\hat{\mathbf{P}}_{k,\text{serial}} - \hat{\mathbf{P}}_{k,\text{parallel}}\right)\right]}.$$

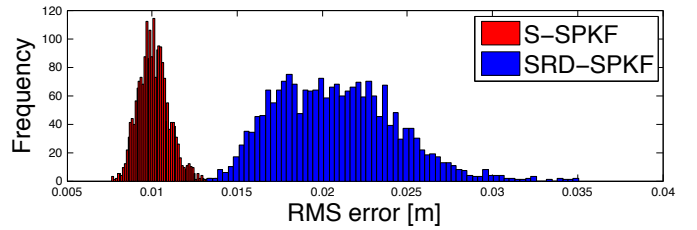Also, to ensure a fair comparison, we used the same random noise for each method.

Fig. 5. Histograms of the estimation error, $E_{\hat{\mathbf{x}}}$, for the S-SPKF and SRD-SPKF vs. ground truth, over 1000 runs. The S-SPKF achieved a mean error that was almost half the error for the SRD-SPKF.

As expected, both the error in the mean and covariance between the S-SPKF and P-SPKF agreed to within machine zero and the S-SPKF outperformed the EKF; the EKF's mean error was 3.7 times greater than the S-SPKF. As noted earlier, the EKF is not well suited for this problem as numerical Jacobians are required since the observation model does not have an analytical expression. Not only does this increase the EKF's computational cost, but it also adds a potential instability in the filter.

In addition, we examined the accuracy of the SRD-SPKF against the S-SPKF to demonstrate that re-drawing, or re-linearizing, after each measurement is not necessarily the best approach. Figure 5 shows the mean error histograms with respect to groundtruth for both methods over a 1000 random runs, where we see that the mean error for the S-SPKF is almost half that of the SRD-SPKF. Thus, we found that re-drawing sigma-points provided absolutely no advantage, as it was more costly and did not result in better estimates.

VI. DISCUSSION

The SPKF can encounter stability issues if the standard measurement-update step is used to naively process the measurements serially. This is due to a coupling between the predicted measurements (i.e., the off-diagonal terms in equation (2)) that is being ignored in the 'naive' serial approach. For our localization simulations, we were never able to run a complete simulation without the state covariance matrix losing its positive definiteness. In contrast, our serial approach does not encounter this stability issue since it is algebraically equivalent to the parallel processing case and therefore includes the coupling information. Re-drawing sigma-points also takes into account the coupling between the predicted measurements, in the sense that each update is based on the sigma-points from the previous iteration. However, as was shown, the SRD-SPKF did not perform as well as the S-SPKF and is in fact, more costly since in each iteration sigma-points are being re-drawn.

In addition, serial filters that employ re-linearization can be sensitive to the order in which the measurements are processed. For example, if an outlier or noisy measurement is processed early, it can perturb the estimate for subsequent iterations. For radar target tracking problems, it is well known that the order in which the measurements are serially processed has a significant effect on the accuracy [15, 18, 2]. For instance, the serial SPKF method by Duan et al. [2], which re-draws sigma-points, processed measurements in order of

decreasing accuracy: azimuth, elevation and range. In our approach, the same estimate is obtained regardless of the order in which the measurements are serially processed.

In our simulations, we found that this problem of re-linearizing about a poor operating point (i.e., a poor estimate) was an issue for the SRD-SPKF, depending on the selection of the sigma-point weights. In other words, we found the SRD-SPKF to be more sensitive to outliers than the S-SPKF. Although RANSAC could be applied to the SRD-SPKF, as it has been for the EKF [23], we note that our S-SPKF would only help RANSAC, as it is less sensitive to outliers. It is not our contention that re-linearizing will always be problematic or less accurate. However, we recognize that in some problems, such as our localization problem, re-linearizing does not result in better estimates and is not as robust to outliers.

VII. CONCLUSION

This paper presents a novel serial processing approach to handling high-dimensional measurements within the SPKF. We demonstrated that our S-SPKF algorithm is more accurate than either the S-EKF or SRD-UKF and is the fastest SPKF algorithm of all considered, with a computational cost that is linear in the number of measurements. We also proved that naively applying the SPKF measurement update step in a serial manner cannot guarantee a positive-definite state covariance matrix. By re-writing the SPKF equations in its statistical Jacobian form and utilizing the Sherman-Morrison-Woodbury identity, we showed how to decouple the state and measurement variables in order to serially process the measurements. For practical field robotic systems that incorporate data-rich sensors, we have provided a new form of the SPKF that offers the robustness and accuracy of parallel processing, but with a minimal computational cost.

REFERENCES

[1] H.G. Asl and S.H. Pourtakdoust. Ud covariance factorization for unscented kalman filter using sequential measurements update. *World Academy of Science, Engineering and Technology*, 34: 368–376, 2007.

[2] Z. Duan, Li. X.R., C. Han, and H. Zhu. Sequential unscented kalman filter for radar target tracking with range rate measurements. In *Proc. of the 8th Int'l. Conf. on Information Fusion*, volume 1, Philadelphia, Pennsylvania, USA, July 25-28 2005.

[3] A. Gelb, J. Kasper Jr., R. Nash Jr., C. Price, and A. Sutherland Jr. *Applied Optimal Estimation*. The M.I.T. Press, 1974.

[4] K. Ito and K. Xiong. Gaussian filters for nonlinear filtering problems. *IEEE Trans. on Automatic Control*, 45(5):910–927, 1999.

[5] S. Julier. The spherical simplex unscented transformation. In *Proc. of the American Control Conf.*, Denver, Colorado, USA, June 2-5 2003.

[6] S. Julier and J. Uhlmann. A general method for approximating nonlinear transformations of probability distributions. Technical report, Robotics Research Group, Dept. of Engineering Science, University of Oxford, 1996.

[7] S. Julier and J. Uhlmann. Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations. In *Proc. of the IEEE American Control Conf.*, volume 2, pages 887 – 892, Anchorage, Alaska, USA, May 8-10 2002.

[8] S. Julier and J. Uhlmann. Unscented filtering and nonlinear estimation. *Proc. of the IEEE*, 92(3):401–422, 2004.

[9] S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proc. of the American Control Conf.*, volume 3, pages 1628–1632, Seattle WA, USA, 1995.

[10] Deok-Jin Lee. Nonlinear estimation and multiple sensor fusion using unscented information filtering. *IEEE Signal Processing Letters*, 15:861–864, 2008.

[11] Deok-Jin Lee. Unscented information filtering for distributed estimation and multiple sensor fusion. In *AIAA Guidance, Navigation and Control Conf. and Exhibit*, Honolulu, Hawaii, USA, 18-21 August 2008.

[12] T. Lefebvre, H. Bruyninckx, and J. De Schutter. Kalman filters for nonlinear systems: a comparison of performance. *Int'l. Journal of Control*, 77(7):639–653, 2001.

[13] M. Lei and C. Han. Sequential nonlinear tracking using ukf and raw range-rate measurements. *IEEE Trans. on Aerospace and Electronic Systems*, 43(1):239–250, 2007.

[14] J. Marshall, T.D. Barfoot, and J. Larsson. Autonomous underground tramming for center-articulated vehicles. *Journal of Field Robotics*, 25(6-7):400–421, 2008.

[15] K.S. Miller and D.M. Leskiw. Nonlinear estimation with radar observations. *IEEE Trans. Aerospace and Electronic Systems*, 18(2):192–200, 1982.

[16] N. Norgaard, N. Poulsen, and O. Ravn. New developments in state estimation for nonlinear systems. *Automatica*, 36(11): 1627–1638, 2000.

[17] B. Quine. A derivative-free implementation of the extended kalman filter. *Automatica*, 42(11):1927–34, 2006.

[18] X. Rong Li and V.P. Jilkov. A survey of maneuvering target tracking-part iii: Measurement models. In *Proc. of SPIE Conf. on Signal and Data Processing of Small Targets*, volume 4473(41), pages 423–446, San Diego, California, USA, July-August 2001.

[19] J. Sherman and W. J. Morrison. Adjustment of an inverse matrix corresponding to changes in the elements of a given column or given row of the original matrix. *Annals of Mathematics and Statistics*, 20:621, 1949.

[20] G. Sibley, G. Sukhatme, and L. Matthies. The iterated sigma point kalman filter with applications to long range stereo. In *Proceedings of Robotics Science and Systems*, Philadelphia, Pennsylvania, USA, Aug 2006.

[21] R. van der Merwe. *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. PhD thesis, OGI School of Science and Engineering, Oregon Health and Science University, June 2004.

[22] R. van der Merwe and E. Wan. The square-root unscented kalman filter for state and parameter-estimation. In *IEEE Int'l. Conf. on Acoustics, Speech, and Signal Processing*, volume 6, pages 3461–3464, Salt Lake City, Utah, USA, May 7-11 2001.

[23] A. Vedaldi, H. Jin, P. Favaro, and S. Soatto. Kalmansac: Robust filtering by consensus. In *Proc. of the Int'l. Conf. on Computer Vision (ICCV)*, volume 1, pages 663–640, Beijing, China, 2005.

[24] T. Vercauteren and X. Wang. Decentralized sigma-point information filters for target tracking in collaborative sensor networks. *IEEE Trans. on Signal Processing*, 53(8):2997–3009, 2005.

[25] E. Wan and R. van der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of Symposium 2000 on Adaptive Systems for Signal Processing, Communications, and Control (AS-SPCC)*, pages 153–158, Lake Louise, Alberta, Canada, Oct. 1-4 2000.