A Kalman Filter for Robust Outlier Detection

Jo-Anne Ting[1], Evangelos Theodorou[1], and Stefan Schaal[1,2]
[1]University of Southern California, Los Angeles, CA, 90089
[2]ATR Computational Neuroscience Laboratories, Kyoto, Japan
{joanneti, etheodor, sschaal}@usc.edu

Abstract— **In this paper, we introduce a modified Kalman filter that can perform robust, real-time outlier detection in the observations, without the need for parameter tuning. Robotic systems that rely on high quality sensory data can be sensitive to data containing outliers. Since the standard Kalman filter is not robust to outliers, other variations of the Kalman filter have been proposed to overcome this issue, but these methods may require parameter tuning, use of heuristics or complicated parameter estimation. Our Kalman filter uses a weighted least squares-like approach by introducing weights for each data sample. A data sample with a smaller weight has a weaker contribution when estimating the current time step's state. We learn the weights and system dynamics using a variational Expectation-Maximization framework. We evaluate our Kalman filter algorithm on synthetic data and data from a robotic dog.**

I. INTRODUCTION

In order to maintain robust control in robotic systems, a high quality of sensory data is needed. While data from sensors such as potentiometers and optical encoders are easily interpretable in their noise characteristics, other sensors such as visual systems, GPS devices and sonar sensors often provide measurements populated with outliers. As a result, robust, reliable detection and removal of outliers is essential in order to process these kinds of data. For example, the application domain of legged locomotion is particularly vulnerable to perceptual data of poor quality, since one undetected outlier can disturb the balance controller to the point that the robot loses stability.

An outlier is generally defined as an observation that "lies outside some overall pattern of distribution" [1]. Outliers may originate from sensor noise (producing values that fall outside a valid range), from temporary sensor failures, or from unanticipated disturbances in the environment (e.g., a brief change of lighting conditions for a visual sensor).

For real-time applications, storing data samples may not be a viable option due to the high frequency of sensory data and insufficient memory resources. In this scenario, sensor data are made available one at a time and must be discarded once they have been observed. Hence, techniques that require access to the entire set of data samples, such as the Kalman smoother (e.g., [2], [3]), are not applicable. Instead, the Kalman filter (e.g., [4], [5]) is a more suitable method, since it assumes that only data samples up to the current time step have been observed. The Kalman filter propagation and update equations are recursive and do not require direct access to previously observed data.

The Kalman filter is a widely used tool for estimating the state of a dynamic system, given noisy measurement data. It is the optimal *linear* estimator for linear Gaussian systems, giving the minimum mean squared error [6]. Using state estimates, the filter can also estimate what the corresponding (output) data are. However, the performance of the Kalman filter degrades when the observed data contains outliers.

To address this, previous work has tried to make the Kalman filter more robust to outliers by addressing the sensitivity of the squared error criterion to outliers [7], [8]. One class of approaches considers non-Gaussian distributions for random variables (e.g., [9], [10], [11], [12]), since multivariate Gaussian distributions are known to be susceptible to outliers. For example, [13] use multivariate Student-t distributions. However, the resulting estimation of parameters may be quite complicated for systems with transient disturbances.

Alternatively, it is possible to model the observation and state noise as non-Gaussian, heavy-tailed distributions to account for non-Gaussian noise and outliers (e.g., [14], [15], [16]). Unfortunately, these filters are typically more difficult to implement and may no longer provide the conditional mean of the state vector. Other approaches use resampling techniques (e.g., [17], [18]) or numerical integration (e.g., [19], [20]), but these may require heavy computation not suitable for real-time applications.

Yet another class of methods uses a weighted least squares approach, as done in robust least squares [21], [22], where the measurement residual error is assigned some statistical property. Some of these algorithms fall under the first category of approaches as well, assuming non-Gaussian distributions for variables. Each data sample is assigned a weight that indicates its contribution to the hidden state estimate at each time step. This technique has been used to produce a Kalman filter that is more robust to outliers (e.g., [23], [24]). However, these methods usually model the weights as some heuristic function of the data (e.g., the Huber function [22]) and often require tuning of threshold parameters for optimal performance. Using incorrect or inaccurate estimates for the weights may lead to deteriorated performance, so special attention and care is necessary when using these techniques.

In this paper, we are interested in the problem of identifying outliers while tracking the observed data using the Kalman filter. Identifying outliers in the state is different problem entirely, and this is left for another paper. We introduce a modified Kalman filter that can detect outliers in the observed data without the need for parameter tuning or use of heuristic methods on the user's part.

This filter learns the weights of each data sample, as well as the system dynamics, using an Expectation-Maximization (EM) framework [25]. For ease of analytical computation, we assume Gaussian distributions for variables and states. We illustrate the performance of this robust Kalman filter on synthetic and robotic data, comparing it with other robust approaches and demonstrating its effectiveness at detecting outliers in the observations.

II. Outlier Detection in the Kalman Filter

Let us assume we have data $\{\mathbf{z}_k\}_{k=1}^N$, observed over N time steps, where $\mathbf{z}_k \in \Re^{d_1 \times 1}$. We denote the corresponding hidden states as $\{\boldsymbol{\theta}_k\}_{k=1}^N$, where $\boldsymbol{\theta}_k \in \Re^{d_2 \times 1}$. If we assume that the system is time-invariant, then the Kalman filter system equations are:

$$\begin{aligned} \mathbf{z}_k &= \mathbf{C}\boldsymbol{\theta}_k + \mathbf{v}_k \\ \boldsymbol{\theta}_k &= \mathbf{A}\boldsymbol{\theta}_{k-1} + \mathbf{s}_k \end{aligned} \quad (1)$$

where $\mathbf{C} \in \Re^{d_1 \times d_2}$ is the observation matrix, $\mathbf{A} \in \Re^{d_2 \times d_2}$ is the state transition matrix, $\mathbf{v}_k \in \Re^{d_1 \times 1}$ is the observation noise at time step k, and $\mathbf{s}_k \in \Re^{d_2 \times 1}$ is the state noise at time step k. We assume $\mathbf{v}_k$ and $\mathbf{s}_k$ to be uncorrelated additive mean-zero Gaussian noise, $\mathbf{v}_k \sim \text{Normal}(0, \mathbf{R})$, $\mathbf{s}_k \sim \text{Normal}(0, \mathbf{Q})$, where $\mathbf{R} \in \Re^{d_1 \times d_1}$ is a diagonal matrix with $\mathbf{r} \in \Re^{d_1 \times 1}$ on its diagonal, and $\mathbf{Q} \in \Re^{d_2 \times d_2}$ is a diagonal matrix with $\mathbf{q} \in \Re^{d_2 \times 1}$ on its diagonal. $\mathbf{R}$ and $\mathbf{Q}$ are covariance matrices for the observation and state noise, respectively. The standard Kalman filter propagation and update equations are, for $k = 1, .., N$:

Propagation:

$$\boldsymbol{\theta}'_k = \mathbf{A}\left\langle \boldsymbol{\theta}_{k-1} \right\rangle \quad (2)$$

$$\boldsymbol{\Sigma}'_k = \mathbf{A}\boldsymbol{\Sigma}_{k-1}\mathbf{A}^T + \mathbf{Q} \quad (3)$$

Update:

$$\mathbf{S}'_k = \left(\mathbf{C}\boldsymbol{\Sigma}'_k\mathbf{C}^T + \mathbf{R}\right)^{-1} \quad (4)$$

$$K'_k = \boldsymbol{\Sigma}'_k\mathbf{C}^T\mathbf{S}'_k \quad (5)$$

$$\left\langle \boldsymbol{\theta}_k \right\rangle = \boldsymbol{\theta}'_k + K'_k\left(\mathbf{z}_k - \mathbf{C}\boldsymbol{\theta}'_k\right) \quad (6)$$

$$\boldsymbol{\Sigma}_k = \left(\mathbf{I} - K'_k\mathbf{C}\right)\boldsymbol{\Sigma}'_k \quad (7)$$

where $\langle \boldsymbol{\theta}_k \rangle$ is the posterior mean vector of the state $\boldsymbol{\theta}_k$, $\boldsymbol{\Sigma}_k$ is the posterior covariance matrix of $\boldsymbol{\theta}_k$, and $\mathbf{S}'_k$ is the covariance matrix of the residual prediction error—all at time step k. In this problem, the system dynamics ($\mathbf{C}$, $\mathbf{A}$, $\mathbf{R}$ and $\mathbf{Q}$) are unknown, and it is possible to use a maximum likelihood framework to estimate these parameter values [26]. Unfortunately, this standard Kalman filter model considers all data samples to be part of the data cloud and is not robust to outliers.

A. Robust Kalman Filtering with Bayesian Weights

To overcome this limitation, we introduce a novel Bayesian algorithm that treats the weights associated with each data sample probabilistically. In particular, we introduce a scalar weight w_k for each observed data sample $\mathbf{z}_k$ such that the variance of $\mathbf{z}_k$ is weighted with w_k, as done in [27]. [27] consider a weighted least squares regression model and assumes that the weights are known and given. We model the weights to be Gamma distributed random variables, as done previously in [28] for weighted linear regression. The resulting prior distributions are then:

$$\begin{aligned} \mathbf{z}_k|\boldsymbol{\theta}_k, w_k &\sim \text{Normal}\left(\mathbf{C}\boldsymbol{\theta}_k, \mathbf{R}/w_k\right) \\ \boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1} &\sim \text{Normal}\left(\mathbf{A}\boldsymbol{\theta}_{k-1}, \mathbf{Q}\right) \\ w_k &\sim \text{Gamma}\left(a_{w_k}, b_{w_k}\right) \end{aligned} \quad (8)$$

We can treat this entire problem as an Expectation-Minimization-like (EM) learning problem [25], [29] and maximize the log likelihood $\log p(\boldsymbol{\theta}_{1:N})$ (otherwise known as the "incomplete" log likelihood with the hidden probabilistic variables marginalized out). Due to analytical issues, we only have access to a lower bound of this measure. This lower bound is based on an expected value of the "complete" data likelihood $\langle \log p\left(\boldsymbol{\theta}_{1:N}, \mathbf{z}_{1:N}, \mathbf{w}\right) \rangle$ [1], formulated over all variables of the learning problem:

$$\begin{aligned} &\log p\left(\boldsymbol{\theta}_{1:N}, \mathbf{z}_{1:N}, \mathbf{w}\right) \\ &= \sum_{i=1}^{N} \log p\left(\mathbf{z}_i|\boldsymbol{\theta}_i, w_i\right) + \sum_{i=1}^{N} \log p\left(\boldsymbol{\theta}_i|\boldsymbol{\theta}_{i-1}\right) \\ &\quad + \log p\left(\boldsymbol{\theta}_0\right) + \sum_{i=1}^{N} \log p\left(w_i\right) \\ &= \frac{1}{2}\sum_{i=1}^{N} \log w_i - \frac{N}{2}\log|\mathbf{R}| - \frac{N}{2}\log|\mathbf{Q}| \\ &\quad - \frac{1}{2}\sum_{i=1}^{N} w_i\left(\mathbf{z}_i - \mathbf{C}\boldsymbol{\theta}_i\right)^T\mathbf{R}^{-1}\left(\mathbf{z}_i - \mathbf{C}\boldsymbol{\theta}_i\right) \\ &\quad - \frac{1}{2}\sum_{i=1}^{N}\left(\boldsymbol{\theta}_i - \mathbf{A}\boldsymbol{\theta}_{i-1}\right)^T\mathbf{Q}^{-1}\left(\boldsymbol{\theta}_i - \mathbf{A}\boldsymbol{\theta}_{i-1}\right) \\ &\quad - \frac{1}{2}\log|\mathbf{Q}_0| - \frac{1}{2}\left(\boldsymbol{\theta}_0 - \hat{\boldsymbol{\theta}}_0\right)^T\mathbf{Q}_0^{-1}\left(\boldsymbol{\theta}_0 - \hat{\boldsymbol{\theta}}_0\right) \\ &\quad + \sum_{i=1}^{N}\left(a_{w_i,0}\right)\log w_i - \sum_{i=1}^{N} b_{w_i,0}w_i + \text{const}_{\boldsymbol{\theta},\mathbf{z},\mathbf{w}} \end{aligned} \quad (9)$$

where $\boldsymbol{\theta}_0$ is the initial state, $\hat{\boldsymbol{\theta}}_0$ is the mean of $\boldsymbol{\theta}_0$, $\mathbf{Q}_0$ is the noise variance of $\boldsymbol{\theta}_0$, $\mathbf{w} \in \Re^{N \times 1}$ has coefficients w_i ($i = 1, .., N$), and $\mathbf{z}_{1:N}$ denotes samples $\{\mathbf{z}_1, \mathbf{z}_2, .., \mathbf{z}_N\}$. Since we are considering this problem as a real-time one (i.e. data samples arrive sequentially, one at a time), we will have observed only data samples $\mathbf{z}_{1:k}$ at time step k. Consequently, in order to estimate the posterior distributions of the random variables and parameter values at time step k, we should consider the log evidence of only the data samples observed to date (i.e., $\log p\left(\boldsymbol{\theta}_{1:k}, \mathbf{z}_{1:k}, \mathbf{w}_{1:k}\right)$).

The expectation of the complete data likelihood should be taken with respect to the true posterior distribution of all hidden variables $Q\left(\mathbf{w}, \boldsymbol{\theta}\right)$. However, since this is an analytically intractable expression, we use a technique from variational calculus to construct a lower bound and make a factorial

[1] Note that $\langle\rangle$ denotes the expectation operator

Preprint submitted to 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. Received April 8, 2007.

approximation of the true posterior as follows: $Q(\mathbf{w}, \boldsymbol{\theta}) = \prod_{i=1}^{N} Q(w_i) \prod_{i=1}^{N} Q(\boldsymbol{\theta}_i | \boldsymbol{\theta}_{i-1}) Q(\boldsymbol{\theta}_0)$ (e.g., [29]). Note that the factorization of $\boldsymbol{\theta}$ only considers the influence of each $\boldsymbol{\theta}_i$ from within its Markov blanket (i.e. $\boldsymbol{\theta}_i$ is dependent on $\boldsymbol{\theta}_{i-1}$, $\mathbf{z}_i$ and w_i). While losing a small amount of accuracy, all resulting posterior distributions over hidden variables become analytically tractable. We can derive the final EM update equations by standard manipulations of Normal and Gamma distributions and arrive at the following for time step k:

E-step:

$$\boldsymbol{\Sigma}_k = \left(\langle w_k \rangle \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k + \mathbf{Q}_k^{-1} \right)^{-1} \tag{10}$$

$$\langle \boldsymbol{\theta}_k \rangle = \boldsymbol{\Sigma}_k \left(\mathbf{Q}_k^{-1} \mathbf{A}_k \langle \boldsymbol{\theta}_{k-1} \rangle + \langle w_k \rangle \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{z}_k \right) \tag{11}$$

$$\langle w_k \rangle = \frac{a_{w_k,0} + \frac{1}{2}}{b_{w_k,0} + \left\langle (\mathbf{z}_k - \mathbf{C}_k \boldsymbol{\theta}_k)^T \mathbf{R}_k^{-1} (\mathbf{z}_k - \mathbf{C}_k \boldsymbol{\theta}_k) \right\rangle} \tag{12}$$

M-step:

$$\mathbf{C}_k = \left(\sum_{i=1}^{k} \langle w_i \rangle \mathbf{z}_i \langle \boldsymbol{\theta}_i \rangle^T \right) \left(\sum_{i=1}^{k} \langle w_i \rangle \left\langle \boldsymbol{\theta}_i \boldsymbol{\theta}_i^T \right\rangle \right)^{-1} \tag{13}$$

$$\mathbf{A}_k = \left(\sum_{i=1}^{k} \langle \boldsymbol{\theta}_i \rangle \langle \boldsymbol{\theta}_{i-1} \rangle^T \right) \left(\sum_{i=1}^{k} \left\langle \boldsymbol{\theta}_{i-1} \boldsymbol{\theta}_{i-1}^T \right\rangle \right)^{-1} \tag{14}$$

$$r_{km} = \frac{1}{k} \sum_{i=1}^{k} \langle w_i \rangle \left\langle (\mathbf{z}_{im} - \mathbf{C}_k(m,:) \boldsymbol{\theta}_i)^2 \right\rangle \tag{15}$$

$$q_{kn} = \frac{1}{k} \sum_{i=1}^{k} \left\langle (\boldsymbol{\theta}_{in} - \mathbf{A}_k(n,:) \boldsymbol{\theta}_{i-1})^2 \right\rangle \tag{16}$$

where $\langle \boldsymbol{\theta}_k \rangle$ is the posterior mean of $\boldsymbol{\theta}_k$; $\boldsymbol{\Sigma}_k$ is the posterior covariance of $\boldsymbol{\theta}_k$; r_{km} is the mth coefficient of the vector $\mathbf{r}_k$ (for $m = 1, .., d_1$); q_{kn} is the nth coefficient of the vector $\mathbf{q}_k$ (for $n = 1, .., d_2$); $\mathbf{C}_k(m,:)$ is the mth row of the matrix $\mathbf{C}_k$; $\mathbf{A}_k(n,:)$ is the nth row of the matrix $\mathbf{A}_k$; and $a_{w_k,0}$ and $b_{w_k,0}$ are prior scale parameters for the weight w_k. Equations (10) to (16) need to be computed once for each time step k (e.g., [30] [31]), when the data sample $\mathbf{z}_k$ becomes available.

Since sensor data is not collected over time, but discarded soon after it is received, we are unable to store all past data samples. As a result, (13) to (16) need to be re-written in incremental form (i.e., using only values observed and calculated in the current time step k). We can do this by collecting sufficient statistics in (13) to (16). The resulting revised M-update equations are then, at time step k:

$$\mathbf{C}_k = \mathrm{sum}_k^{\mathbf{wz}\boldsymbol{\theta}^T} \left(\mathrm{sum}_k^{\mathbf{w}\boldsymbol{\theta}\boldsymbol{\theta}^T} \right)^{-1} \tag{17}$$

$$\mathbf{A}_k = \mathrm{sum}_k^{\boldsymbol{\theta}\boldsymbol{\theta}'} \left(\mathrm{sum}_k^{\boldsymbol{\theta}'\boldsymbol{\theta}'} \right)^{-1} \tag{18}$$

$$\begin{aligned} r_{km} = \frac{1}{k} \Big[& \mathrm{sum}_{km}^{\mathbf{w}zz} - 2\mathbf{C}_k(m,:) \left(\mathrm{sum}_{km}^{\mathbf{w}z\boldsymbol{\theta}} \right) \\ & + \mathrm{diag} \left\{ \mathbf{C}_k(m,:) \left(\mathrm{sum}_k^{\mathbf{w}\boldsymbol{\theta}\boldsymbol{\theta}^T} \right) \mathbf{C}_k(m,:)^T \right\} \Big] \end{aligned} \tag{19}$$

$$\begin{aligned} q_{kn} = \frac{1}{k} \Big[& \mathrm{sum}_{kn}^{\theta^2} - 2\mathbf{A}_k(n,:) \left(\mathrm{sum}_{kn}^{\theta\boldsymbol{\theta}'} \right) \\ & + \mathrm{diag} \left\{ \mathbf{A}_k(n,:) \left(\mathrm{sum}_k^{\boldsymbol{\theta}'\boldsymbol{\theta}'} \right) \mathbf{A}_k(n,:)^T \right\} \Big] \end{aligned} \tag{20}$$

where $m = 1, .., d_1$, $n = 1, .., d_2$, and the sufficient statistics are:

$$\begin{aligned} \mathrm{sum}_k^{\mathbf{wz}\boldsymbol{\theta}^T} &= \langle w_k \rangle \mathbf{z}_k \langle \boldsymbol{\theta}_k \rangle^T + \mathrm{sum}_{k-1}^{\mathbf{wz}\boldsymbol{\theta}^T} \\ \mathrm{sum}_k^{\mathbf{w}\boldsymbol{\theta}\boldsymbol{\theta}^T} &= \langle w_k \rangle \left\langle \boldsymbol{\theta}_k \boldsymbol{\theta}_k^T \right\rangle + \mathrm{sum}_{k-1}^{\mathbf{w}\boldsymbol{\theta}\boldsymbol{\theta}^T} \\ \mathrm{sum}_k^{\boldsymbol{\theta}\boldsymbol{\theta}'} &= \langle \boldsymbol{\theta}_k \rangle \langle \boldsymbol{\theta}_{k-1} \rangle^T + \mathrm{sum}_{k-1}^{\boldsymbol{\theta}\boldsymbol{\theta}'} \\ \mathrm{sum}_k^{\boldsymbol{\theta}'\boldsymbol{\theta}'} &= \left\langle \boldsymbol{\theta}_{k-1} \boldsymbol{\theta}_{k-1}^T \right\rangle + \mathrm{sum}_{k-1}^{\boldsymbol{\theta}'\boldsymbol{\theta}'} \\ \mathrm{sum}_{km}^{\mathbf{w}zz} &= \langle w_k \rangle z_{km}^2 + \mathrm{sum}_{k-1}^{\mathbf{w}zz} \\ \mathrm{sum}_{km}^{\mathbf{w}z\boldsymbol{\theta}} &= \langle w_k \rangle z_{km} \boldsymbol{\theta}_k + \mathrm{sum}_{k-1,m}^{\mathbf{w}z\boldsymbol{\theta}} \\ \mathrm{sum}_{kn}^{\theta^2} &= \left\langle \theta_{kn}^2 \right\rangle + \mathrm{sum}_{k-1,n}^{\theta^2} \\ \mathrm{sum}_{kn}^{\theta\boldsymbol{\theta}'} &= \langle \theta_{kn} \rangle \langle \boldsymbol{\theta}_{k-1} \rangle + \mathrm{sum}_{kn}^{\theta\boldsymbol{\theta}'} \end{aligned}$$

A few remarks should be made regarding the initialization of priors used in (10) to (12), (17) to (20). In particular, the prior scale parameters $a_{w_k,0}$ and $b_{w_k,0}$ should be selected so that the weights $\langle w_k \rangle$ are 1 with some confidence. That is to say, the algorithm starts by assuming most data samples are inliers. For example, we can set $a_{w_k,0} = 1$ and $b_{w_k,0} = 1$ so that $\langle w_k \rangle$ has a prior mean of $a_{w_k,0}/b_{w_k,0} = 1$ with a variance of $a_{w_k,0}/b_{w_k,0}^2 = 1$. Secondly, the algorithm is relatively insensitive to the the initialization of $\mathbf{A}$ and $\mathbf{C}$ and will always converge to the same final solution, regardless of these values. For our experiments, we use $\mathbf{C} = \mathbf{A} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Finally, the initial values of $\mathbf{R}$ and $\mathbf{Q}$ should be set based on the user's initial estimate of how noisy the observed data is (e.g., $\mathbf{R} = \mathbf{Q} = 0.01\mathbf{I}$ for noisy data, $\mathbf{R} = \mathbf{Q} = 10^{-4}\mathbf{I}$ for less noisy data [32]).

B. Relationship to the Kalman Filter

The equations (10) and (11) for the posterior mean and posterior covariance of $\boldsymbol{\theta}_k$ may not look like the standard Kalman filter equations in (2) to (7), but with a little algebraic manipulation, we can show that the model derived in Section II-A is indeed a variant of the Kalman filter. If we substitute the propagation equations, (2) and (3), into the update equations, (4) to (7), we reach recursive expressions for $\langle \boldsymbol{\theta}_k \rangle$ and $\boldsymbol{\Sigma}_k$. By applying this sequence of algebraic manipulations in reverse order to (10) and (11), we arrive at the following:

Propagation:

$$\boldsymbol{\theta}'_k = \mathbf{A}_k \langle \boldsymbol{\theta}_{k-1} \rangle \tag{21}$$

$$\boldsymbol{\Sigma}'_k = \mathbf{Q}_k \tag{22}$$

Update:

$$\mathbf{S}'_k = \left(\mathbf{C}_k \boldsymbol{\Sigma}'_k \mathbf{C}_k^T + \frac{1}{\langle w_k \rangle} \mathbf{R}_k \right)^{-1} \tag{23}$$

$$K'_k = \boldsymbol{\Sigma}'_k \mathbf{C}_k^T \mathbf{S}'_k \tag{24}$$

$$\langle \boldsymbol{\theta}_k \rangle = \boldsymbol{\theta}'_k + K'_k \left(\mathbf{z}_k - \mathbf{C}_k \boldsymbol{\theta}'_k \right) \tag{25}$$

$$\boldsymbol{\Sigma}_k = \left(\mathbf{I} - K'_k \mathbf{C}_k \right) \boldsymbol{\Sigma}'_k \tag{26}$$

Close examination of the above equations show that (10) and (11) in the Bayesian model correspond to standard Kalman filter equations, with modified expressions for $\boldsymbol{\Sigma}'_k$ and $\mathbf{S}'_k$

and time-varying system dynamics. $\boldsymbol{\Sigma}'_k$, is now $\mathbf{Q}_k$ instead of $\left(\mathbf{A}\boldsymbol{\Sigma}_{k-1}\mathbf{A}^T + \mathbf{Q}\right)$.

Additionally, the term $\mathbf{R}_k$ in $\mathbf{S}'_k$ is now weighted. Equation (12) reveals that if the prediction error in $\mathbf{z}_k$ is so large that it dominates the denominator, then the weight $\langle w_k \rangle$ of that data sample will be very small. As this prediction error term in the denominator goes to ∞, $\langle w_k \rangle$ approaches 0. If $\mathbf{z}_k$ has a very small weight $\langle w_k \rangle$, then $\mathbf{S}'_k$, the posterior covariance of the residual prediction error, will be very small, leading to a very small Kalman gain K'_k. In short, the influence of the data sample $\mathbf{z}_k$ will be downweighted when predicting $\boldsymbol{\theta}_k$, the hidden state at time step k.

C. Monitoring the Residual Error

A common sanity check is to monitor the residual error of the data $\mathbf{z}_{1:N}$ and the hidden states $\boldsymbol{\theta}_{1:N}$ in order to ensure that the residual error values stay within the 3σ bounds computed by the filter [32]. If we had access to the true state θ_k for time step k, we would plot the residual state error $(\boldsymbol{\theta}_k - \langle \boldsymbol{\theta}_k \rangle)$ for all time steps k, along with the corresponding $\pm 3\sigma_k$ values, where $\sigma_k^2 = \text{diag}\left\{\boldsymbol{\Sigma}_k\right\}$. We would also plot the residual prediction error $(\mathbf{z}_k - \mathbf{C}\mathbf{A}\langle \boldsymbol{\theta}_{k-1} \rangle)$ for all time steps k, along with the corresponding $\pm 3\sigma_{z_k}$ values, where $\sigma_{z_k}^2 = \text{diag}\left\{\mathbf{S}'_k\right\}$.

With these graphs, we should observe the residual error values remaining within the $\pm 3\sigma$ bounds and check that the residual error does not diverge over time. Residual monitoring may be useful to verify that spurious data samples are rejected, since processing of these samples may result in corrupted filter computations. It offers a peek into the Kalman filter, providing insights as to how the filter performs.

D. An Alternative Kalman Filter

To examine $\boldsymbol{\Sigma}_k$'s lack of dependency on the previous state's posterior covariance, we explored a variation of the previously introduced robust Kalman filter. In this version, we did not perform a full Bayesian treatment of the weighted Kalman filter. Instead, we use the standard Kalman filter equations, (2) to (7), and modify (4) in an ad hoc manner so that the output variance for $\mathbf{z}_k$, $\mathbf{R}_k$, is now weighted (as in our original model in (8)):

$$\mathbf{S}'_k = \left(\mathbf{C}_k \boldsymbol{\Sigma}'_k \mathbf{C}_k^T + \frac{1}{\langle w_k \rangle}\mathbf{R}_k\right)^{-1} \tag{27}$$

We learn the weights $\langle w_k \rangle$ using (12) from the robust Kalman filter. Additionally, we estimate the system dynamics ($\mathbf{C}$, $\mathbf{A}$, $\mathbf{R}$ and $\mathbf{Q}$) at each time step using a maximum likelihood framework (i.e., using (17) to (20) from the robust Kalman filter). In this alternative filter, $\boldsymbol{\Sigma}_k$ is still a function of $\boldsymbol{\Sigma}_{k-1}$. While this filter is unprincipled and somewhat arbitrarily derived, we introduce it in order to examine the effect of this dependency on the previous state's covariance in our experiments.

III. Experimental Results

We evaluated our weighted robust Kalman filter on synthetic and robotic data sets and compared it with three other filters: i) the standard Kalman filter, ii) the alternative weighted Kalman filter introduced in Section II-D, and iii) a Kalman filter where outliers are determined by thresholding on the Mahalanobis distance. If the Mahalanobis distance is less than a certain threshold value, then it is considered an inlier and processed. Otherwise, it is an outlier and ignored. This threshold value is hand-tuned manually in order to find the optimal value for a particular data set. Given prior knowledge of the data set, this thresholded Kalman filter gives near-optimal performance.

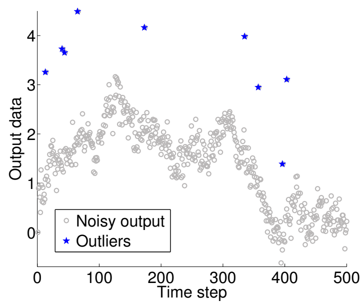
First, we evaluate all four methods on a synthetic data set where the system dynamics ($\mathbf{C}$, $\mathbf{A}$, $\mathbf{R}$ and $\mathbf{Q}$) of the generative model are known. Then, we try to simulate a real data set where the hidden states are unknown and only access to observed data is available. Although they are linear, Kalman filters are commonly used to track more interesting "nonlinear" behaviors (i.e., not just a straight line). For this reason, we try the methods on a synthetic data set exhibiting nonlinear behavior, where the system dynamics are unknown.

For this paper and these experiments, we are interested in the prediction of the observed (output) data and detecting outliers in the observations. We are not interested in the estimation of the system dynamics or in the estimation (or outlier detection) of the states. Estimation of the system matrices is a parameter identification problem and is not addressed in this paper. Similarly, detecting outliers in the states in a different problem and left to another paper Finally, we run all filters on data collected from a robotic dog, LittleDog, manufactured by Boston Dynamics Inc. (Cambridge, MA).

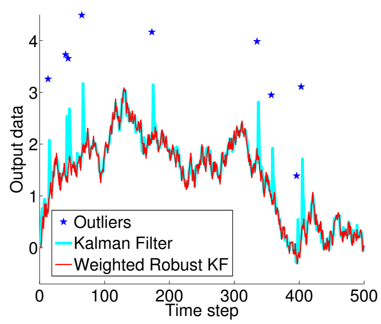
A. Synthetic Data with Known System Dynamics

We generated a data set using (1), using known values for the system dynamics $\mathbf{C}$, $\mathbf{A}$, $\mathbf{R}$ and $\mathbf{Q}$. A one-dimensional data set was generated for ease of visualization, but the following observations and results hold for multi-dimensional data. Figure 1(a) shows observed noisy output data with outliers, collected over 500 time steps (assuming 1 data sample/time step). Data samples were outliers with 1% probability, and $\mathbf{C} = 1, \mathbf{A} = 1, \mathbf{R} = 0.2, \mathbf{Q} = 0.1$. These system matrices were learnt by both the weighted robust Kalman filter and the alternative filter proposed in Section II-D. In contrast, the true $\mathbf{C}$, $\mathbf{A}$, $\mathbf{R}$ and $\mathbf{Q}$ values were used in the standard Kalman filter and thresholded Kalman filter.

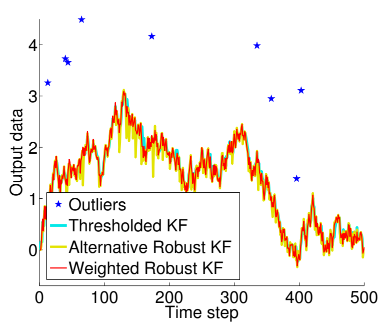
Figure 1(b) shows how sensitive the standard Kalman filter (in the light colored line) is to outliers, while our weighted filter is more robust to outliers. Furthermore, Figure 1(c) compares our filter with the alternative Kalman filter and the thresholded Kalman filter. All three filters appear to detect outliers equally well. Figure 1(d) monitors the residual prediction error for the weighted robust Kalman filter and demonstrates how the residual error remains within the $\pm 3\sigma$ bounds at all times. We omit the residual error plots for the three other filters due to lack of space, but the graphs show similar results.



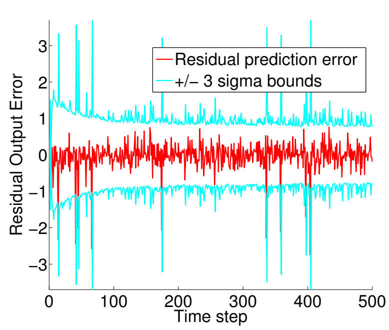
(a) Observed noisy output data with outliers



(b) Predicted data for the Kalman filter and the weighted robust Kalman filter

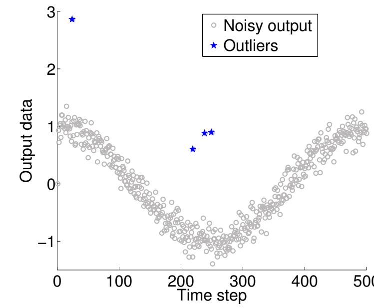


(c) Predicted data for the thresholded Kalman filter, alternative Kalman filter and weighted robust Kalman filter. All three filters perform similarly.

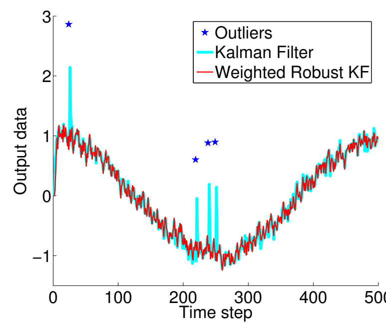


(d) Residual prediction error for the weighted robust Kalman filter

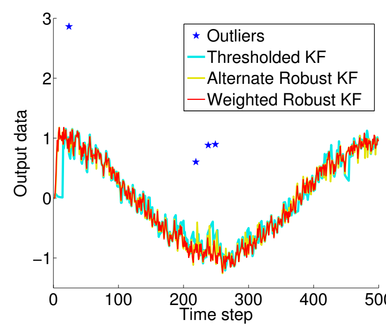
Fig. 1. One-dimensional synthetic output data with noise & outliers (and known system dynamics, $\mathbf{C}$, $\mathbf{A}$, $\mathbf{R}$ and $\mathbf{Q}$) for 500 samples at 1 sample/time step



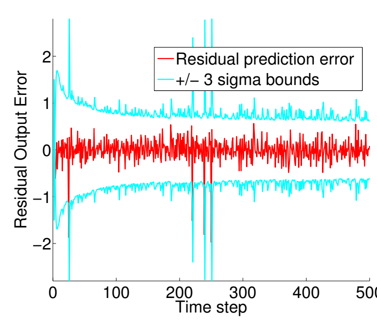
(a) Observed noisy output data with outliers



(b) Predicted data for the Kalman filter and the weighted robust Kalman filter



(c) Predicted data for the thresholded Kalman filter, alternative Kalman filter and weighted robust Kalman filter. All three filters perform similarly.



(d) Residual prediction error for the weighted robust Kalman filter

Fig. 2. One-dimensional data showing a cosine function with noise & outliers (and unknown system dynamics) for 500 samples at 1 sample/time step

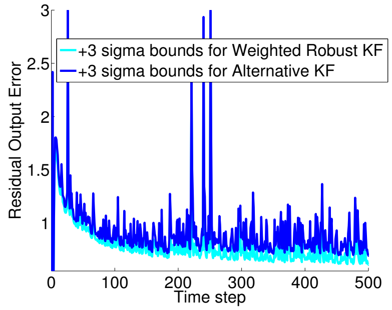


Fig. 3. $+3\sigma$ bounds for the weighted robust Kalman filter and alternative Kalman filter

B. Synthetic Data with Unknown System Dynamics

Next, we created data exhibiting nonlinear behavior, where $\mathbf{C}$, $\mathbf{A}$, $\mathbf{R}$, $\mathbf{Q}$ and states are unknown, high noise[2] (i.e., $r^2 = 0.9$) is added to the (output) data, and a data sample is an outlier with 1% probability. Again, one-dimensional data is used for ease of visualization, and Figure 2(a) shows a noisy cosine function with outliers, over 500 time steps. For optimal performance, $\mathbf{C}$, $\mathbf{A}$, $\mathbf{R}$ and $\mathbf{Q}$ were manually tuned for the standard Kalman filter—a tricky and time-consuming process. In contrast, the system dynamics were learnt for the thresholded Kalman filter using a maximum likelihood framework (i.e. using (17) to (20) without any weights).

Figure 2(b) shows how sensitive the standard Kalman filter is to outliers, while the weighted robust Kalman filter seems to detect them quite well. In Figure 2(c), we compare the weighted robust Kalman filter with the alternative filter and thresholded filter. All three filters appear to perform as well, which is unsurprising, given the amount of manual tuning required by the thresholded Kalman filter.

Figure 2(d) shows that the residual prediction error on the outputs stays within the $\pm 3\sigma$ bounds. In Figure 3, we can see that the covariance of the residual error is slightly smaller for the weighted robust filter (i.e. we are slightly more confident in our estimates for the weighted robust filter). This, in turn, translates to a slightly higher Kalman gain, K'_k, for the alternative filter (this is easily seen by plotting both Kalman gains). A higher K'_k means that more consideration is given to the sample $\mathbf{z}_k$ when estimating the current time step's hidden state.

C. LittleDog Robot

We evaluated all filters on a 12 degree-of-freedom (DOF) robotic dog, LittleDog, shown in Figure 4. The robot dog has two sources that measure its orientation: a motion capture (MOCAP) system and an on-board inertia measurement unit (IMU). Both provide a quaternion q of the robot's orientation: q_{MOCAP} from the MOCAP and q_{IMU} from the IMU.

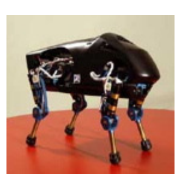
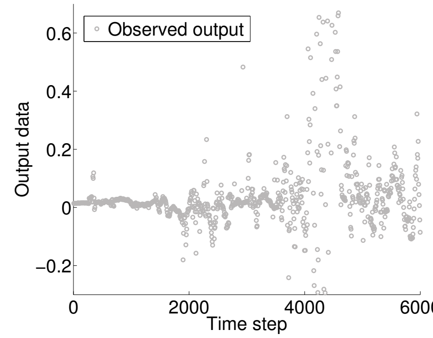


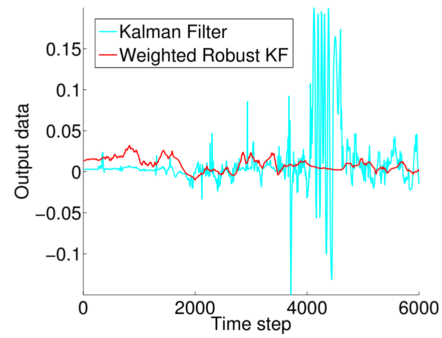
Fig. 4. LittleDog (Boston Dynamics)

q_{IMU} drifts over time, since the IMU cannot provide stable orientation estimation but its signal is clean. The drift that occurs in the IMU is quite common in systems where sensors collect data that need to be integrated. In contrast, q_{MOCAP} has outliers and noise, but no drift. We would like to estimate the offset between q_{MOCAP} and q_{IMU}, and this offset is a noisy slowly drifting signal containing outliers. For optimal performance, we, once again, manually tuned $\mathbf{C}$, $\mathbf{A}$, $\mathbf{R}$ and $\mathbf{Q}$ for the standard Kalman filter. The system dynamics of the thresholded Kalman filter were learnt, and its threshold parameter was manually tuned for best performance on this

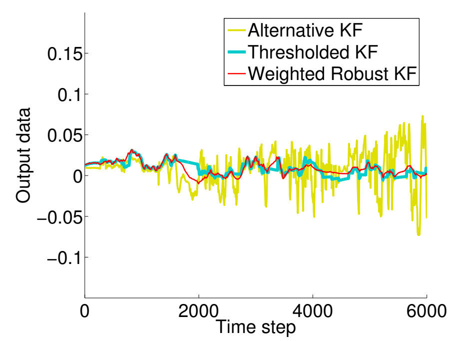
[2]Noise is parameterized by the coefficient of determination, r^2. Noise is added, scaled to the output variance (i.e., $\sigma_{\text{noise}} = c\sigma_y$), where $c = \sqrt{1/r^2 - 1}$.



(a) Observed data, $(q_{\text{IMU}} - q_{\text{MOCAP}})$, from LittleDog robot: a slowly drifting noisy signal with outliers



(b) Predicted data for the Kalman filter and weighted robust Kalman filter. Note the change of scale in axis from Figure 5(a).



(c) Predicted data for the thresholded Kalman filter, alternative Kalman filter and weighted robust Kalman filter

Fig. 5. Observed vs. predicted data from LittleDog robot shown for all Kalman filters, over 6000 samples

Preprint submitted to 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. Received April 8, 2007.

data set.

Figure 5(a) shows the offset data between q_{MOCAP} and q_{IMU} for one of the four quaternion coefficients, collected over 6000 data samples, at 1 sample/time step. As expected, the standard Kalman filter fails to detect and ignore the outliers occurring between the 4000th and 5000th sample, as seen in Figure 5(b). When comparing our weighted robust Kalman filter with the other remaining two filters, Figure 5(c) shows that the thresholded Kalman filter does not react as violently as the standard Kalman filter to outliers and, in fact, appears to perform similarly to the weighted robust Kalman filter. This is to be expected, given that we hand-tuned the threshold parameter for optimal performance. Notice that the weighted robust filter does not track noise in the data as closely as the alternative filter. This is a direct result of higher Kalman gains in the alternative filter and a consequence of the dependency on the previous state state's covariance.

IV. CONCLUSIONS AND FUTURE WORKS

We derived a novel Kalman filter that is robust to outliers by using a weighted least squares approach and introducing weights for each data sample. This Kalman filter learns the weights, as well as the system dynamics, without any need for parameter tuning by the user, heuristics or sampling. We compared this algorithm with other robust approaches and demonstrated the effectiveness of this robust Kalman filter on synthetic and robotic data. The filter was able to perform as well as a hand-tuned approach (that required prior knowledge of the data), without the need for parameter tuning. It offers a competitive, easy-to-use alternative for filtering sensor data.

V. ACKNOWLEDGMENTS

This research was supported in part by National Science Foundation grants ECS-0325383, IIS-0312802, IIS-0082995, ECS-0326095, ANI-0224419, a NASA grant AC#98 − 516, an AFOSR grant on Intelligent Control, the ERATO Kawato Dynamic Brain Project funded by the Japanese Science and Technology Agency, and the ATR Computational Neuroscience Laboratories.

REFERENCES

[1] D. S. Moore and G. P. McCabe. *Introduction to the Practice of Statistics*. W.H. Freeman & Company, March 1999.

[2] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.

[3] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Wiley, 2001.

[4] R. E. Kalman. A new approach to linear filtering and prediction problems. *In Transactions of the ASME - Journal of Basic Engineering*, 183:35–45, 1960.

[5] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering, Transactions ASME, Series D*, 83:95–108, 1961.

[6] J. M. Morris. The Kalman filter: A robust estimator for some classes of linear quadratic problems. *IEEE Transactions on Information Theory*, 22:526–534, 1976.

[7] J. W. Tukey. A survey of sampling from contaminated distributions. In I. Olkin, editor, *Contributions to Probability and Statistics*, pages 448–485. Stanford University Press, 1960.

[8] P. J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35:73–101, 1964.

[9] H. W. Sorensen and D. L. Alspach. Recursive Bayesian estimation using Gaussian sums. *Automatica*, 7:467–479, 1971.

[10] M. West. Robust sequential approximate Bayesian estimation. *Journal of the Royal Statistical Society, Series B*, 43:157–166, 1981.

[11] M. West. *Aspects of Recursive Bayesian Estimation*. PhD thesis, Dept. of Mathematics, University of Nottingham, 1982.

[12] A. F. M. Smith and M. West. Monitoring renal transplants: an application of the multiprocess Kalman filter. *Biometrics*, 39:867–878, 1983.

[13] R. J. Meinhold and N. D. Singpurwalla. Robustification of Kalman filter models. *Journal of the American Statistical Association*, pages 479–486, 1989.

[14] C. Masreliez. Approximate non-Gaussian filtering with linear state and observation relations. *IEEE Transactions on Automatic Control*, 20:107–110, 1975.

[15] C. Masreliez and R. Martin. Robust Bayesian estimation for the linear model and robustifying the Kalman filter. *IEEE Transactions on Automatic Control*, 22:361–371, 1977.

[16] I. C. Schick and S. K. Mitter. Robust recursive estimation in the presence of heavy-tailed observation noise. *Annals of Statistics*, 22(2):1045–1080, 1994.

[17] G. Kitagawa. Non-Gaussian state-space modeling of nonstationary time series. *Journal of the American Statistical Association*, 82:1032–1063, 1987.

[18] S. C. Kramer and H. W. Sorenson. Recursive Bayesian estimation using piece-wise constant approximations. *Automatica*, 24(6):789–801, 1988.

[19] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state-space models. *Journal of the American Statistical Association*, 93:1203–1215, 1996.

[20] G. Kitagawa and W. Gersch. Smoothness priors analysis of time series. In *Lecture Notes in Statistics*. Springer-Verlag, 1996.

[21] T. P. Ryan. *Modern Regression Methods*. Wiley, 1997.

[22] P. J. Huber. *Robust Statistics*. Wiley, 1973.

[23] Z. M. Durovic and B. D. Kovacevic. Robust estimation with unknown noise statistics. *IEEE Transactions on Automatic Control*, 44:1292–1296, 1999.

[24] S. C. Chan, Z. G. Zhang, and K. W. Tse. A new robust Kalman filter algorithm under outliers and system uncertainties. In *IEEE International Symposium on Circuits and Systems*, pages 4317–4320. IEEE, 2005.

[25] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society. Series B*, 39(1):1–38, 1977.

[26] K. A. Myers and B. D. Tapley. Adaptive sequential estimation with unknown noise statistics. *IEEE Transactions on Automatic Control*, 21:520–523, 1976.

[27] A. Gelman, J. Carlin, H.S. Stern, and D.B. Rubin. *Bayesian Data Analysis*. Chapman and Hall, 2000.

[28] J. Ting, A. D'Souza, and S. Schaal. Automatic outlier detection: A Bayesian approach. In *IEEE International Conference on Robotics and Automation*, 2007.

[29] Z. Ghahramani and M.J. Beal. Graphical models and variational methods. In D. Saad and M. Opper, editors, *Advanced Mean Field Methods - Theory and Practice*. MIT Press, 2000.

[30] Z. Ghahramani and G. Hinton. Parameter estimation for linear dynamical systems. Technical report, University of Toronto, 1996.

[31] R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. MIT Press, 1999.

[32] Peter S. Maybeck. *Stochastic models, estimation, and control*, volume 141 of *Mathematics in Science and Engineering*. Academic Press, 1979.