A Kalman Filter for Robust Outlier Detection

Jo-Anne Ting, Evangelos Theodorou, Stefan Schaal

Computational Learning & Motor Control Lab
University of Southern California

IROS 2007
October 31, 2007

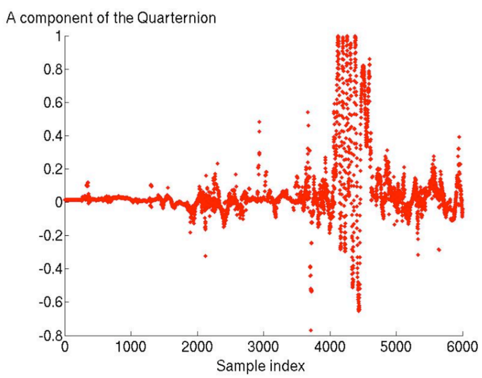
Outline

- Motivation
- Quick review of the Kalman filter
- Robust Kalman filtering with Bayesian weights
- Experimental results
- Conclusions

Motivation

• Consider real-time applications where storing data samples may not be a viable option due to high frequency of sensory data

• In systems where high quality sensory data is needed, reliable detection of outliers is essential for optimal performance (e.g. legged locomotion):



• The Kalman filter (Kalman, ’60) is commonly used for real-time tracking, but it is not robust to outliers!

Previous Methods

Robust Kalman filter approach	Drawback
1) Use non-Gaussian distributions for random variables (Sorenson & Alspach ’71, West ’82)	Complicated resulting parameter estimation for systems with transient disturbances
2) Model observation & state noise as non-Gaussian, heavy-tailed distributions (Masreliez ’75)	Difficult & involved filter implementation
3) Use resampling or numerical integration (Kitagawa ’87)	Heavy computation not suitable for real-time applications
4) Use a robust least squares approach & model weights with heuristic functions (e.g., Durovic & Kovacevic, ’99)	Need to determine the optimal values of open parameters

Outline

- Motivation
- Quick review of the Kalman filter
- Robust Kalman filtering with Bayesian weights
- Experimental results
- Conclusions

A Quick Review of the Kalman Filter

- The system equations for the Kalman filter are as follows:

$$\mathbf{z}_k = \mathbf{C}\theta_k + \mathbf{v}_k$$

$$\theta_k = \mathbf{A}\theta_{k-1} + \mathbf{s}_k$$

Observation matrix: $\mathbf{C}$

State transition matrix: $\mathbf{A}$

Observation noise: $\mathbf{v}_k \sim \mathrm{Normal}(0, \mathbf{R})$

State noise: $\mathbf{s}_k \sim \mathrm{Normal}(0, \mathbf{Q})$

Standard Kalman Filter Equations

Propagation:

$$\theta_k^{'} = \mathbf{A}\left\langle \theta_{k-1} \right\rangle$$

$$\Sigma_k^{'} = \mathbf{A}\Sigma_{k-1}\mathbf{A}^T + \mathbf{Q}$$

Update:

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

$$K_k^{'} = \Sigma_k^{'}\mathbf{C}^T\mathbf{S}_k^{'}$$

$$\left\langle \theta_k \right\rangle = \theta_k^{'} + K_k^{'}\left(\mathbf{z}_k - \mathbf{C}\theta_k^{'}\right)$$

$$\Sigma_k = \left(\mathbf{I} - K_k^{'}\mathbf{C}\right)\Sigma_k^{'}$$

Can use ML framework to estimate system dynamics (Myers & Tapley, 1976)

Outline

- Motivation
- Quick review of the Kalman filter
- Robust Kalman filtering with Bayesian weights
- Experimental results
- Conclusions

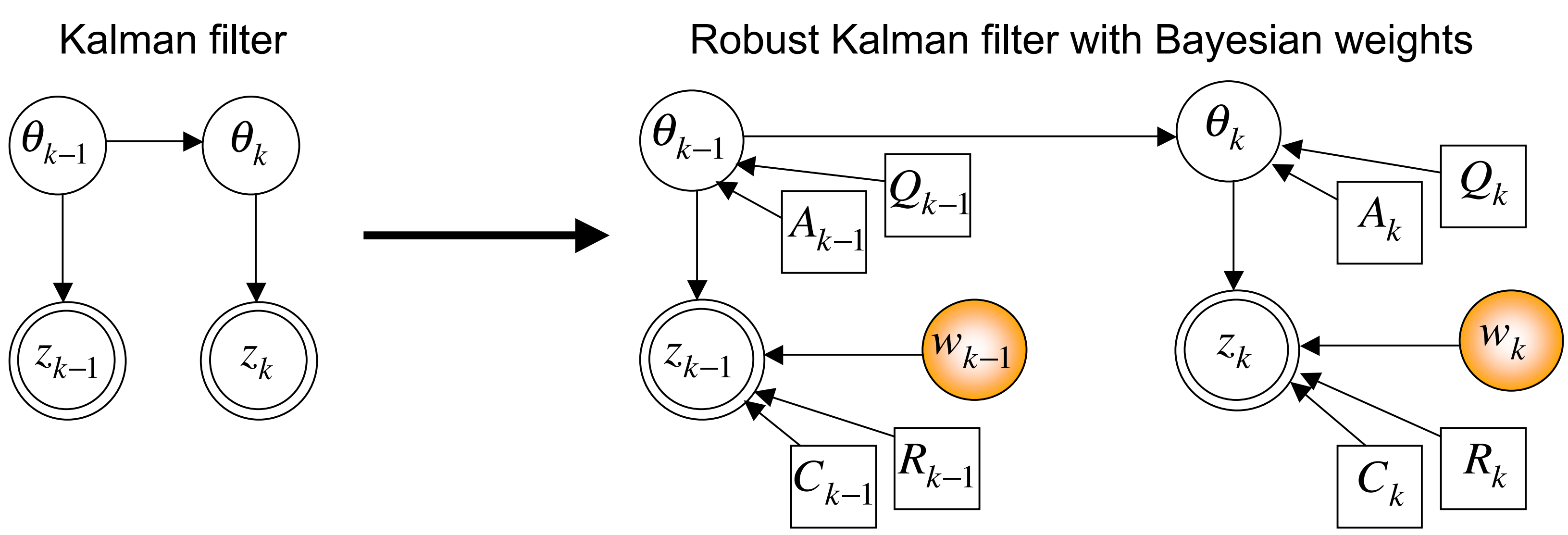
Robust Kalman Filtering with Bayesian Weights

- Use a weighted least squares approach & learn the optimal weights:

$$\mathbf{z}_k | \theta_k, w_k \sim \text{Normal}\left(\mathbf{C}\theta_k, \mathbf{R} / w_k\right)$$

$$\theta_k | \theta_{k-1} \sim \text{Normal}\left(\mathbf{A}\theta_{k-1}, \mathbf{Q}\right)$$

$$w_k \sim \text{Gamma}\left(a_{w_k}, b_{w_k}\right)$$



Inference Procedure

• We can treat this as an EM learning problem (Dempster & Laird, '77):

$$\text{Maximize } \log \prod_{i=1}^{N} p\left(\theta_{1:k}, \mathbf{z}_i, w_{1:k}\right)$$

• We use a variational factorial approximation of the true posterior distribution:

$$\mathrm{Q}\left(w,\theta\right) = \prod_{i=1}^{N} \mathrm{Q}\left(w_i\right) \prod_{i=1}^{N} \mathrm{Q}\left(\theta_i \middle| \theta_{i-1}\right) \mathrm{Q}\left(\theta_0\right)$$

to get analytically tractable inference (e.g., Ghahramani & Beal, '00).

Robust Kalman Filter Equations

Propagation:

$$\theta_k' = \mathbf{A}_k \langle \theta_{k-1} \rangle$$

$$\Sigma_k' = \mathbf{Q}_k$$

Update:

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

$$K_k' = \Sigma_k' \mathbf{C}_k^T \mathbf{S}_k'$$

$$\langle \theta_k \rangle = \theta_k' + K_k' \left(\mathbf{z}_k - \mathbf{C}_k \theta_k' \right)$$

$$\Sigma_k = \left(\mathbf{I} - K_k' \mathbf{C}_k \right) \Sigma_k'$$

$$\langle w_k \rangle = \frac{a_{w_{k0}} + \frac{1}{2}}{b_{w_{k0}} + \left\langle \left(\mathbf{z}_k - \mathbf{C}_k \theta_k \right)^T \mathbf{R}_k^{-1} \left(\mathbf{z}_k - \mathbf{C}_k \theta_k \right) \right\rangle}$$

Compare to standard Kalman filter

Propagation:

$$\theta_k' = \mathbf{A} \langle \theta_{k-1} \rangle$$

$$\Sigma_k' = \mathbf{A} \Sigma_{k-1} \mathbf{A}^T + \mathbf{Q}$$

Update:

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

$$K_k' = \Sigma_k' \mathbf{C}^T \mathbf{S}_k'$$

$$\langle \theta_k \rangle = \theta_k' + K_k' \left(\mathbf{z}_k - \mathbf{C} \theta_k' \right)$$

$$\Sigma_k = \left(\mathbf{I} - K_k' \mathbf{C} \right) \Sigma_k'$$

Important Things to Note

- Our robust Kalman filter:

1) Has the same computational complexity as the standard Kalman filter

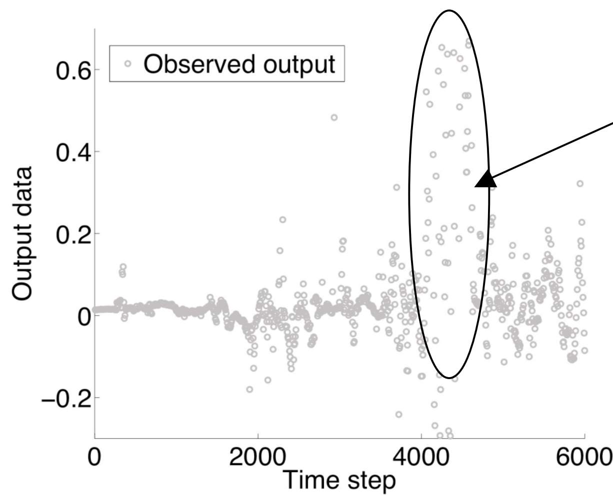
2) Is principled & easy to implement (no heuristics)

3) Offers a natural framework to incorporate prior knowledge of the presence of outliers

Outline

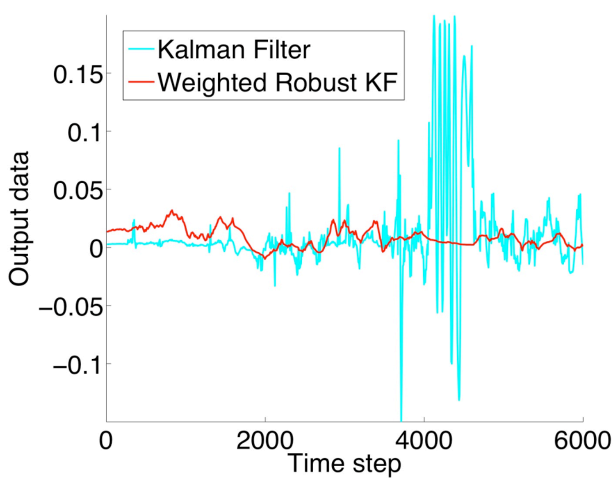
- Motivation
- Quick review of the Kalman filter
- Robust Kalman filtering with Bayesian weights
- Experimental results
- Conclusions

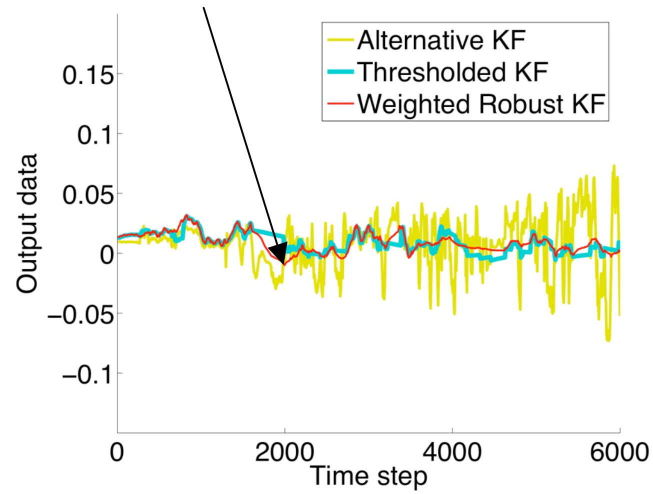
Real-time Outlier Detection on LittleDog



Outliers

Our robust KF performs as well as a hand-tuned KF (that required prior knowledge and, hence, is near-optimal)





Outline

- Motivation
- Quick review of the Kalman filter
- Robust Kalman filtering with Bayesian weights
- Experimental results
- Conclusions

Conclusions

- We have introduced an outlier-robust Kalman filter that:

1) Is principled & easy to implement
2) Has the same computational complexity as the Kalman filter
3) Provides a natural framework to incorporate prior knowledge of noise

- This framework can be extended to other more complex, nonlinear filters & methods in order to incorporate automatic outlier detection abilities.

Final Posterior EM Update Equations

E-step:

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

$$\langle \theta_k \rangle = \Sigma_k \left(\mathbf{Q}_k^{-1} \mathbf{A}_k \langle \theta_{k-1} \rangle + \langle w_k \rangle \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{z}_k \right)$$

$$\langle w_k \rangle = \frac{a_{w_{k0}} + \frac{1}{2}}{b_{w_{k0}} + \left\langle \left(\mathbf{z}_k - \mathbf{C}_k \theta_k \right)^T \mathbf{R}_k^{-1} \left(\mathbf{z}_k - \mathbf{C}_k \theta_k \right) \right\rangle}$$

M-step:

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

$$\mathbf{A}_k = \left(\sum_{i=1}^{k} \langle \theta_i \rangle \langle \theta_{i-1} \rangle^T \right) \left(\sum_{i=1}^{k} \langle \theta_{i-1} \theta_{i-1}^T \rangle \right)^{-1}$$

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

$$q_{kn} = \frac{1}{k} \sum_{i=1}^{k} \langle w_i \rangle \left\langle \left(\theta_i - \mathbf{A}_k(n,:)\theta_{i-1} \right)^2 \right\rangle$$

Need to be written in incremental form

These are computed once for each time step *k* (e.g., Ghahramani & Hinton, 1996)

Incremental Version of M-step Equations

- Gather sufficient statistics to re-write M-step equations in incremental form (i.e., only using values observed or calculated in the current time step, *k):*

M-step:

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

$$\mathbf{A}_k = \sum\nolimits_k^{\theta\theta^T} \left(\sum\nolimits_k^{\theta'\theta'} \right)^{-1}$$

$$r_{km} = \frac{1}{k}\left[\sum\nolimits_{km}^{wzz} - 2\mathbf{C}_k(m,:)\sum\nolimits_{km}^{wz\theta} + diag\left\{ \mathbf{C}_k(m,:)\sum\nolimits_k^{w\theta\theta^T} \mathbf{C}_k(m,:)^T \right\} \right]$$

$$q_{kn} = \frac{1}{k}\left[\sum\nolimits_{kn}^{\theta^2} - 2\mathbf{A}_k(n,:)\sum\nolimits_{kn}^{\theta\theta'} + diag\left\{ \mathbf{A}_k(n,:)\sum\nolimits_k^{\theta'\theta'} \mathbf{A}_k(n,:)^T \right\} \right]$$