An Optimization Approach to Adaptive Kalman Filtering

M. Karasalo and X. Hu

Abstract

In this paper, an optimization-based adaptive Kalman filtering method is proposed. The method produces an estimate of the process noise covariance matrix Q by solving an optimization problem over a short window of data. The algorithm recovers the observations $h(x)$ from a system $\dot{x} = f(x)$, $y = h(x) + v$ without a priori knowledge of system dynamics. Potential applications include target tracking using a network of nonlinear sensors, servoing, mapping, and localization. The algorithm is demonstrated in simulations on a tracking example for a target with coupled and nonlinear kinematics. Simulations indicate superiority over a standard MMAE algorithm for a large class of systems.

Keywords: Adaptive filtering, optimization, tracking

B.1 Introduction

In mobile robotics and sensor networks the need for adaptive and robust filtering methods is increasing with the growing use of large networks of cheap mobile agents with simple sensors. For instance, [18] and [19] discuss the application of adaptive filtering to localization of mobile robots. Examples in the field of visual servoing include [20] where an adaptive extended Kalman filter is applied to improve pose estimates, and the recent contributions in [21], where a similar approach is used to enhance robustness of position and orientation estimates of moving 3D objects.

A thorough treatment of the fundamentals of Kalman filtering can for instance be found in the classic series [1–3] where some adaptive methods of filtering are also outlined. The adaptive Kalman filtering schemes most frequently found in the literature are Innovation-based Adaptive Estimation (IAE) and Multiple Model Adaptive Estimation (MMAE). A neat summary of both strategies is found in [4]. IAE methods estimate the covariance matrix of the process noise Q and/or the measurement noise R and utilize the fact that for the right values of Q and R the innovation sequence of the Kalman filter is white noise. By tuning Q and/or R and studying the resulting innovation sequence one can get an idea of the appropriate values of the covariance matrices. An early example of IAE is found in [5], while more recent attempts are for instance [15] and [16]. However convergence to the "right" values of Q and R is not guaranteed with IAE and most algorithms require estimation made over rather large windows of data to achieve reliable covariance measurements, making the method impractical for rapidly changing systems.

MMAE methods handle model uncertainty by implementing a bank of several different models and computing the bayesian probability for each model to be the true system model given the measurement sequence and under the assumption that one of the models in the model bank is the correct one. The state estimate can be either the output of the most probable model or a weighted sum of the outputs of all models. This method is suitable for applications such as fault detection where one has some a priori information on the system dynamics. For instance, if the dynamics of an engine is well known, each model in the bank can represent the engine dynamics if one or several components fail. If the probability of one of the failure models gets to high an alarm is raised. Examples of MMAE algorithms are found in for instance [17] and [22].

Since MMAE works under the assumption that one of the models in the model bank is the right one, it is an unsuitable method for systems with unknown dynamics. It has been shown in the series of papers [7–13] that using too large a model bank actually decreases performance since, in that case, the true model gets too much competition from false models. Therefore safeguarding with a lot of plausible system models is not an option. Attempts with dynamic model banks have been made in [7–13] and [14], but to the extent of our knowledge all such algorithms assume one large model bank, hopefully including the true model (which may vary over time), and then study different subsets of this large bank. Therefore some way of guessing possible models beforehand is still needed. Convergence results for MMAE methods in general are still lacking.

In this paper we propose an optimization approach to adaptive Kalman filtering, Optimization-based Adaptive Estimation (OAE), that estimates Q componentwise for measurable states at each time step. The method is based on solving an optimization problem to find the Q - value that minimizes both bias and oscillation of the state estimate. The approach is scalable and requires no a priori knowledge of the system dynamics. Q is computed over short windows of data, making the algorithm suitable for rapidly changing systems and online applications. Intended applications include path following, servoing, localization and tracking of a target or signal with unknown dynamics.

The paper is organized as follows. In Section B.2 we introduce notation and review the basics of Kalman filtering. Section B.3 derives the OAE method, which is the main contribution of this paper. In Section B.4 we evaluate OAE in simulation compared with an MMAE approach. Finally, Section B.5 summarizes the paper and provides concluding remarks.

B.2 Preliminaries

Consider a linear discrete-time system

$$x(k+1) = Ax(k) + Bw(k), \quad (\text{B.1})$$

where $x(k) \in \mathbb{R}^n$, and a measurement $y(k) \in \mathbb{R}^m$ governed by

$$y(k) = Cx(k) + v(k), \quad (\text{B.2})$$

with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times s}$ and $C \in \mathbb{R}^{m \times n}$. The signals $w(k)$ and $v(k)$ are random process noise and measurement noise, respectively, and are assumed to be independent, gaussian noise with covariance matrices

$$\begin{aligned} Q(k) &\triangleq E(w(k)w(k)^T) \\ R(k) &\triangleq E(v(k)v(k)^T). \end{aligned} \quad (\text{B.3})$$

Here, $E(\cdot)$ is the expectation value of $(\cdot)$ and $Q(k) \in \mathbb{R}^{s \times s}$, $R(k) \in \mathbb{R}^{m \times m}$.

The Kalman filter is an observer that gives an optimal estimate $\hat{x}(k)$ of the state $x(k)$ at time step k , given the estimate $\hat{x}(k-1)$ and the observation $y(k)$, in a least squares sense. In other words, if we define the estimation error as

$$e(k) \triangleq x(k) - \hat{x}(k), \quad (\text{B.4})$$

then the Kalman filter will give the estimate $\hat{x}(k)$ that minimizes $E(e(t_k)^T e(t_k))$. Now let

$$P(k) \triangleq E(e(k)e(k)^T) \quad (\text{B.5})$$

denote the covariance matrix of $e(k)$. Then, given initial estimates $\hat{x}_0$ and P_0 , the discrete Kalman filter is the recursive process

$$\begin{aligned} \hat{x}(k)^- &= A\hat{x}(k-1) \\ P(k)^- &= AP(k-1)A^T + BQ(k-1)B^T \\ K(k) &= P(k)^-C^T (CP(k)^-C^T + R(k))^{-1} \\ \hat{x}(k) &= \hat{x}(k)^- + K(k)(y(k) - C\hat{x}(k)^-) \\ P(k) &= (I - K(k)C)P(k)^-. \end{aligned} \quad (\text{B.6})$$

$R(k)$ and $Q(k)$ play important roles in the recursions. Convergence to the optimal estimate $\hat{x}(k)$ requires correct values of both matrices. In this paper, we assume that $R(k)$ is known and focus on the estimation of $Q(k)$. Basically, too small values of the components of $Q(k)$ yields a biased estimate, while too large values yields an estimate that oscillates around the true state. In the next section we derive the OAE method that finds a $Q(k)$ value that locally minimizes both bias and oscillations.

B.3 An Optimization Approach To Finding Q

Consider the system

$$\begin{aligned} \dot{x} &= f(x) \\ y &= h(x) + v, \end{aligned} \quad (\text{B.7})$$

where $f(x)$ represents the system dynamics which are assumed unknown, $h(x) = (h_1(x) \dots h_N(x))^T$ is the measurement vector and v is a gaussian measurement noise with unknown covariance. In this section, we derive the OAE approach to Kalman filtering, which recovers $h(x) = (h_1(x) \dots h_N(x))^T$, and possibly their derivatives. From now on we use the notation $h(x) = (h_1 \dots h_N)^T$ for brevity. $f(x)$ and $h(x)$ of (B.7) may be nonlinear and coupled but we will use n -dimensional linear systems of the type (B.1) - (B.2) as a system model for each component h_i , $i = 1, \dots, N$, and adaptively design filters for each of the N linear systems separately. Associated with each such linear system, we define a state vector $\mathbf{h}_i = (\mathbf{h}_i^1 \dots \mathbf{h}_i^n)^T$ such that $\mathbf{h}_i^1 \triangleq h_i$. Denoting the output of system i by $\mathbf{y}_i$, we get, for $i = 1, \dots, N$,

$$\begin{aligned} \mathbf{h}_i(k+1) &= A_i \mathbf{h}_i(k) + B_i w_i(k) \\ \mathbf{y}_i(k) &= C_i \mathbf{h}_i(k) + v_i(k), \end{aligned} \quad (\text{B.8})$$

where $A_i \in \mathbb{R}^{n \times n}$, $B_i \in \mathbb{R}^{n \times s}$ and $C_i \in \mathbb{R}^{m \times n} \triangleq (1 \ 0 \dots 0)$, so that $C_i \mathbf{h}_i = \mathbf{h}_i^1 = h_i$.

The unknown properties of the underlying system (B.7), such as coupling and nonlinearity, are modeled as an added scalar gaussian noise $w_i(k)$. As $Q_i(k)$ is the covariance of $w_i(k)$, each of the Kalman filters has a scalar covariance matrix Q_i to be estimated. Viewing the entire N -dimensional vector $h(x)$, this in a sense means assuming a diagonal covariance matrix. This is an assumption frequently used in the literature.

Modeling the dynamics of each component h_i separately allows for scalability without increasing model complexity. Let us illustrate the concept and the notation with an example.

Example B.3.1 Consider tracking a target in 3D, such as an airplane or helicopter whose planned trajectory is unknown but whose position $(x_1 \ x_2 \ x_3)^T$ can be observed ($N = 3$). Applying the OAE approach, we would have a measurable state vector $h(x) = (h_1(x) \ h_2(x) \ h_3(x))^T = (x_1 \ x_2 \ x_3)^T$ representing the position of the target, and need to model the dynamics of h_1, h_2 and h_3 separately.

Introduce state vectors $\mathbf{h}_i$ and linear systems A_i, B_i, C_i for $i = 1, 2, 3$. For tracking in 1D a natural model for the target dynamics is a random walk model of order $n = 2$ where the noise w represents the unknown acceleration of the target. Therefore, we let (B.8) be realized as

$$\begin{aligned}\mathbf{h}_i(k+1) &= \begin{pmatrix} 1 & \delta \\ 0 & 1 \end{pmatrix} \mathbf{h}_i(k) + \begin{pmatrix} \frac{\delta^2}{2} \\ \delta \end{pmatrix} w_i(k) \\ \mathbf{y}_i(k) &= \begin{pmatrix} 1 & 0 \end{pmatrix} \mathbf{h}_i(k) + v_i(k).\end{aligned}\tag{B.9}$$

Here δ is the sampling rate, $\mathbf{h}_i^1$ corresponds to the position and $\mathbf{h}_i^2$ to the velocity of the target.

OAE computes state estimates $\hat{\mathbf{h}}_i = (\hat{\mathbf{h}}_i^1 \ \hat{\mathbf{h}}_i^2)^T$ for each component $\mathbf{h}_i$ and the final output is the complete estimate $\hat{\mathbf{h}} = (\hat{\mathbf{h}}_1^T \ \hat{\mathbf{h}}_2^T \ \hat{\mathbf{h}}_3^T)^T = (\hat{x}_1 \ \hat{x}_1' \ \hat{x}_2 \ \hat{x}_2' \ \hat{x}_3 \ \hat{x}_3')^T$ of the 3D position and its derivatives.

The idea of OAE is to, at each iteration k , compute an, in some sense, optimal value of $Q_i(k)$ for each component model i , and apply separate Kalman filters to each component. To emphasize that we are dealing with scalars we will from now on use q_i to denote the process noise covariance for $\mathbf{h}_i$. The rest of this section will be devoted to deriving the OAE approach to computing $q_i(k)$.

B.3.1 General Idea of OAE

As mentioned in the preliminaries, a too small estimate of the process noise causes bias while a too large estimate causes oscillation. For very large values of $q_i(k)$, $\hat{h}_i$ may even have a larger mean amplitude of oscillation around the true value h_i than the noisy sample data $\mathbf{y}_i$ so that the Kalman filter actually adds disturbances rather than filtering them out. This is illustrated in Figure 1, where two very different $\hat{h}$'s are obtained for the same trajectory, dynamic model, and data set due to different choices of Q . The ideal choice of $q_i(k)$ lies somewhere between these extremes. We want

Figure 1: Two Kalman estimates (dotted) for the same trajectory and data set. **Left:** $Q = 0$, **right** $Q = 5$.

to find a $q_i(k)$ such that both the bias and the mean amplitude are small. Therefore we define a cost function

$$J_i(q) = \varepsilon a_i(q) + (1 - \varepsilon) b_i(q),\tag{B.10}$$

where $a_i(q)$ and $b_i(q)$ are to be defined further down. $a_i(q)$ is a measure of the amplitude of $\hat{h}_i$, and $b_i(q)$ is a measure of the bias of $\hat{h}_i$, compared to the true value h_i . $\varepsilon \in (0, 1)$ is a scalar weight. Note that $\hat{h}_i$ is a function of q through the Kalman recursions (B.6). As will be seen further down, $a_i(q)$ and $b_i(q)$ will actually be functions of $\hat{h}_i$ and thereby implicit functions of q . Let

$$(q_i(k))^* = \arg \min_{q \geq 0} J_i(q). \quad (\text{B.11})$$

The most critical part is to construct $a_i(q)$ and $b_i(q)$ without prior knowledge of the true vector $h(x)$. In the following our approach is described.

B.3.2 Measure of Bias and Amplitude

First, recall that the output of system i is modeled as $\mathbf{y}_i(k) = \mathbf{h}_i^1(k) + v(k) = h_i(k) + v(k)$. Since $v(k)$ is a zero mean gaussian noise, $\mathbf{y}_i$ is not biased. Therefore $\mathbf{y}_i(k)$ and $\hat{h}_i(k)$ can be compared to measure the bias of $\hat{h}_i(k)$. This is done simply by fitting second order polynomials to $\mathbf{y}_i(\kappa)$ and $\hat{h}_i(\kappa)$, where $\kappa \in [k \pm \Delta]$, where Δ is a positive integer. We define the vectors $f_{\mathbf{y}_i}$ and $f_{\hat{h}_i}$ as the discretizations of these polynomials with sampling points defined by κ . Locally, $f_{\mathbf{y}_i}$ can be considered a non-biased estimate of h while $f_{\hat{h}_i}$ is a smoothing of $\hat{h}_i$ on the interval κ . The bias measure can now be defined as

$$b_i(q) \triangleq \|f_{\mathbf{y}_i} - f_{\hat{h}_i}\|^2, \quad (\text{B.12})$$

where $\|\cdot\|$ is the Euclidean norm. Note that, for large values of q , $\hat{h}_i$ may oscillate rapidly enough for $f_{\mathbf{y}_i} - f_{\hat{h}_i}$ to change sign within the window κ . In such cases, $b_i(q)$ is not a measure of bias but a measure of amplitude of the oscillations. Therefore $b_i(q)$ punishes bias for small values of q and enhances punishment on amplitude for large values of q .

The construction of $a_i(q)$ follows along the same lines as that of $b_i(q)$. Note that if the amplitude of $\hat{h}_i$ is high, the mean distance between $\hat{h}_i$ and $f_{\hat{h}_i}$ is large. Therefore, let

$$a_i(q) \triangleq \|f_{\hat{h}_i} - \hat{h}_i\|^2. \quad (\text{B.13})$$

This choice of bias and amplitude measurements has proved to be robust in simulations and is easy to implement. In the next section it is shown that the resulting expression for $J_i(q)$ is actually a convex quadratic function in $h(q)$. However, from a theoretical viewpoint, other cost functions might be a more reasonable choice. The best choice of measurements $a_i(q)$ and $b_i(q)$ should be investigated further in future work.

If no knowledge of the tracked system is available beforehand, the most natural choice of weight is $\varepsilon = 0.5$. To give an idea of the effect of the terms in $J_i(q)$, we have constructed estimates $\hat{h}$ for the same curve and data set with a few different values of the weight ε . The results are shown in Figure 2. One way to construct a strict bias measure is to set $b_i(q)$ to zero if $f_{\mathbf{y}_i} - f_{\hat{h}_i}$ changes sign on the interval κ . However, the resulting cost function $J_i(q)$ becomes much less appealing, and performance actually deteriorates, as is illustrated in Figure 2.

B.3.3 Derivation of $J_i(q)$

First, expressions for $f_{\mathbf{y}_i}$ and $f_{\hat{h}_i}$ are given. Let $\kappa = 1, \dots, 2\Delta + 1$ and let $\mathbf{y}_i(\kappa)$ be measurement data. A second order polynomial has the form

$$f_{\mathbf{y}_i}(\kappa) = \alpha_0 + \alpha_1 \kappa + \alpha_2 \kappa^2 \quad (\text{B.14})$$

Figure 2: Tracking with OAE using different values of ε . **Top left** : $\varepsilon = 0.5$. **Top right** : $\varepsilon = 1$ yields a smooth but biased estimate. **Bottom left** : $\varepsilon = 0$ yields an oscillating estimate. **Bottom right** : only including $b_i(q)$ if $f_{y_i} - f_{\hat{y}_i}$ does not change sign gives more oscillations.

and the coefficients α_0 , α_1 and α_2 are determined in a least squares sense by minimizing

$$\sum_{\kappa=1}^{2\Delta+1} (\mathbf{y}_i(\kappa) - (\alpha_0 + \alpha_1 \kappa + \alpha_2 \kappa^2))^2. \quad (\text{B.15})$$

Using matrix notation (B.15) takes the form

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ \vdots & \vdots & \vdots \\ 1 & 2\Delta+1 & (2\Delta+1)^2 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} \mathbf{y}_i(1) \\ \vdots \\ \mathbf{y}_i(2\Delta+1) \end{pmatrix}. \quad (\text{B.16})$$

Note that for a general $\kappa = k - \Delta, \dots, k + \Delta$ the matrix in (B.16) may become ill conditioned if k is large. However for the purpose of finding $f_{y_i}(\kappa)$ we can always translate the data set and use the matrix in (B.16) to avoid singularities. Introducing the notation

$$\mathcal{K} \triangleq \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ \vdots & \vdots & \vdots \\ 1 & 2\Delta + 1 & (2\Delta + 1)^2 \end{pmatrix}, \quad \mathbf{y}_i(\kappa) \triangleq \begin{pmatrix} \mathbf{y}_i(1) \\ \vdots \\ \mathbf{y}_i(2\Delta + 1) \end{pmatrix}, \quad (\text{B.17})$$

and noting that $(\alpha_0 \ \alpha_1 \ \alpha_2)^T = (\mathcal{K}^T \mathcal{K})^{-1} \mathcal{K}^T \mathbf{y}_i(\kappa)$ we can write

$$f_{y_i}(\kappa) = \mathcal{K} (\mathcal{K}^T \mathcal{K})^{-1} \mathcal{K}^T \mathbf{y}_i(\kappa) \triangleq \mathbf{K} \mathbf{y}_i(\kappa). \quad (\text{B.18})$$

Note that $\mathcal{K}$ is singular for $\Delta = 0$. From now on it is assumed that $\Delta > 0$, or a minimum of 3 measurements in $\mathbf{y}_i(\kappa)$ for each computation of $J_i(q)$. This yields, with $I \in \mathbb{R}^{(2\Delta+1) \times (2\Delta+1)}$ denoting the identity matrix,

$$a_i(q) = \|(\mathbf{K} - I)\hat{h}_i(\kappa)\|^2, \quad (\text{B.19})$$

$$b_i(q) = \|\mathbf{K}(\hat{h}_i(\kappa) - \mathbf{y}_i(\kappa))\|^2, \quad (\text{B.20})$$

where $\hat{h}_i(\kappa) \triangleq (\hat{h}_i(1) \dots \hat{h}_i(2\Delta + 1))^T$. It is easy to show that $\mathbf{K} \triangleq \mathcal{K} (\mathcal{K}^T \mathcal{K})^{-1} \mathcal{K}^T$ is symmetric and idempotent. After some straightforward calculations (recalling that $\mathbf{y}_i(\kappa)$ is a vector of *given* measurement data and not a function of q , while $\hat{h}_i(\kappa)$ is a function of q as it is the first component of the state vector of the Kalman estimate based on q) the final expression for the cost function (B.10) on the interval κ and for component h_i is

$$J_{i,\kappa}(q) = \frac{1}{2} \hat{h}_i(\kappa)^T \mathbf{M} \hat{h}_i(\kappa) - 2(1 - \varepsilon) \mathbf{y}_i(\kappa)^T \mathbf{K} \hat{h}_i(\kappa) \quad (\text{B.21})$$

with

$$\mathbf{M} \triangleq 2((1 - 2\varepsilon)\mathbf{K} + \varepsilon I). \quad (\text{B.22})$$

For the special case $\varepsilon = 0.5$ we get

$$J_{i,\kappa}(q) = \frac{1}{2} \hat{h}_i(\kappa)^T \hat{h}_i(\kappa) - \mathbf{y}_i(\kappa)^T \mathbf{K} \hat{h}_i(\kappa). \quad (\text{B.23})$$

The particular choices of $a_i(q)$ and $b_i(q)$ yield the following result.

Proposition B.3.1 *The function $J_{i,\kappa}$ as defined by (B.21) is convex in $\hat{h}_i$.*

Proof Note that $J_{i,\kappa}$ is a quadratic function in the argument $\hat{h}_i$. For convexity in $\hat{h}_i$, it suffices that $\mathbf{M}$ is positive definite. For idempotent matrices such as $\mathbf{K}$ it holds that the eigenvalues are either 0 or 1. Since $\mathbf{K}$ is also symmetric, it is positive semidefinite. Now

$$\begin{aligned} x^T \mathbf{M} x &= 2x^T ((1 - 2\varepsilon)\mathbf{K} + \varepsilon I) x = 2(1 - 2\varepsilon) x^T \mathbf{K} x + 2\varepsilon x^T I x \\ &\geq \{(1 - 2\varepsilon) \geq 0, \mathbf{K} \text{ is pos. sem. def.}\} \geq 2\varepsilon x^T I x > 0 \quad \forall x \in \mathbb{R}^{2\Delta+1}. \end{aligned} \quad (\text{B.24})$$

■

It should be noted that Proposition B.3.1 is not sufficient for convexity in q . Since $\hat{h}_i$ as a function of q depends on input data and is defined by the Kalman recursions, the optimization problem is in general not convex in q , and thus in most cases we get a locally optimal solution. Our method can be summarized as follows:

Given Δ and the interval $\kappa = k - \Delta, \dots, k + \Delta$, for each measurement $\mathbf{y}_i(k)$ and each model i let

$$(q_i(k))^* = \arg \min_{q \geq 0} J_{i,\kappa}(q). \quad (\text{B.25})$$

The value $(\hat{h}_i(k))^*$ corresponding to $(q_i(k))^*$ is chosen as the output estimate $\hat{h}_i(k)$.

Next we illustrate how OAE works with an example.

B.3.4 A Simple Example

In this section $J(q)$ is derived for a simple example, assuming $h(x) \in \mathbb{R}$ ($N = 1$) and using a first order one dimensional random walk as a system model ($n = 1$):

$$\begin{aligned} \mathbf{h}(k+1) &= \mathbf{h}(k) + w(k) \\ \mathbf{y}(k) &= \mathbf{h}(k) + v(k). \end{aligned} \quad (\text{B.26})$$

For a scalar state variable $\mathbf{h}$, all matrices of the Kalman filter are scalar and the recursions boil down to

$$\begin{aligned} \hat{\mathbf{h}}(k) &= \frac{(P(k-1) + Q(k-1))(\mathbf{y}(k) - \hat{\mathbf{h}}(k-1))}{P(k-1) + Q(k-1) + R(k-1)} \\ P(k) &= \frac{R(k)(P(k-1) + Q(k-1))}{P(k-1) + Q(k-1) + R(k-1)}. \end{aligned} \quad (\text{B.27})$$

We will give an analytic expression for the cost function $J_{i,\kappa}(q)$ for this example with the shortest possible window, $\kappa = 1, 2, 3$. With initial values $\hat{\mathbf{h}}^0$, P^0 and assuming a constant measurement noise covariance matrix R , we get

$$\mathbf{y}(\kappa) \triangleq \begin{pmatrix} \mathbf{y}^0 \\ \mathbf{y}^1 \\ \mathbf{y}^2 \end{pmatrix}, \quad \mathcal{K} \triangleq \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{pmatrix} \Rightarrow \mathbf{K} = I \quad (\text{B.28})$$

$$\mathbf{h}(\kappa) = \begin{pmatrix} \hat{\mathbf{h}}^0 \\ \frac{(P^0+q)(\mathbf{y}^1-\hat{\mathbf{h}}^0)}{P^0+q+R} \\ \frac{(\mathbf{y}^2(P^0+q+R) - (\mathbf{y}^1-\hat{\mathbf{h}}^0)(P^0+q))(P^0R + (P^0+q+2R)q)}{(P^0+q+R)(2P^0R + (P^0+q+3R)q + R^2)} \end{pmatrix}. \quad (\text{B.29})$$

With $\varepsilon = 0.5$ the explicit expression for the cost function reads

$$\begin{aligned} J(q) &= \frac{1}{2} \hat{\mathbf{h}}(\kappa)^T \hat{\mathbf{h}}(\kappa) - \mathbf{y}(\kappa)^T \hat{\mathbf{h}}(\kappa) = \frac{1}{2} \left((\hat{\mathbf{h}}^0)^2 + \left(\frac{(P^0+q)(\mathbf{y}^1-\hat{\mathbf{h}}^0)}{P^0+q+R} \right)^2 \right. \\ &\quad \left. + \left(\frac{(\mathbf{y}^2(P^0+q+R) - (\mathbf{y}^1-\hat{\mathbf{h}}^0)(P^0+q))(P^0R + (P^0+q+2R)q)}{(P^0+q+R)(2P^0R + (P^0+q+3R)q + R^2)} \right)^2 \right) - \\ &\quad \left(\mathbf{y}^0 \hat{\mathbf{h}}^0 + \frac{\mathbf{y}^1(P^0+q)(\mathbf{y}^1-\hat{\mathbf{h}}^0)}{P^0+q+R} + \frac{\mathbf{y}^2(\mathbf{y}^2(P^0+q+R) - (\mathbf{y}^1-\hat{\mathbf{h}}^0)(P^0+q))(P^0R + (P^0+q+2R)q)}{(P^0+q+R)(2P^0R + (P^0+q+3R)q + R^2)} \right). \end{aligned} \quad (\text{B.30})$$

With a model order bigger than 1, such as for instance (B.9) in Example 1, the Kalman recursions become more complex, and increasing the window size Δ increases the number of terms in the cost

function. In the general case, the cost function $J(q)$ consists of a sum of $2(2\Delta + 1)$ quotients of polynomials in q with increasing degree for larger indices.

As mentioned before, in general $J(q)$ is not convex, and consequently the best we can hope for is a locally optimal solution. However, if we limit the domain to an interval $q \in [q_{min}, q_{max}]$ we can find the optimal solution on the interval with an accuracy determined by the chosen discretization of $[q_{min}, q_{max}]$. Computationally this method requires the use of a small value of Δ and a simple model such as (B.26) to be applicable online.

Choosing the optimal q on an interval may be viewed as a kind of MMAE with a very large model bank given by the discretization of the interval. The advantages of this approach over regular MMAE are that here accuracy increases with the number of models, and the problem of convergence to zero for model probabilities is eliminated.

B.4 Simulation Results

B.4.1 Evaluation of Performance

To evaluate the method we have performed simulations on both smooth (S) and non-smooth (NS) trajectories with different levels of noise. The smooth trajectories were generated by adding a random number of sines and cosines with varying amplitudes and periods and the non-smooth trajectories by adding a random number of piecewise constant functions. In both cases we let the parameter η describe the rate of change in the trajectories. In the smooth case, increasing η decreases the period while in the non-smooth case it decreases the length of each constant piece of the trajectory. Examples of 1D trajectories generated with this procedure are shown in Figure 3. A curve in 3D, for instance, is represented by three such sets, one for each state component.

The final simulated data is generated by adding a gaussian noise with standard deviation σ . To summarize, a simulated data set is defined by (S, η, σ) or (NS, η, σ) and generally sets with large values of η and σ are more challenging.

We let $\eta = \{0.05, 0.10, 0.15, \dots, 0.50\}$, $\sigma = \{0.05, 0.20, 0.35, \dots, 1.40\}$ and generate 100 data sets of the type (S, η, σ) and (NS, η, σ) respectively for each combination (η, σ) . To evaluate OAE we let it run on each of the data sets, using the system model (B.9) with $\delta = 1$. Since we assume no a priori knowledge of system dynamics, we let $\varepsilon = 0.5$. Note that for a fixed value of ε , Δ is the only tuning parameter of OAE. For all 20000 investigated trajectories we let $\Delta = 5$ to investigate sensitivity to tuning.

For comparison we run an MMAE algorithm on the same sets, using a model bank of 4 models of the type (B.9). The characteristics of each model in the bank are defined by its value of Q . We use $Q_1 = 0$, $Q_2 = 10^{-1}$, $Q_3 = 10^0$, $Q_4 = 10^1$. The choices of Q -values and number of models for the model bank will inevitably effect performance of MMAE. For some of the investigated trajectories performance might increase by adding or removing models or using other Q -values. However, the sensitivity to the choice of model bank is one of the drawbacks with MMAE and since we allow no alterations of OAE for different trajectories, we keep the model bank constant for MMAE. To get an in some sense fair comparison we impose an upper bound $q_{max} = 10$ on (B.11) so that both methods will employ Q -values on the interval $[0, 10]$. Figure 4 shows tracking with OAE and MMAE for a typical smooth (left) and non-smooth (right) trajectory. Performance is measured as the relative error

$$e(\eta, \sigma) \triangleq \left\| \frac{h(\eta, \sigma) - \hat{h}(\eta, \sigma)}{h(\eta, \sigma)} \right\| \quad (\text{B.31})$$

where $e = e_{OAE}$ for $\hat{h} = \hat{h}_{OAE}$ and $e = e_{MMAE}$ for $\hat{h} = \hat{h}_{MMAE}$. To get a robust analysis, 100 data sets are generated for each pair (η, σ) and the average $\bar{e}(\eta, \sigma)$ over the 100 runs is measured. The resulting

Figure 3: Examples of trajectories. **Left:** $\eta = 0.05$, right: $\eta = 0.5$. Top: smooth, bottom: non-smooth.

error difference $\bar{e}_{MMAE}(\eta, \sigma) - \bar{e}_{OAE}(\eta, \sigma)$ is plotted in Figure 5 for all pairs (η, σ) .

B.4.1.1 Smooth Curves

$(\bar{e}_{MMAE} - \bar{e}_{OAE}) > 0$ for all pairs (η, σ) except $(\sigma, \eta) = (0.5, 0.5)$, that is, for 99% of the tested pairs. See the left plot of Figure 5. Averaging over all runs and all pairs (η, σ) , the error of OAE is 83% of the error of MMAE for the smooth test cases.

B.4.1.2 Non Smooth Curves

$(\bar{e}_{MMAE} - \bar{e}_{OAE}) > 0$ for 87 out of the 100 tested pairs (η, σ) , see the right plot of Figure 5. MMAE was on average the better choice for $\sigma > 0.5$, $\eta \in [0.2, 0.3]$. Averaging over all runs and all pairs (η, σ) , the error of OAE is 85% of the error of MMAE for the non-smooth test cases.

Figure 4: Tracking with MMAE and OAE. **Left:** smooth trajectory. **Right:** non-smooth trajectory.

Figure 5: Difference in errors for MMAE and OAE as a function of η and σ . **Left:** smooth curves. **Right:** non-smooth curves.

B.4.2 Example: Tracking a Platform With Unicycle Dynamics

In this section we demonstrate OAE in a tracking application where the true target kinematics are both nonlinear and coupled. We assume the target has unicycle kinematics:

$$\begin{aligned} \dot{x}_1 &= u \cos \phi \\ \dot{x}_2 &= u \sin \phi \\ \dot{\phi} &= \omega, \end{aligned} \tag{B.32}$$

where (x_1, x_2) is the position of the target and ϕ is its heading. (u, ω) are control inputs. In terms of Equation (B.7), we have, with state vector $x = (x_1 \ x_2 \ x_3)^T = (x_1 \ x_2 \ \phi)^T$,

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} f_1(x, u, \omega) \\ f_2(x, u, \omega) \\ f_3(x, u, \omega) \end{pmatrix} = \begin{pmatrix} u \cos x_3 \\ u \sin x_3 \\ \omega \end{pmatrix} \quad (\text{B.33})$$

$$y = \begin{pmatrix} h_1(x) \\ h_2(x) \end{pmatrix} + \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \quad (\text{B.34})$$

The task for the tracking device is to track the target position using noise contaminated observations $y = h(x) + v$, obtained from range sensors. The tracking device has no information on the true target kinematics $f(x, u, \omega)$ or control (u, ω) . With no a priori information, a natural choice is to model the target kinematics as a random walk (RW) of the type (B.9). The model uncertainty w with covariance Q represents nonlinearity of the true model and coupling between states. For highly nonlinear and coupled true target kinematics such as (B.32), a good estimate of Q is essential.

With the notation introduced in Section B.3, we get

$N = 2$	Dimension of $h(x)$
$h = (h_1 \ h_2)^T = (x_1 \ x_2)^T$	Observation
$\mathbf{h}_1 = (h_1 \ \dot{h}_1)^T$	State in RW model of x_1
$\mathbf{h}_2 = (h_2 \ \dot{h}_2)^T$	State in RW model of x_2

The dynamics of $\mathbf{h}_1$ and $\mathbf{h}_2$ are modeled as (B.9) and estimates $\hat{\mathbf{h}}_1 = (\hat{h}_1 \ \dot{\hat{h}}_1)^T$ and $\hat{\mathbf{h}}_2 = (\hat{h}_2 \ \dot{\hat{h}}_2)^T$ are computed using OAE.

B.4.2.1 Tracking using OAE

Target tracking is performed for a target with the kinematics (B.32), traveling along a path generated by plugging in the control

$$u(\phi) = \min\{|\tan(4\phi)|, 1\} \quad (\text{B.35})$$

$$w(\phi) = -1.8 / \sin(\phi^2). \quad (\text{B.36})$$

into Equation (B.32).

A gaussian noise with standard deviation $[\sigma_x \ \sigma_y \ \sigma_\phi] = 0.05 [mean(x_1) \ mean(x_2) \ mean(\phi)]$ was added to the path to simulate measurement noise. The resulting OAE estimate is shown in Figure 6, together with the OAE values of q_1 and q_2 .

B.4.2.2 Estimation of non measurable states

According to the assumptions made, ϕ is not measurable. However, using a second order RW model, OAE outputs estimates of $(\dot{x}, \dot{y})$ as well. Noting that

$$\phi = \arctan(\dot{y}/\dot{x}) \quad (\text{B.37})$$

we can compute a rough estimate $\hat{\phi} = \arctan(\dot{\hat{y}}/\dot{\hat{x}})$. This estimate is plotted together with the true value in Figure 7.

B.5 Conclusions

A novel adaptive Kalman filtering method was presented in this paper. An estimate of the covariance matrix Q was found by solving an optimization problem over a short window of data. No a priori knowledge of system dynamics is needed and the method is scalable to N dimensions.

Figure 6: Tracking a unicycle target using OAE. **Left:** Target trajectory given by (B.35) - (B.36), noisy data and resulting OAE estimate. **Right:** The OAE values of q_1 and q_2 .

Figure 7: The estimate $\hat{\phi}$.

Trajectory tracking simulations show satisfying performance for a large class of systems and superiority over the reference method, a standard MMAE algorithm. Simulations indicate robustness to tuning in the sense that the value of Δ can be chosen the same for vastly different trajectories, such as the examples in Figure 3, and still perform better than the reference method.

Possible extensions of this work include introducing dynamic weights in the cost function $J(q)$ to emphasize punishment on bias or amplitude depending on knowledge of traits of the system

or magnitude of the measurement noise, and further investigation of the relation between Δ and performance for different types of systems. Alternative designs of $a(q)$ and $b(q)$ should also be evaluated.

B.6 Acknowledgments

The authors gratefully acknowledge the helpful comments of Danica Kragic at CAS, KTH.

B.7 References

- [1] P.S. Maybeck, Stochastic models, estimation, and control, vol. 1, *Academic Press*, 1979.
- [2] P.S. Maybeck, Stochastic models, estimation, and control, vol. 2, *Academic Press*, 1979.
- [3] P.S. Maybeck, Stochastic models, estimation, and control, vol. 3, *Academic Press*, 1982.
- [4] A.H. Mohamed, and K.P. Schwarz, Adaptive Kalman filtering for INS/GPS, *Journal of Geodesy*, vol. 73, pp. 193–203, 1999.
- [5] R. Mehra, On the identification of variances and adaptive Kalman filtering, *IEEE Trans. on Automatic Control*, vol. 15, pp. 175–184, 1970.
- [6] P.S. Maybeck and P.D Hanlon, Performance enhancement of a multiple model adaptive estimator, *IEEE Trans. on Aerospace and Electronic Systems*, vol. 31, pp. 1240–1254, 1995.
- [7] X.R. Li and Y. Bar-Shalom, Multiple-model estimation with variable structure, *IEEE Trans. on Automatic Control*, vol. 41, no. 4, pp. 478–493, 1996.
- [8] X.R. Li, Multiple-model estimation with variable structure: some theoretical considerations, *Proc. of the 33rd IEEE Conference on Decision and Control (CDC)*, vol. 2, pp. 1199–1204, 1994.
- [9] X.R. Li, Multiple-model estimation with variable structure, ii. model-set adaptation, *IEEE Trans. on Automatic Control*, vol. 45, no. 11, pp. 2047–2060, 2000.
- [10] X.R. Li, X. Zwi, and Y. Zwang, Multiple-model estimation with variable structure. iii. model-group switching algorithm, *IEEE Trans. on Aerospace and Electronic Systems*, vol. 35, no. 1, pp. 225–241, 1999.
- [11] X.R. Li, Y. Zhang, and X. Zhi, Multiple-model estimation with variable structure. iv. design and evaluation of modelgroup switching algorithm, *IEEE Trans. on Aerospace and Electronic Systems*, vol. 35, no. 1, pp. 242–254, 1999.
- [12] X.R. Li and Y. Zhang, Multiple-model estimation with variable structure. v. likely-model set algorithm, *IEEE Trans. on Aerospace and Electronic Systems*, vol. 36, no. 2, pp. 448–466, 2000.
- [13] X.R. Li, V. Jilkov, and J. Ru, Multiple-model estimation with variable structure- part vi: expected-mode augmentation, *IEEE Trans. on Aerospace and Electronic Systems*, vol. 41, no. 3, pp. 853–867, 2005.
- [14] P.S. Maybeck and K.P. Hentz, Investigation of moving-bank multiple model adaptive algorithms, *Proc. of the 24th IEEE Conference on Decision and Control (CDC)*, vol. 24, pp. 1874–1881, 1985.

- [15] J.W. Ding and C. Rizos, Improving adaptive kalman estimation in gps/ins integration, *Journal Of Navigation*, vol. 60, pp. 517–529, 2007.
- [16] Y. Yang and W. Gao, An optimal adaptive kalman filter, *Journal of Geodesy*, vol. 80, no. 4, pp. 177–183, 2006.
- [17] X. Wang and V. Syrmos, Vehicle health monitoring system using multiple-model adaptive estimation, *Proc. of the 11th IEEE Mediterranean Conference on Control and Automation*, 2003.
- [18] L. Jetto, S. Longhi, and G. Venturini, Development and experimental validation of an adaptive extended kalman filter for the localization of mobile robots, *IEEE Trans. on Robotics and Automation*, vol. 15, no. 2, pp. 219–229, 1999.
- [19] G. Reina, A. Vargas, K. Nagatani, and K. Yoshida, Adaptive kalman filtering for gps-based mobile robot localization, *Proc. of IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR)*, pp. 1–6, 2007.
- [20] M. Ficocelli and F. Janabi-Sharifi, Adaptive filtering for pose estimation in visual servoing, *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 19–24, 2001.
- [21] V. Lippiello and L. Villani, Adaptive extended Kalman filtering for visual motion estimation of 3d objects, *Control Engineering Practice*, vol. 15, no. 1, pp. 123–134, 2007.
- [22] P. Maybeck and P. Hanlon, Performance enhancement of a multiple model adaptive estimator, *IEEE Trans. on Aerospace and Electronic Systems*, vol. 31, no. 4, pp. 1240–1254, 1995.