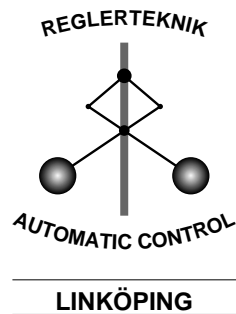


# THREE STATISTICAL BATCH ALGORITHMS FOR TRACKING MANOEUVRING TARGETS

Niclas Bergman and Fredrik Gustafsson

Department of Electrical Engineering  
Linköping University, SE-581 83 Linköping, Sweden  
WWW: <http://www.control.isy.liu.se>  
Email: {niclas,fredrik}@isy.liu.se

September 24, 1999



Report no.: LiTH-isy-R-2189

Accepted for publ. in European Control Conference 1999 (ECC'99)

Technical reports from the Automatic Control group in Linköping are available by anonymous ftp at the address <ftp.control.isy.liu.se>. This report is contained in the compressed postscript file 2189.ps.Z.

# THREE STATISTICAL BATCH ALGORITHMS FOR TRACKING MANOEUVRING TARGETS

N. Bergman and F. Gustafsson

*Division of Automatic Control  
Dept. of Electrical Engineering  
Linköping University  
SE-58183 Linköping, Sweden  
{niclas,fredrik}@isy.liu.se*

**Keywords** : target tracking, MCMC, EM algorithm, Kalman filtering, change detection

## Abstract

The EM algorithm and two MCMC algorithms are applied to manoeuvre detection in target tracking. These statistical methods are off-line and the intended use is to compute upper performance limits of on-line algorithms as well as for off-line analysis. A consequence of the MCMC theory is that an approximation of the a posteriori distribution for the manoeuvre times is obtained.

## 1 Introduction

In Kalman filter design, there is an inherent trade-off between tracking ability and noise attenuation. In target tracking, this means that we have the option of choosing a fast or slow filter. The extreme designs include a very fast filter, whose position filter estimate essentially follows the measurements in Figure 1.a, or a very slow filter as in Figure 1.b, whose noise attenuation is very good during straight courses, but with a clear overshoot at the manoeuvres.

Change detection is the problem of determining when abrupt changes in a signal characteristic occur, using noisy measurements of the signal. In target tracking, the standard use is to speed up the filter after a detected manoeuvre. This means that we can use the very slow filter in Figure 1.b, and increase the adaptation gain with a factor of, say, 10 after each change detection. This covariance increase is the approach taken in this contribution, but alternatives exist.

We will here compare the performance of some well-known on-line change detectors with some batch algorithms. These kind of off-line algorithms are not previously considered in the literature of target tracking, but we believe that there are applications for them in off-line data analysis and for computing upper performance bounds for on-line algorithms.

Proposed on-line (non-linear) filters, with a change detection component, for target tracking include the variable dimension filter, the Interacting Multiple Model (IMM) algorithm, the General Pseudo Bayesian (GPB) filter, the IMM algorithm. A general survey of these is found in [1].

Due to the real-time application of target tracking, algorithms for manoeuvre detection are usually designed for recursive operation. However, several algorithms that have been applied to change detection using off-line techniques can be applied to manoeuvre detection as well. In off-line processing, change detection is sometimes referred to as *segmentation* since abrupt changes define segments of data, which for modeling purposes can be considered as quasi-stationary. In this work we present three algorithms for segmentation of batch data applied to target tracking. The problem statement and the notational conventions are described in Section 2. Two of the algorithms presented here are based on the concept of Markov Chain Monte Carlo (MCMC) methods. They are presented in Section 4. The third algorithm is presented in Section 3 and applies the Expectation Maximization (EM) algorithm to the segmentation problem. The off-line characters of the algorithms make them suitable for analysis of tracking data. A simulation study comparing the performance of the algorithms is presented in Section 5. All three algorithms can be made “pseudo recursive” by buffering data and applying them to batches of collected measurements as they become available. Approximate on-line versions of the EM-algorithm can be constructed by replacing the smoothed estimates with filtered estimates as described in [6]. Some MCMC algorithms have been applied to recursive estimation [4]. This is an active field of research at the time of writing.

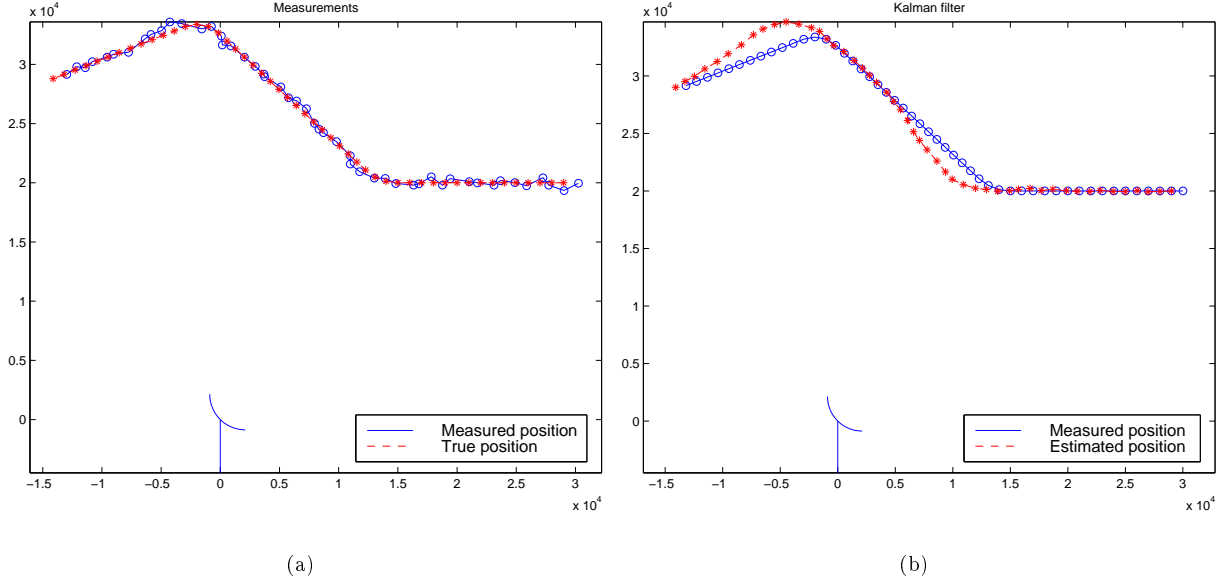


Figure 1: A simulated flight path with coordinated turns with measurements from a radar. Position measurements (a) and filtered position estimates from a slow Kalman filter (b).

In all simulations, we will restrict ourselves to use a 2D acceleration model with four states and all Kalman filters will have the same design parameters. Modeling is an interesting area, both for research and implementation. Alternatives include acceleration or jerk linear model and non-linear models for coordinated turns and the choice of state vector in these. Another practically important problem is that of tuning: optimize all design parameters to a set of real data sets or a simulation setup.

## 2 Problem Description

Consider a linear Markovian state space model for the sought  $n_x$ -dimensional target state vector  $x_k$  where the dynamics depend on the unknown binary *segmentation sequence*  $\delta_k \in \{0, 1\}$ ,

$$\begin{aligned} x_{k+1} &= F_{k,\delta} x_k + G_{k,\delta} w_{k,\delta} \\ y_k &= H_{k,\delta} x_k + e_{k,\delta} \end{aligned} \quad k = 1, 2, \dots, N \quad (1)$$

Here,  $\{w_{k,\delta}\}$  and  $\{e_{k,\delta}\}$  are two independent i.i.d. sequences of zero mean Gaussian random variables with known full rank covariances  $Q_{k,\delta}$  and  $R_{k,\delta}$ , respectively. The notation above indicates that the system parameters have a known time dependency as well as that they depend on the current segmentation parameter  $\delta_k$ . The explicit time dependent system matrices in (1) arises for instance from local linearizations along the estimated target trajectory in an Extended Kalman Filter (EKF) setting.

In the manoeuvring target tracking application we consider in this work, the segmentation parameter  $\delta_k$  typically affects only the process noise covariance, e.g., increasing it with a scalar fudge factor  $\gamma \gg 1$  at the change instant

$$Q_{k,\delta} = (1 - \delta_k)Q_k + \gamma\delta_k Q_k. \quad (2)$$

The algorithms described in this work apply to general  $r$ -valued segmentations, i.e., for  $\delta_k \in \{0, 1, \dots, r\}$ , and hence the model (1) can be seen as a mode jumping or switching model. We will however use (2) in the simulation study.

The initial state  $x_1$  is Gaussian with known mean  $\hat{x}_1$  and covariance  $P_1$ , the noise sequences  $\{w_{k,\delta}\}$ ,  $\{e_{k,\delta}\}$  and the initial state  $x_1$  are mutually uncorrelated. A length  $N$  batch of measurement data  $Y^N = \{y_k\}_{k=1}^N$  is available for off-line processing. The objective of the segmentation algorithm is to find the sequence  $\delta^N = \{\delta_k\}_{k=1}^N$  that maximizes the posterior probability  $p(\delta^N | Y^N)$ . The prior distribution of  $\delta^N$  is also assumed to be known, we will use different models in the different algorithms.

### 3 Expectation Maximization

The Expectation Maximization (EM) algorithm [3, 2] is a multiple pass batch algorithm where a local maximum of the posterior distribution of the parameters  $\Theta$  given the incomplete measurement set  $Z$  is obtained. The data set is divided into two disjoint parts  $Z = Z_1 \cup Z_2$  where  $Z_1$  is the observed, or measured, data and  $Z_2$  is an unobserved data set. The algorithm is defined by the recursion

$$\Theta_{p+1} = \arg \max_{\Theta} \mathbb{E}_{Z_2} (\ln p(\Theta | Z) | Z_1, \Theta_p)$$

where  $p$  denotes the pass. Since  $p(\Theta | Z) = p(\Theta, Z)/p(Z)$  the maximization can be performed over the joint density  $p(\Theta, Z)$  which has the same maximizing argument.

The Expectation Maximization Segmentation (EMS) algorithm is obtained by treating the segmentation sequence  $\delta^N$  as the parameters and the states  $X^N$  as the unobserved data. The EM pass therefore is defined by

$$\delta_{p+1}^N = \arg \max_{\delta^N} \mathbb{E}_{X^N} (\ln p(Y^N, X^N, \delta^N) | Y^N, \delta_p^N). \quad (3)$$

Given the segmentation sequence  $\delta_p^N$ , the conditional expectation of the sequence of states  $X_N$  given the batch of data  $Y^N$  is found from fixed interval Kalman smoothing. The maximization is performed over the discrete variable  $\delta^N$ .

For reasons that will be clear later, we model the prior distribution of the segmentation parameters  $\delta_k$  as independent Bernoulli variables,

$$\delta_k = \begin{cases} 0 & \text{with probability } q \\ 1 & \text{with probability } 1 - q \end{cases} \quad k = 1, 2, \dots, N. \quad (4)$$

Straightforward, but tedious, calculations show that the EMS algorithm (3) is given by the iterations

$$\delta_{p+1}^N = \arg \max_{\delta^N} Q(\delta^N, \delta_p^N)$$

where the return function is

$$\begin{aligned} Q(\delta^N, \delta_p^N) = & \sum_{k=1}^N \left( 2\delta_k \ln \left( \frac{1-q}{q} \right) - \|y_k - H_{k,\delta} \hat{x}_k^p\|_{R_{k,\delta}^{-1}}^2 \right. \\ & \left. - \ln |R_{k,\delta}| - \text{tr} \left( H_{k,\delta}^T R_{k,\delta}^{-1} H_{k,\delta} P_{k,k}^p \right) \right) \\ & + \sum_{k=1}^{N-1} \left( - \|G_{k,\delta}^\dagger (\hat{x}_{k+1}^p - F_{k,\delta} \hat{x}_k^p)\|_{Q_{k,\delta}^{-1}}^2 - \ln |Q_{k,\delta}| - \right. \\ & \left. - \text{tr} \left( G_{k,\delta}^{\dagger T} Q_{k,\delta}^{-1} G_{k,\delta}^\dagger (P_{k+1,k+1}^p - 2F_{k,\delta} P_{k,k+1}^p \right. \right. \\ & \left. \left. + F_{k,\delta} P_{k,k}^p F_{k,\delta}^T) \right) \right). \quad (5) \end{aligned}$$

Above,  $\text{tr}(A)$  is the trace of the matrix  $A$ ,  $|A|$  is the determinant,  $\|x\|_A^2$  is the quadratic norm  $x^T A x$ ,  $A^\dagger = (A^T A)^{-1} A^T$  is the Moore-Penrose pseudo inverse, and

$$\begin{aligned} \hat{x}_k^p &= \mathbb{E} (x_k | Y^N, \delta_p^N) \\ P_{l,k}^p &= \mathbb{E} ((x_l - \hat{x}_l^p)(x_k - \hat{x}_k^p)^T | Y^N, \delta_p^N) \end{aligned}$$

which are computed from fixed interval Kalman smoothing based on the segmentation sequence  $\delta_p^N$  and the measurement set  $Y^N$ .

Note that due to the prior (4), there is no relation between  $\delta_k$  for different  $k$  in (5). Each term in (5) depends on only one  $\delta_k$ . Hence, the maximization step in the EMS algorithm is performed for each  $\delta_k$  independently. This means that given the estimates  $\hat{x}_k^p$  and covariances  $P_{l,k}^p$  the maximizing argument of (5) is found by comparing the return value with and without a jump at each time instant.

**Algorithm 1 (EM segmentation).** The EMS algorithm is summarized in the following steps:

1. (Initialize,  $p = 1$ )  
Assume no jumps by setting  $\delta_1^N = 0$ .

2. (Expectation)  
Compute the estimates  $\hat{x}_k^p$  and the error covariances  $P_{l,k}^p$  using fixed interval Kalman smoothing based on the segmentation sequence  $\delta_p^N$ .
3. (Maximization)  
Compute the next segmentation sequence  $\delta_{p+1}^N$  as the maximizing argument of (5) by choosing the alternative with highest return for each  $k$ .
4. Set  $p := p + 1$  and return to item 2.

The iterations are halted when no significant improvement of (5) is obtained.

In the EM algorithm, the return is monotonically increasing with the number of passes  $p$  through the algorithm [3, 2]. This ensures that a local maxima of the conditional density is reached.

## 4 Monte Carlo Algorithms

An algorithm proposed in [5] for segmentation can be generalized to the detection problem described in Section 2. In our framework, it is here formulated for the case of binary independent  $\delta_k$ , which can be represented by  $n$  change times  $k^n = \{k_i\}_{i=1}^n$  we assume that  $n$  is known or estimated before applying the algorithm.

**Algorithm 2 (Gibbs–Metropolis MCMC detection).** Initiate the algorithm with  $n$  change points  $k^n$  uniformly distributed over the sequence. Iterate for each Monte Carlo run the following three steps: ( $i = 1$ )

1. Iterate Gibbs sampler for component  $i$  in  $k^n$ , generating a random number from

$$\bar{k}_i \sim p(k_i | k_1, k_2, \dots, k_{i-1}, k_{i+1}, \dots, k_n) = p(k_i)$$

due to independent change times. Denote the new candidate sequence  $\bar{k}^n$ . The distribution above may be taken as uniform over the interval,  $U(1, N)$ , or Gaussian centered around  $k_i$ ,  $N(k_i, \sigma)$ .

2. Run the Kalman filter using the sequence  $\bar{k}^n$ , and save the innovations  $\varepsilon_k$  and its covariances  $S_k$ .
3. The candidate  $\bar{k}^n$  is accepted with probability

$$\min \left( 1, \frac{p(\varepsilon^N(\bar{k}^n))}{p(\varepsilon^N(k^n))} \right)$$

(the Metropolis step). Increase  $i$  and repeat from item one above unless  $i > n$ .

After the *burn-in* (convergence) time, the distribution of change times can be computed by standard Monte Carlo techniques.

The second Monte Carlo algorithm is similar to the EMS algorithm described in Section 3. The maximization of the return function (5) is replaced with a Monte Carlo randomization step. Using the manoeuvre model (2) and the Bernoulli prior (4) the conditional probability of the segmentation sequence is

$$\Pr(\delta^N | x^N) \propto \prod_{k=1}^{N-1} \frac{1}{\sqrt{|Q_{k,\delta}|}} \exp \left( -\frac{1}{2} \|G_{k,\delta}^\dagger (x_{k+1} - F_{k,\delta} x_k)\|_{Q_{k,\delta}^{-1}}^2 \right). \quad (6)$$

**Algorithm 3.** Initialize assuming no jumps  $\delta_1^N = 0$  and set  $p = 1$ .

1. (Expectation)  
Compute the estimates  $\hat{x}_k^p$  and the error covariances  $P_{l,k}^p$  using fixed interval Kalman smoothing based on the segmentation sequence  $\delta_p^N$ .
2. (Randomization)  
Generate the next segmentation sequence  $\delta_{p+1}^N$  as a random sample from the conditional probability (6) inserting the smoothed sequence  $\{\hat{x}_k^p\}_{k=1}^N$ .
3. Set  $p := p + 1$  and iterate back to item one.

The interpretation of the randomization step is the following: If the smoothed sequence  $\hat{x}_k^p$  is changing rapidly at some instance (large derivative), then a change is associated with that time instant with high probability. This algorithm is not evaluated in the next section.

## 5 Simulations

In this simulation example, we use a five state nonlinear model of an aircraft making four abrupt changes in turn rate, as depicted in Figure 1.a. The aircraft starts to the right in the figure and flies towards the left. The position measurements are generated using a sampling period of 5 seconds and independent additive Gaussian noise with standard deviation  $300 \text{ m}^2$  in each channel. The target is turning during sample 15–18 and sample 35–38.

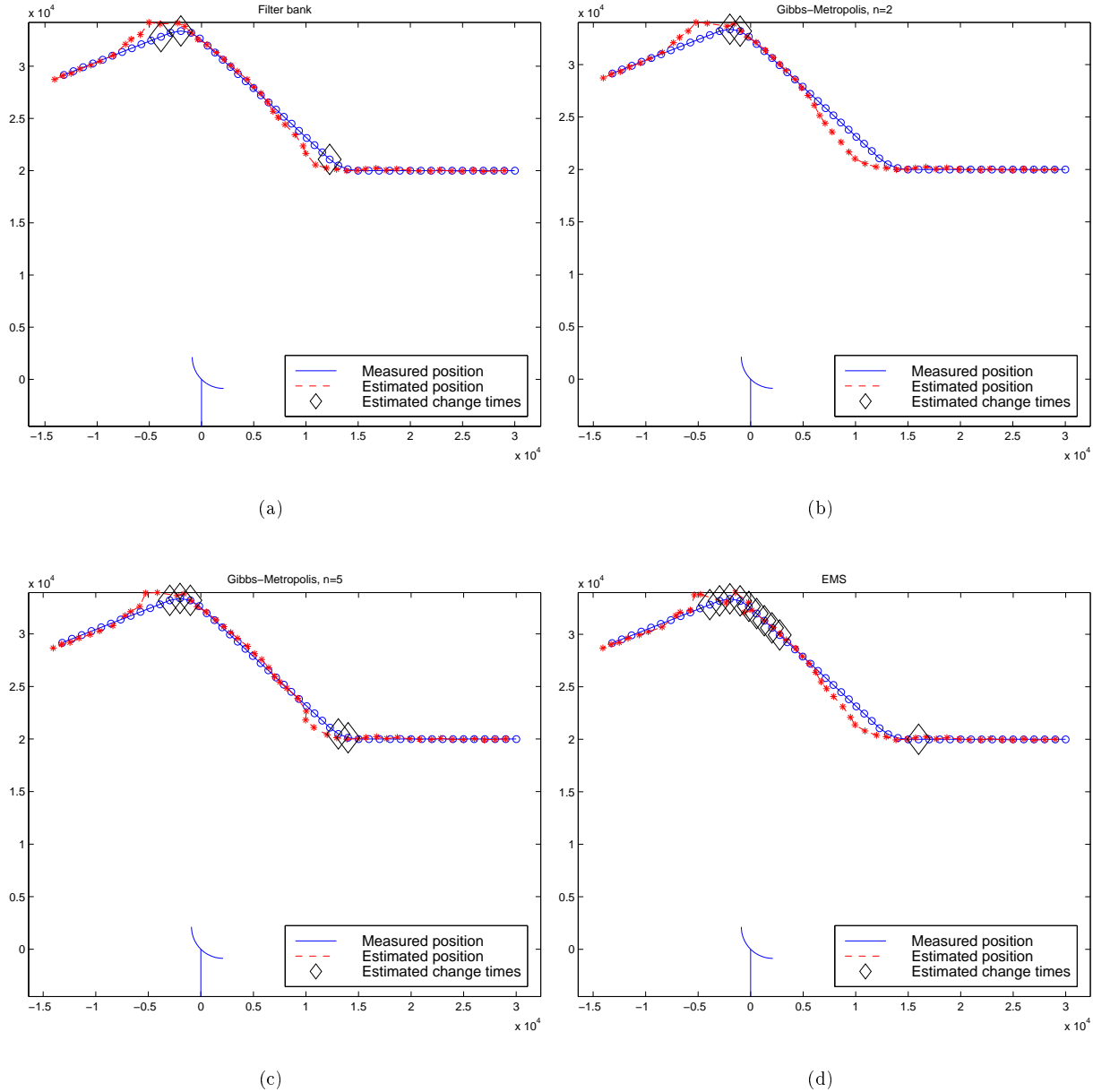


Figure 2: Tracking performance for filter bank (a), Gibbs-Metropolis with  $n = 2$  (b), Gibbs-Metropolis with  $n = 5$  (c) and EMS (d).

The simplest possible change detection strategy applies a innovation whiteness test to the Kalman filter. Here, the CUSUM test of the normalized squared innovations is applied. The histogram in Figure 3.s shows the result of a Monte Carlo simulation. In the filter bank strategy of manoeuvre detection several Kalman filters are run in parallel under different manoeuvre hypotheses. The ever increasing tree of hypotheses is pruned on basis of the likelihood of each filter. Figure 5.a shows the results of this recursive algorithm.

Figure 5.d presents the result of applying the EMS algorithm to the batch of data depicted in Figure 1.a. In the filter, a four dimensional linear model of the aircraft movement was used together with the manoeuvring model (2).

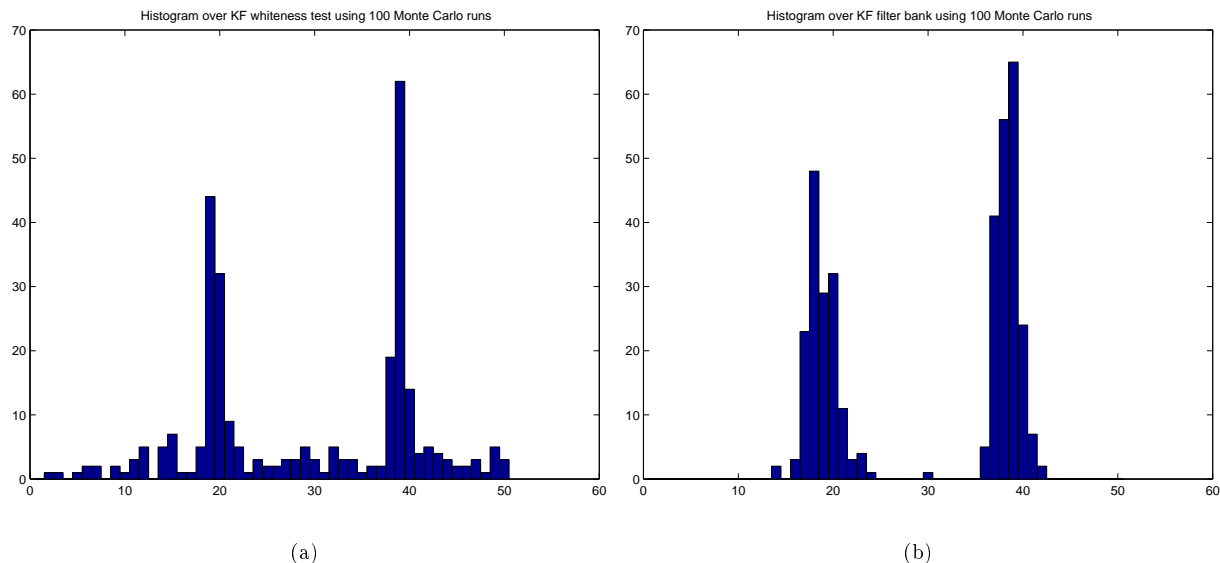


Figure 3: Histograms over change times from Kalman filter whitening (a) test and filter bank (b).

Since the simulation model is nonlinear the distinct jumps in the fifth turn rate state are hard to detect using the linear model. There is a trade-off between detection sensitivity and non-manoeuvre performance since the first turn has less turn rate than the second. The trade-off is controlled by choosing the three filter parameters process noise covariance  $Q_k$ , the fudge factor  $\gamma$ , and the probability of not manoeuvring  $q$ . After some tuning of the filter parameters the filter detected manoeuvres at sample 15 and during samples 32–40.

From the tuning of EMS, we noted the following phenomena:

- The convergence is very fast, typically in the order of 2 or 3 iterations of the EM algorithm.
- On the other hand, if the signal to noise ratio is decreased, by increasing the measurement noise variance, it was hard to get any reasonable result at all.

Figure 5.c and d show the predictions when using the Gibbs Metropolis scheme in Algorithm 2. A fudge factor of  $\gamma = 10$  is used in the manoeuvre model (2). The jump sequence as a function of iterations and sub iterations of the Gibbs sampler are shown in Figure 4. The iteration scheme converges to two change points at 20 and 36. In case too many change points are introduced they are placed either at the first or at the last time instant in the batch. A problem here is that the iterations get stuck and the obtained histograms do not resemble a posteriori distributions as we had expected.

It might be interesting to compare the results in terms of root mean square position error for the *prediction* (thus, the velocity estimate is used implicitly):

Method	Prediction RMSE	Estimated change times
Measurement	416	
Kalman filter	385	
KF with whitening test	335	20, 39
Filter bank	309	19, 38, 40
Gibbs (2)	379	37, 38
Gibbs (5)	299	17, 18, 37, 38, 39
EMS	418	15, 32, 33, 34, 35, 36, 37, 38, 39, 40

Table 1: Comparison of root mean square error (RMSE) of position predictions.

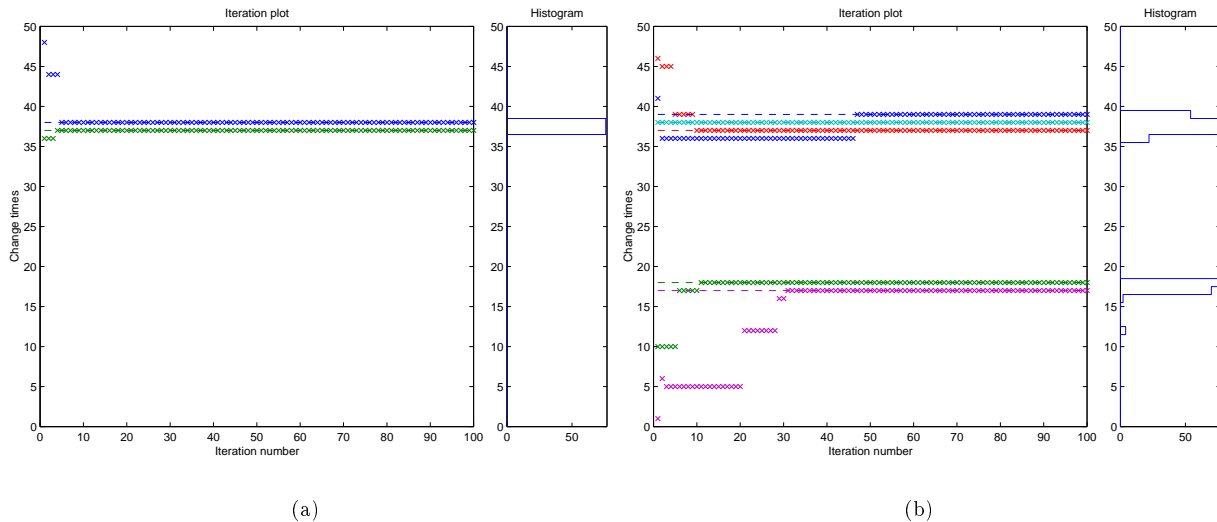


Figure 4: Iteration plot with projected histograms over change times for Gibbs Metropolis with  $n = 2$  (a) and  $n = 5$  (b).

## 6 Conclusions

Three algorithms for segmentation of data has been applied to detection of manoeuvres for target tracking. All three algorithms require batch data but can be adjusted for recursive operation in a number of ways.

## References

- [1] Y. Bar-Shalom and X.R. Li. *Estimation and Tracking: Principles, Techniques and Software*. Artech House, 1993.
- [2] L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occuring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Stat.*, 41:161–171, 1970.
- [3] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc.*, 39(1):1–38, 1977.
- [4] A. Doucet. On sequential simulation-based methods for Bayesian filtering. Technical Report CUED/F-INFENG/TR.310, Signal Processing Group, Department of Engineering, University of Cambridge, 1998.
- [5] W.J. Fitzgerald, J.J.K. Ruanaidh, and J.A. Yates. Generalised changepoint detection. Technical Report CUED/F-INFENG/TR.187, Signal Processing Group, Department of Engineering, University of Cambridge, England, 1994.
- [6] V. Krishnamurthy and J.B. Moore. On-line estimation of hidden Markov model parameters based on the Kullback-Leibler information measure. *IEEE Trans. on Signal Processing*, 41(8):2557–2573, 1993.