A TWO-FILTER OFF-LINE SOLUTION TO OPTIMAL DETECTION

Fredrik Gustafsson
Dept. of Electrical Engineering
Linköping University, Sweden

November 11, 1991

Abstract

The problem of detecting abrupt changes of the dynamics in linear systems is addressed. First the *generalized likelihood ratio* test for detection in a state space model is examined. This test can be interpreted as a simultaneous *maximum likelihood* (ML) estimator of the jump time and jump magnitude. The problem is that this test requires a large number of matched filters which increases linearly with time. This is shown by simulations to be very time consuming. Therefore, a two-filter implementation of the ML estimator of the jump time is derived, which is very efficient in off-line detection. It consists of two Kalman filters, one running forwards and one backwards in time. The algorithm gives a considerable improvement on-line as well, if a batchwise data processing is allowed. A major problem in detection is the sensivity to incorrect values of the noise variance. This is solved by treating the noise variance as a stochastic variable, possibly changing at the jump time. A modified algorithm is given, which is still optimal for this case.

1 INTRODUCTION

1.1 Preliminaries

In this paper we address the classical detection problem introduced in [10] and [9]. They considered a general linear state space model, in which the state vector undergoes sudden additive changes. These changes can for instance be caused by system failures.

A general approach for detection is the *generalized likelihood ratio* (GLR) test, see for instance [2], [4] and [5] for some applications and [3] for a comparative study. The key point in the GLR test is that the influence of the state jump can be transformed into a linear regression framework for the innovations from a Kalman filter acting as if no jump has occured. From the regression model, one matched filter for each possible jump instant can be constructed. The test statistic in the

GLR test can then be derived from the output from these matched filters. The main problem is that the number of matched filters increases in time. Although each filter is supposed to be computationally efficient, compared to a Kalman filter, this makes this optimal test less attractive in practice. Therefore an approximation is suggested in [10], where only recent possible changes are considered. Other limitations are the lack of robustness to incorrectly modeled measurement noise and the difficulty in choosing a certain threshold in the test.

The aforementioned problems are solved here. The approach is to compute the *maximum likelihood* (ML) estimate of the unknown jump instant. The ML estimate will be shown to be closely related to the estimate from the GLR test. First a direct solution to the detection problem is derived which requires N Kalman filters, where N is the number of measurements. This algorithm is used as a Benchmark for the alternative approaches. Then a new algorithm to compute the ML estimate is derived, which uses only two Kalman filters, one running forwards and one backwards in time. This method is very fast, even when compared to suggested approximations of the GLR test. The only drawback is that it basically is an off-line approach. A related two-filter idea is found in [7], where a sub-optimal two filter detection algorithm is proposed to detect changes in the parameters for a finite impulse response model.

A criticism often given to filter-based detection methods is the sensitivity to incorrectly given values of the noise variance Λ, see *e.g.* [2] and [7]. Almost every algorithm can be fooled to estimate a jump if the noise variance is suddenly increased, or clearly underestimated. A new approach is suggested here, where the noise variance is assumed to be a stochastic variable. The likelihoods are then found by integration over the prior of Λ. Two possibilities will be investigated, the noise variance can either be constant or change to a new value after the jump.

1.2 Signal Model

The signal model used herein is a linear state space model with additive state jump,

$$\begin{aligned} x_{t+1} &= F_t x_t + w_t + \delta(k-t)\nu \\ y_t &= H_t^T x_t + e_t. \end{aligned} \tag{1}$$

Here w_t and e_t and x_0 are assumed to be stochastic variables with a Gaussian prior,

$$\begin{aligned} w_t &\in \mathrm{N}(0, Q) \\ e_t &\in \mathrm{N}(0, \Lambda) \\ x_0 &\in \mathrm{N}(0, \Pi_0) \end{aligned}$$

All variables are asssumed to be independent. The state jump occurs at the unknown time instant k, and $\delta(j)$ is the pulse function which is one if $j = 0$ and zero otherwise.

The jump magnitude ν is most often supposed to be an unknown constant. However, in our Bayesian approach it must be considered as stochastic and we have to assert a prior to it. In order to get comparable results, we will use the non-informative prior with infinite variance. Although this yields an improper probability density function, the *a posteriori* distribution will be proper. Thus, the unknown constant will sometimes be treated as a stochastic variable with infinite variance. We will also investigate the case of a finite variance by assuming a Gaussian prior of ν, that is $\nu \in \mathrm{N}(\nu_0, P_\nu)$.

For compact notations, the set of measurements from time k to time l will be denoted by $Y_k^l = \{y(t); t = k, k+1, .., l\}$.

1.3 Maximum Likelihood Estimators

From the signal model (1), the detection problem can now be formulated as estimating the unknown jump instant k, given the measurements Y_1^N. We adopt the convention that $k = N$ means no jump. A natural approach is to derive the maximum likelihood (ML) estimator of the jump instant. The following two likelihoods will be considered, where k denote the jump instant,

$$p(Y_1^N|k, \nu) \quad \text{and} \quad p(Y_1^N|k).$$

The difference of the two likelihoods is that the former gives the simultaneous ML estimate of the jump instant *and* the jump magnitude,

$$\widehat{(k, \nu)}^{ML} = \arg\max_{k,\nu} p(Y_1^N|k, \nu), \tag{2}$$

while the latter gives the ML estimate of the jump instant alone,

$$\hat{k}^{ML} = \arg\max_{k} p(Y_1^N|k). \tag{3}$$

The estimate from (2) is computed by first maximizing the likelihood with respect to ν to get $\hat{\nu}(k)$ and then compute the simultaneous estimate of the jump instant, $\hat{k}_s^{ML}$, from

$$\hat{k}_s^{ML} = \arg\max_{k} p(Y_1^N|k, \hat{\nu}(k)). \tag{4}$$

It is important to note the difference, although it will be shown that these two likelihoods are closely related.

1.4 The Motivating Ideas

Our task is to compute the different likelihoods as efficient as possible and we will now indicate how this will be done. In the GLR test described in [10], the innovations, ϵ_t, from a Kalman filter assuming no jump are used. The basic point is that these innovations can be expressed as a linear regression in the jump magnitude,

$$\epsilon_t = \varphi_t^T(k)\nu + \epsilon_t(k).$$

Here $\varphi_t(k)$ is the regression vector for a jump at time k and $\epsilon_t(k)$ is a new residual. Now a matched filter can be used to determine whether these new residuals are significant less than the original.

The new two-filter implementation is a completely different approach. When a jump occurs in (1) we see that the state vector before the jump contains no information about its value after the jump, because of the unknown jump magnitude. It will be shown that the likelihood can be separated into two parts under the hypotheses of jump at time k,

$$p(Y_1^N|k) = p(Y_1^k)p(Y_{k+1}^N). \tag{5}$$

Since all possible jump hypotheses must be considered we have to compute equation (5) for all k. The idea is to recursively compute the first factor $p(Y_1^k)$ by one Kalman filter running forwards in time, and the second factor $p(Y_{k+1}^N)$ by one Kalman filter running backwards in time. The important point is that the same two filters can be used for all jump instants k.

2 A DIRECT IMPLEMENTATION

Suppose we let the jump magnitude be Gaussian distributed with zero mean and covariance matrix P_ν. It is then clear that (1) is just a plain linear model when conditioned on a particular jump instant. This means that the Kalman filter applies directly. It is then tempting to run one Kalman filter for each jump instant $k = 1, 2, .., N$ and see which one performs best, in a sense which will be made clear soon. This leads to a direct solution to the detection problem, as formulated in the introduction. The likelihoods for the N possible jump instants are in this implementation computed by using one Kalman filter for each likelihood. Because the magnitudes of the likelihoods turn out to be of completely different orders, the log likelihood will be considered to avoid possible numerical problems. The following algorithm is not believed to be very efficient, but because of its simplicity and the straightforward derivation it will be used as a Benchmark for alternative detection methods derived later.

Algorithm 2.1 (Direct on-line detection) *Consider the signal model (1), where the covariance matrix of the Gaussian distributed jump magnitude is P_ν. For each $k = 1, 2, ..., t$, update the k'th Kalman filter,*

$$\begin{aligned}
\hat{x}_{1|0}(k) &= x_0 \\
P_{1|0}(k) &= P_0 \\
\epsilon_t(k) &= y_t - H_t^T \hat{x}_{t|t-1}(k) \\
V_t(k) &= H_t P_{t|t-1}(k) H_t^T + \Lambda_t \\
\hat{x}_{t+1|t}(k) &= F_t \hat{x}_{t|t-1}(k) + F_t P_{t|t-1}(k) H_t V_t^{-1}(k) \epsilon_t(k) \\
P_{t+1|t}(k) &= F_t \left(P_{t|t-1}(k) - P_{t|t-1}(k) H_t^T V_t^{-1}(k) H_t P_{t|t-1}(k) \right) F_t + Q_t + \delta(t-k) P_\nu .
\end{aligned}$$

The log likelihood, conditioned on a jump at time k, is then recursively computed by

$$\log p(Y_1^t) = \log p(Y_1^{t-1}) - \frac{1}{2}\log 2\pi - \frac{1}{2}\log\det V_t(k) - \frac{1}{2}(\epsilon_t(k))^T V_t^{-1}(k)\epsilon_t(k). \quad (6)$$

Proof: By Bayes' rule we have

$$p(Y_1^t) = p(y_t|Y_1^{t-1})p(Y_1^{t-1}).$$

It is a well known property of the Kalman filter that

$$(y_t|k) \in \mathrm{N}(H_t\hat{x}_{t|t-1}(k), H_tP_{t|t-1}(k)H_t^T + \Lambda_t).$$

Thus, we get

$$p(y_t|k) = \gamma(y_t - H_t\hat{x}_{t|t-1}(k), H_tP_{t|t-1}(k)H_t^T + \Lambda_t).$$

which gives the desired result by taking the logarithm. Here $\gamma(x-\mu, P)$ denotes the Gaussian probability density function, with mean μ and covariance P, evaluated in x. □

This approach requires a linearly growing number with N of Kalman filters. In the next sections we will derive two completely different methods for computing the same thing.

3 THE GLR TEST

3.1 Introduction

The Generalized Likelihood Ratio (GLR) test is a multiple hypothesis test, where the jump hypotheses are compared to the no jump hypothesis pairwise. The hypotheses under consideration are

$$\begin{aligned} H_0 &: \quad \text{no jump} \\ H_1(k) &: \quad \text{a jump of magnitude } \nu \text{ at time } k. \end{aligned}$$

The test is as follows. Introduce the log ratio of the likelihoods for the hypotheses, maximized with respect to the jump magnitude ν, as the test statistic:

$$l_N(k) = \max_\nu 2\log\frac{p(Y_1^N|H_1(k))}{p(Y_1^N|H_0)}. \quad (7)$$

In our notation, this can be written

$$\begin{aligned} l_N(k) &= \max_\nu 2\log\frac{p(Y_1^N|k,\nu)}{p(Y_1^N|k=N)} \quad (8) \\ &= 2\log\frac{p(Y_1^N|k,\hat{\nu}(k))}{p(Y_1^N|k=N)}, \end{aligned}$$

where $\hat{\nu}(k)$ is the maximum likelihood estimate of ν, given a jump at time k.

A jump candidate is given by

$$\hat{k} = \arg\max_k l_N(k) = \arg\max_k 2\log\frac{p(Y_1^N|k,\hat{\nu}(k))}{p(Y_1^N|k=N)}$$

and it is accepted if

$$l_N(\hat{k}) > \alpha.$$

We will see that this test statistic for a given k is χ^2 distributed with $n = \dim\nu$ degrees of freedom. Thus, given the confidence level on the test, the threshold α can be found from standard statistical tables. If the zero-jump hypothesis is rejected, the state estimate can easily be compensated for the detected jump. Note that this is a multiple hypotheses test performed for each $k = 1, 2, .., N-1$, so nothing can be said about the total confidence level.

We remark that if α is chosen to zero, this test yields the simultaneous ML estimate of k and ν,

$$(\widehat{k,\nu}) = \arg\max_{k,\nu} p(Y_1^N|k,\nu),$$

where the convention is that $\hat{k} = N$ means no jump.

Since we want to maximize the likelihood with respect to ν, its dependence on the likelihood must be made explicit. This task is solved in the first subsection. The computation of the test statistic is treated in the following subsections.

3.2 A Linear Regression in the Jump Magnitude

First, process the measurements through a Kalman filter acting under the hypotheses of no jump. With notation as in Algorithm 2.1, denote its state estimate, gain, prediction error and prediction error covariance by

$$\hat{x}_{t|t},\quad K_t = P_{t|t-1}H_tV_t^{-1},\quad \epsilon_t = y_t - H_t\hat{x}_{t|t-1},\quad V_t = R_t + H_tP_{t|t-1}H_t^T.$$

Next, suppose that there was a jump ν at time k. Because of the linear model, the dependence of ν on the state estimates and innovations will be linear as well. That is, we can postulate the following model,

$$\hat{x}_{t|t}(k) = \hat{x}_{t|t} + \mu_t(k)\nu \quad (9)$$

$$\epsilon_t = \epsilon_t(k) + \varphi_t^T(k)\nu. \quad (10)$$

We have the following update formulas for the $n \times n$ matrix $\mu_t(k)$ and the $n \times 1$ vector $\varphi_t(k)$. This result was first given in [10].

Result 3.1 *The relation between the residuals from a Kalman filter conditioned on a jump of magnitude ν at time k and from a Kalman filter conditioned on no jump is given by the linear regression*

$$\epsilon_t = \epsilon_t(k) + \varphi_t^T(k)\nu,$$

where $\epsilon_t(k)$ is a white noise sequence with variance V_t. Here $\varphi_t(k)$ is computed recursively by

$$\varphi_{t+1}^T(k) = H_{t+1}\left(\prod_{i=k}^{t} F_i - F_t\mu_t(k)\right) \tag{11}$$

$$\mu_{t+1}(k) = F_t\mu_t(k) + K_{t+1}\varphi_{t+1}^T(k), \tag{12}$$

with the initial conditions $\mu_k(k) = 0$ and $\varphi_k(k) = 0$.

Proof: The result is proved by induction. The initial condition is a direct consequence of the signal model. The induction step follows from the Kalman equations, (9) and (10). First we have the relation $x_t(k) = x_t + \prod_{i=k}^{t-1} F_i\nu$. This and equations (9) and (10) give

$$\begin{aligned}
\varphi_{t+1}^T\nu &= \epsilon_{t+1} - \epsilon_{t+1}(k) \\
&= H_{t+1}\left(x_{t+1}(k) - x_{t+1}\right) - H_{t+1}F_t\left(\hat{x}_{t|t}(k) - \hat{x}_{t|t}\right) \\
&= H_{t+1}\left(\prod_{i=k}^{t} F_i - F_t\mu_t(k)\right)\nu,
\end{aligned}$$

and

$$\begin{aligned}
\mu_{t+1}(k)\nu &= \hat{x}_{t+1|t+1}(k) - \hat{x}_{t+1|t+1} \\
&= F_t\left(\hat{x}_{t|t}(k) - \hat{x}_{t|t}\right) + K_{t+1}\left(\epsilon_{t+1} + \varphi_{t+1}^T\nu - \epsilon_{t+1}\right) \\
&= F_t\mu_t(k)\nu + K_{t+1}\varphi_{t+1}^T\nu.
\end{aligned}$$

Since this holds for all ν, the result follows by induction. □

The detection problem is thus moved from a state space to a linear regression framework.

3.3 The Test with a Deterministic Jump

We begin by deriving the classical GLR test, first given in [10], where the jump magnitude is considered as deterministic. Since the transformation from measurement to residuals is linear with a Jacobian equal to one we have

$$p(Y_1^N) = p(\epsilon_1^N),$$

where

$$\epsilon_t \in \mathrm{N}(\varphi_t^T(k)\nu, V_t). \tag{13}$$

Thus, the test statistic (8) can be written as

$$l_N(k) = 2\log\frac{p(\epsilon_1^N|k,\hat{\nu}(k))}{p(\epsilon_1^N|k=N)}. \tag{14}$$

By Result 3.1, we have transformed the original state space problem into a linear regression problem. Introduce the well known least squares quantities

$$f_N(k) = \sum_{t=1}^{N} \varphi_t(k)\epsilon_t \tag{15}$$

$$R_N(k) = \sum_{t=1}^{N} \varphi_t(k)\varphi_t^T(k). \tag{16}$$

Then the ML estimate of ν, given the jump instant k, can be written

$$\hat{\nu}(k) = R_N^{-1}(k) f_N(k). \tag{17}$$

We get

$$l_N(k) = 2\log \frac{p(\epsilon_1^N | k, \hat{\nu}(k))}{p(\epsilon_1^N | k = N)} \tag{18}$$

$$= \sum_{t=k+1}^{N} \epsilon_t^T V_t^{-1} \epsilon_t - (\epsilon_t - \varphi_t^T(k)\hat{\nu}(k))^T V_t^{-1} (\epsilon_t - \varphi_t^T(k)\hat{\nu}(k)) \tag{19}$$

$$= f_N^T(k) R_N^{-1}(k) f_N(k), \tag{20}$$

where the second equality follows from (13) and the Gaussian probability density function.

Thus, the GLR algorithm can be implemented as follows.

Algorithm 3.1 *Given the signal model (1).*

- *Calculate the innovations from a Kalman filter assuming no jump.*
- *Update the regressors $\varphi_t(k)$ and the linear regression quantities $R_t(k)$ and $f_t(k)$ for each t and $1 \leq k \leq t$.*
- *At time $t = N$, the test statistic is given by $l_N(k) = f_N^T(k) R_N^{-1}(k) f_N(k)$.*
- *A jump candidate is given by $\hat{k} = \arg\max l_N(k)$. This is accepted if $l_N(\hat{k})$ is greater than some threshold (otherwise $\hat{k} = N$) and the corresponding estimate of the jump magnitude is given by $R_N^{-1}(\hat{k}) f_N(\hat{k})$.*

We make some remarks of the algorithm.

- The regressor $\varphi_t(k)$ is called a *failure signature matrix* in [10].
- The regressors are pre-computable. Furthermore, if the system and the Kalman filter is time invariant, the regressor is only a function of $t - k$ which simplifies the calculations.
- The formulation in Algorithm 3.1 is off-line. Since the test statistic involves a matrix inversion of R_N, a more efficient on-line method is as follows. Use the ordinary RLS update formulas which give $\hat{\nu}_t(k)$ and $P_t = R_t^{-1}$. Then the test statistic can be computed as

  $$l_t(k) = f_t(k)\hat{\nu}_t(k).$$

The choice of threshold is tricky. It is intimately connected to the choice of noise variance as can be seen from the following example. Suppose that all covariance matrices are scaled a factor λ,

$$\overline{\Lambda}_t = \lambda \Lambda_t, \;\; \overline{Q}_t = \lambda Q_t, \;\; \overline{\Pi}_0 = \lambda \Pi_0.$$

This scaling does not influence the filtered state estimates since it is well known that the tracking ability of the Kalman filter is invariant under such scalings. Now $\overline{V}_t = \lambda V_t$ and from (19) it follows that

$$\overline{l}_N(k) = l_N(k)/\lambda.$$

This implies that if all covariance matrices are scaled a factor λ, then the optimal threshold should be scaled a factor λ as well. Thus, it is the ratio between the noise variance and the threshold that determines the detection ability, so the problem formulation is over-parameterized.

As already noted, this test yields the ML estimate of ν and k and we have $\hat{k}_s = \arg\max_k \max_\nu p(Y_1^N|k,\nu)$. Note that since $R_N^{-1}(k)$ is positive definite, $l_N(k)$ is always positive. This means that $p(Y_1^N|k,\hat{\nu}(k)) > p(Y_1^N|k=N)$. The important conclusion is that the ML estimator of (k,ν) will never estimate $\hat{k}_s = N$. That is, a jump will always be detected if the threshold is chosen to zero, so one has to choose the threshold greater than zero. This is clear intuitively, because $\hat{\nu}(k) = 0$ is always a possibility and in fact $p(Y_1^N|k=N) = p(Y_1^N|k,\nu=0)$.

Next, the same linear regression approach is used to compute the single ML estimate of k, and the estimators are compared. It turns out that $\hat{k} = N$ is perfectly possible in this case. Since the effect of the jump magnitude is to be eliminated by integration, it has to be considered as a stochastic variable, in contrast to the approach above.

3.4 The Test with a Stochastic Jump

The idea here is to consider the jump magnitude as a stochastic variable. In this way the maximization over ν in $l_N(k)$ (7) can be avoided. Instead, the jump magnitude is eliminated by integration,

$$p(\epsilon_1^N) = \int p(\epsilon_1^N|\nu)p(\nu)d\nu.$$

Here, there are two choices. Either ν is assumed to have a Gaussian prior or it is considered as having infinite variance, so the (improper) prior is constant. Note that the former assumption will yield exactly the same likelihood as in Algorithm 2.1. The log likelihood ratio is given in the following theorem.

Theorem 3.1 (GLR estimator of the jump instant) *Consider the GLR test in Algorithm 3.1 as a ML estimator of k and ν, $k = 1,2,...,N-1$, for the signal* *model (1), given the measurements Y_1^N. The test statistic for the ML estimator of k alone is given by*

$$2\log\frac{p(Y_1^N|k)}{p(Y_1^N)} = l_N(k) + C_{prior}(k)$$

where $l_N(k)$ is given in Algorithm 3.1. Here $C_{prior}(k)$ is a prior dependent constant which equals

$$C_{prior}(k) = n\log(2\pi) - \log\det R_N(k) \tag{21}$$

if the prior is chosen to be non-informative and

$$C_{prior}(k) = -\log\det R_N(k) - \log\det P_\nu + (\hat{\nu}_N(k)-\nu_0)^T P_\nu^{-1}(\hat{\nu}_N(k)-\nu_0) \tag{22}$$

if the prior is chosen to be Gaussian, $\nu \in N(\nu_0, P_\nu)$. Here $\hat{\nu}_N(k)$ and $R_N(k)$ are given by (17) and (16), respectively.

Proof: We begin with the Gaussian case.

$$\begin{aligned}
2\log\frac{p(Y_1^N|k)}{p(Y_1^N)} &= 2\log\frac{p(\epsilon_1^N|k)}{p(\epsilon_1^N)} \\
&= 2\log\frac{p(\epsilon_1^k)p(\epsilon_{k+1}^N)}{p(\epsilon_1^k)p(\epsilon_{k+1}^N|\epsilon_1^k)} \\
&= \sum_{t=k+1}^{N} \epsilon_t^T V_t^{-1}\epsilon_t - (\epsilon_t - \varphi_t^T(k)\hat{\nu}_N(k))^T V_t^{-1}(\epsilon_t - \varphi_t^T(k)\hat{\nu}_N(k)) \\
&\quad + \log\det P_N(k) - \log\det P_\nu + (\hat{\nu}_N(k)-\nu_0)^T P_\nu^{-1}(\hat{\nu}_N(k)-\nu_0) \\
&= l_N(k) + C_{prior}(k)
\end{aligned}$$

The second equality follows from the fact that the innovations before the jump are white and do not contain any information about the jump magnitude. In the third equality the off-line expression from Lemma A.1 is used for $p(\epsilon_{k+1}^N)$.

The case of the non-informative prior is proved as above, where Lemma A.2 is used for $p(\epsilon_{k+1}^N)$.

$$\begin{aligned}
2\log\frac{p(Y_1^N|k)}{p(Y_1^N)} &= 2\log\frac{p(\epsilon_1^k)p(\epsilon_{k+1}^N)}{p(\epsilon_1^k)p(\epsilon_{k+1}^N|\epsilon_1^k)} \\
&= \sum_{t=k+1}^{N} \epsilon_t^T V_t^{-1}\epsilon_t - (\epsilon_t - \varphi_t^T(k)\hat{\nu}_N(k))^T V_t^{-1}(\epsilon_t - \varphi_t^T(k)\hat{\nu}_N(k)) \\
&\quad + \log\det P_N(k) + n\log(2\pi) \\
&= l_N(k) + C_{prior}(k)
\end{aligned}$$

Here it is used that $P_N(k) = R_N^{-1}(k)$ □

The conclusion is that the ML estimates $\widehat{(k,\nu)}$ and $\hat{k}$ are closely related. In fact, they are asymptotically equivalent. The term $(\hat{\nu}_N - \nu_0)^T P_\nu^{-1}(\hat{\nu}_N - \nu_0)$ is negligible if P_ν or, since $\hat{\nu}_N(k)$ converges either to 0 or to ν, if $N-k$ are large. As will be seen in the simulation section, the term $\log\det R_N(k)$ does not alter the likelihood significantly. This is formally proved in the following lemma.

Lemma 3.1 *If the signal model (1) is stable and time-invariant then*

$$R_N(k) = \sum_{t=k+1}^{N} \varphi_t(k)\varphi_t(k)^T$$

converges as $N-k$ tends to infinity. Here $\varphi_t(k)$ is given in Result 3.1.

Proof: Rewriting (12) using (11) gives

$$\mu_{t+1}(k) = F(I-\overline{K}H)\mu_t(k) + \overline{K}HF^{t-k},$$

where $\overline{K}$ is the stationairy Kalman gain in the measurement update. Now by assumption $\lambda_1 = \max\lambda_i(F) < 1$ so the Kalman filter theory gives that $\lambda_2 = \max\lambda_i(F - F\overline{K}H) < 1$ as well, see [1] page 77 (they define $\overline{K}$ as $F\overline{K}$). Thus $\overline{K}HF^{t-k}$ is bounded and

$$\|\mu_t(k)\| < C_1\lambda_2^{t-k}.$$

This implies that

$$\|\varphi_t(k)\| < C_2\lambda^{t-k}$$

where $\lambda = \max(\lambda_1, \lambda_2)$. Now if $l > m$

$$\begin{aligned} \|R_m(k) - R_l(k)\| &= \left\| \sum_{t=m+1}^{l} \varphi_t(k)\varphi_t^T(k) \right\| \leq \sum_{t=m+1}^{l} \left\|\varphi_t(k)\varphi_t^T(k)\right\| \\ &\leq \sum_{t=m+1}^{l} C_2^2\lambda^{2(t-k)} = C_2^2\lambda^{2(m+1-k)}\frac{1-\lambda^{2l}}{1-\lambda} \to 0 \text{ as } m, l \to \infty. \end{aligned}$$

Thus, we have proved that $R_N(k)$ is a Cauchy sequence and since it belongs to a complete space it converges. □

We have now proved that the log likelihoods for $(\widehat{k,\nu})$ and $\hat{k}$ is approximately equal for $k = 1, 2, .., N-1$, except for an unknown constant. Thus, they are likely to give the same ML estimate. Note however that this result does not hold for $k = N$, that is no jump. To get equivalence for this case as well, the threshold in the GLR test has to be chosen to this unknown constant.

4 TWO FILTER OPTIMAL DETECTION

In this section we will make a new derivation of the ML estimator of k, by merging the information of two Kalman filters, one running forwards and one backwards in time. In the first subsection the appropiate backward model of (1) is given. Then the two filter jump estimator is derived. Finally, a discussion of how this off-line approach can be used on-line is given.

4.1 A Backwards Markovian Model

As mentioned in the introduction we need a Kalman filter running backwards in time. This is a well known problem in the context of fixed-interval smoothing. One solution here, which suits our purposes, is to first derive the appropiate backwards model of (1) and then just apply the usual Kalman filter to this model. This will give the conditional mean $\hat{x}^B_{t|t+1}$ and the conditional covariance $P^B_{t|t+1}$, given the measurements Y^N_{t+1}, of the Gaussian variable x_t. The following lemma gives the desired backwards Markovian model which is sample path equivalent to (1). That it is Markovian implies that the noise process $\{w_t\}$ and the initial value of the state vector x_N are independent. It is not only the first and second order statistics that are equal for these two models but they are really sample path equivalent, since they both produce the same state and output vectors.

Lemma 4.1 *The following model is sample path equivalent to (1) (if $k > N$) and is its corresponding backwards Markovian model:*

$$\begin{aligned} x_t &= F^B_t x_{t+1} + w^B_{t+1} \\ y_t &= H_t x_t + e_t. \end{aligned} \tag{23}$$

Here

$$\Pi_{t+1} = F_t \Pi_t F_t^T + Q_t \tag{24}$$

$$F^B_t = \Pi_t F_t^T \Pi^{-1}_{t+1} = F_t^{-1} - F_t^{-1} Q_t \Pi^{-1}_{t+1} \tag{25}$$

$$Q^B_t = \Pi^T_t - \Pi_t F^T_t \Pi^{-1}_{t+1} F_t \Pi_t = F_t^{-1} Q_t F_t^{-T} - F_t^{-1} Q_t \Pi^{-1}_{t+1} Q_t F_t^{-T} \tag{26}$$

where $\Pi_t = E[x_t x_t^T]$ is the a priori *covariance matrix of the state vector.*

Proof: See [8].

We want a backwards Markovian model in the interval $[k+1, N]$ for all possible values of k. The problem here, which is not apparent in smoothing, is that the final value Π_N generally depends on k and is thus different for each hypothesis. Fortunately, thanks to the assumption of infinite variance of the jump magnitude, we have the following result.

Corollary 4.1 *The backwards Markovian model for (1) after a jump (i.e. $k < t \leq N$) is*

$$\begin{aligned} x_t &= F_t^{-1} x_{t+1} + F_t^{-1} w_t = F_t^{-1} x_{t+1} + w^B_t \\ y_t &= H_t x_t + e_t. \end{aligned} \tag{27}$$

Here $Q^B_t = E[w^B_t (w^B_t)^T] = F_t^{-1} Q_t F_t^{-T}$.

Proof: Since the jump magnitude has infinite covariance matrix we have that Π_{k+1} is infinite or, equivalently, $\Pi^{-1}_{k+1} = 0$. Equation (24) now gives

$$\Pi^{-1}_t = 0, \quad t = k+1, ..., N \tag{28}$$

so Π_N is independent of the jump time. The last versions of equations (25) and (26) now give the model (27). □

We now have the backward model and can simply apply the Kalman filter for the estimate $x^B_{t|t+1}$ and its covariance matrix $P^B_{t|t+1}$.

4.2 The Two-Filter Implementation

We are now ready for the main result of this section.

Theorem 4.1 *The likelihood of the measurements for the signal model (1), conditioned on a jump at time k, can be computed by two Kalman filters as follows. First, the likelihoods are separated,*

$$p(Y_1^N|k) = p(Y_1^k)p(Y_{k+1}^N). \tag{29}$$

The first factor is computed by

$$p(Y_1^k) = \prod_{t=1}^{k} \gamma(y_t - H_t^T \hat{x}_{t|t-1}^F, H_t P_{t|t-1}^F H_t^T + \Lambda_t). \tag{30}$$

Here $\gamma(x-\mu, P)$ is the Gaussian probability density function, with mean μ and covariance P, evaluated in x. The quantities $\hat{x}_{t|t-1}^F$ and $P_{t|t-1}^F$ are given by the Kalman filter applied on the forward model (1). The second factor is computed by

$$p(Y_{k+1}^N) = \prod_{t=k+1}^{N} \gamma(y_t - H_t^T \hat{x}_{t|t+1}^B, H_t P_{t|t+1}^B H_t^T + \Lambda_t). \tag{31}$$

The quantities $\hat{x}_{t|t+1}^B$ and $P_{t|t+1}^B$ are given by the Kalman filter applied on the backward model (27) with the final value $\Pi_N^{-1} = 0$.

Proof: Bayes' law gives

$$p(Y_1^N|k) = p(Y_1^k|k)p(Y_{k+1}^N|k, Y_1^k).$$

The jump at time k does not affect the measurements before time k so $p(Y_1^k|k) = p(Y_1^k)$. The jump and its infinite covariance matrix imply that the measurements before time k contain no information about the measurements after time k. It also makes the *a priori* covariance matrix of the state vector infinite ($\Pi_t^{-1} = 0,\ t > k$) but does not affect the data in any other way, so (29) follows.

The first factor of (29) is next expanded by iteratively using Bayes' law,

$$p(Y_1^k) = \prod_{t=1}^{k} p(y_t|Y_1^{t-1}).$$

Since x_t is Gaussian it follows immediately that $y_t|Y_1^{t-1}$ is Gaussian with mean $H_t\hat{x}_{t|t-1}^F$ and covariance $H_t P_{t|t-1}^F H_t^T + \Lambda_t$ and (30) follows. The second factor is expanded backwards in time in the same way,

$$p(Y_{k+1}^N) = \prod_{t=k+1}^{N} p(y_t|Y_{t+1}^N).$$

Now $y_t|Y_{t+1}^N$ is Gaussian with mean $H_t\hat{x}_{t|t+1}^B$ and covariance $H_t P_{t|t+1}^B H_t^T + \Lambda_t$ and (31) follows. □

As seen, all that is needed to compute the likelihoods are one Kalman filter running backwards in time and one running forwards in time. The resulting algorithm is as follows, where the log likelihoods are used because of possible numerical problems caused by very large differences in the magnitude of the likelihoods.

Algorithm 4.1 (Two-filter detection) *The likelihood of a jump at time k, $k = 1, 2, .., N$, is computed with two filters as follows.*

Forward filter

$$
\begin{aligned}
\hat{x}^F_{1|0} &= x_0 \\
P^F_{1|0} &= \Pi_0 \\
\epsilon^F_t &= y_t - H_t^T \hat{x}^F_{t|t-1} \\
V^F_t &= H_t P^F_{t|t-1} H_t^T + \Lambda_t \\
\hat{x}^F_{t+1|t} &= F_t \hat{x}^F_{t|t-1} + F_t P^F_{t|t-1} H_t (V^F_t)^{-1} \epsilon^F_t \\
P^F_{t+1|t} &= F_t \left(P^F_{t|t-1} - P^F_{t|t-1} H_t^T (V^F_t)^{-1} H_t P^F_{t|t-1} \right) F_t^T + Q_t \\
\log p(Y_1^t) &= \log p(Y_1^{t-1}) - \frac{1}{2} \log 2\pi + \frac{1}{2} \log \det (V^F_t)^{-1} - \frac{1}{2} (\epsilon^F_t)^T (V^F_t)^{-1} \epsilon^F_t
\end{aligned}
$$

Backward filter

$$
\begin{aligned}
\hat{x}^B_{N|N+1} &= 0 \\
(P^B_{N|N+1})^{-1} &= 0 \\
\epsilon^B_t &= y_t - H_t^T \hat{x}^B_{t|t+1} \\
V^B_t &= H_t P^B_{t|t+1} H_t^T + \Lambda_t \\
\hat{x}^B_{t-1|t} &= F_t^{-1} \hat{x}^B_{t|t+1} + F_t^{-1} P^B_{t|t+1} H_t (V^B_t)^{-1} \epsilon^B_t \\
P^B_{t-1|t} &= F_t^{-1} \left(P^B_{t|t+1} - P^B_{t|t+1} H_t^T (V^B_t)^{-1} H_t P^B_{t|t+1} \right) F_t^{-T} + F_t^{-1} Q_t F_t^{-T} \\
\log p(Y_t^N) &= \log p(Y_{t+1}^N) - \frac{1}{2} \log 2\pi + \frac{1}{2} \log \det (V^B_t)^{-1} - \frac{1}{2} (\epsilon^B_t)^T (V^B_t)^{-1} \epsilon^B_t
\end{aligned}
$$

Compute the likelihoods *Compute and save*

$$\log p(Y_1^N | k) = \log p(Y_1^k) + \log p(Y_{k+1}^N).$$

We make some remarks on the algorithm.

- The *a posteriori* probability of k is easily computed by using Bayes' law,

  $$p(k|Y_1^N) = \frac{p(k) p(Y_1^N|k)}{p(Y_1^N)} = \frac{p(Y_1^N|k)}{\sum_{i=1}^N P(Y_1^N|i)}.$$

  This means that the probability of a wrong decision can be computed as $1 - p(\hat{k}|Y_1^N)$.

- The backward filter is not completely specified, since $P^B_{t|t+1}$ is not defined for large values of t. The practical solution is to let $P_{N|N+1}$ be a very large constant times the unit matrix. The theoretical solution is to use the *information filter* where only the inverse of $P^B_{t|t+1}$ is used, see [1], until the inverse covariance matrix is invertible. The problem here is how to treat the likelihoods.

- The relation to fixed-interval smoothing is as follows. The smoothed estimates under the no jump hypothesis can be computed, with little extra work, as follows

$$\begin{aligned} P_{t|N} &= \left((P^F_{t|t})^{-1} + (P^B_{t|t+1})^{-1}\right)^{-1} \\ \hat{x}_{t|N} &= P_{t|N}\left((P^F_{t|t})^{-1}\hat{x}^F_{t|t} + (P^B_{t|t+1})^{-1}\hat{x}^B_{t|t+1}\right). \end{aligned}$$

  Here $\hat{x}^F_{t|t}$ and $P^F_{t|t}$ are the filtered estimates from the forward filter (which are not given explicitely above).

It should be stressed that it is necessary for this two-filter implementation that the jump is considered as stochastic with infinite variance, which implies the important separability possibility (29).

4.3 On-line Implementation

Algorithm 4.1 performs off-line optimal detection with just two Kalman filters. The straightforward on-line implementation requires t filters at time t, so at time t is t Kalman filters iteratively updated. That is, a total of $\sum_{t=1}^{N} t = N(N+1)/2$ Kalman filter iterations are performed. We will now investigate how the off-line algorithm can be used to do optimal detection batch-wise on the data. We feel that it is convenient to compare the computational complexity of different methods in how many Kalman filters that have to be used in parallel in the mean, so we introduce the measure Kalman Filter Units, abbreviated KFU. The true on-line implementation thus requires $(N+1)/2$ KFU.

First, we assume that we can tolerate a delay of detection of M time intervals or that we collect the data in batches of length M. The measurements are divided into P segments, each of length M, where $N = MP$. The idea is that Algorithm 4.1 can be applied after each batch of data is collected. After batch k, the backward filter is started at time kM. The same forward filter can be used all the time so it does not have to be restarted. Thus, the number of Kalman filter iterations is

$$N + \sum_{k=1}^{P} kM = N + M\frac{P(P+1)}{2} = N\frac{P+3}{2}.$$

This corresponds to $(P+3)/2$ KFU, which is approximately a reduction of M compared to the direct implementation.

If we want to have a truly recursive algorithm the number of KFU must converge. One solution is to let the batches, and thus also the delay of detection, increase exponentially. Thus we let the k'th batch have length

$$M_k = (1 + \frac{1}{L})^k,$$

where now $N = \sum_{k=1}^{P}(1+1/L)^k$ (choose the nearest integer lengths for the batches). The number of Kalman filter iterations is now

$$N + \sum_{k=1}^{P} M_k = N + \frac{(1+1/L)^{P+1} - 1}{(1+1/L) - 1} \approx N + LN(1+1/L).$$

If the batches are doubled each time, so $L = 1$, then 3 KFU are needed. With a very moderate increase, so L is large, $L + 2$ KFU are needed.

We conclude this section by summarizing the results.

- A direct (on-line) implementation of the jump estimator requires $(N + 1)/2$ KFU.
- The off-line implementation in Algorithm 4.1 requires 2 KFU.
- A batch-wise (length M) implementation using Algorithm 4.1 requires $N/(2M)$ KFU.
- With a constant number L ($>> 1$) of KFU, a batchwise implementation is still possible if the length of the batches increases as $(1 + 1/L)^k$.

The final conclusion is that much can be gained also in on-line applications by the two filter detection algorithm.

5 STOCHASTIC NOISE VARIANCE

In this section we will derive the ML estimator of the jump instant for the case of an unknown noise variance. The assumption is that the measurement noise covariance matrix contains an unknown scaling factor λ, which simplifies the analysis in the multi output case.

The signal model (1) has to be modified slightly. All covariance matrices are assumed to be scaled a factor λ,

$$\overline{\Lambda} = \lambda\Lambda, \quad \overline{P}_0 = \lambda P_0, \quad \overline{Q} = \lambda Q. \tag{32}$$

Here matrices with bars are the true ones and the other are chosen by the user. This means that one chooses the tracking ability of the filter, which is insensitive to scalings. The estimator then estimates the actual level, which is decisive for the likelihoods.

The noise variance will be considered as a stochastic variable, either with completely non-informative prior, so $p(\lambda) = 1$, or as being inversely Wishart distributed. The inverse Wishart distribution will be denoted by $W^{-1}(m, \sigma)$ and it has the probability density function

$$f(\lambda) = \frac{\sigma^{m/2}}{2^{m/2}\Gamma(m/2)} \frac{\exp\left(-\frac{\sigma}{2\lambda}\right)}{\lambda^{(m+2)/2}}. \tag{33}$$

This distribution has mean $\sigma/(m-2)$ and variance $2\sigma^2/((m-2)^2(m-4))$. Figure 1 shows the probability density function (33) with mean 1 and variance 1.

The following theorem is the counterpart to the two filter detection method in Algorithm 4.1, for the case of an unknown λ.

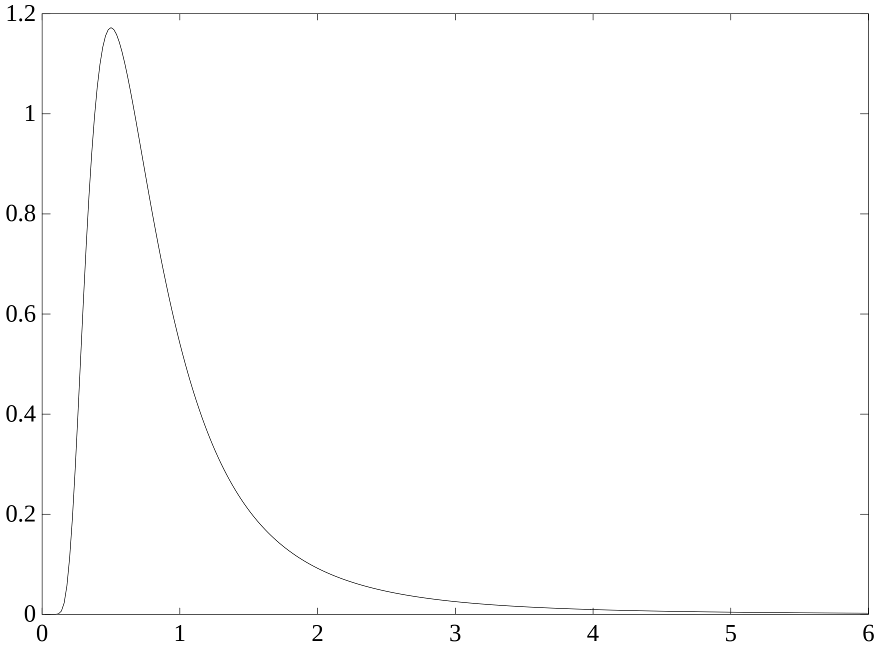

Figure 1: The probability density function for the inverse Wishart distribution.

Theorem 5.1 *Consider the signal model (1) in the case of an unknown noise variance and suppose that (32) holds. With notations from the two filter detector in Algorithm 4.1, let*

$$S^F(k) = \sum_{t=1}^{k} (\epsilon_t^F)^T (V_t^F)^{-1} \epsilon_t^F \tag{34}$$

$$S^B(k) = \sum_{t=k+1}^{N} (\epsilon_t^B)^T (V_t^B)^{-1} \epsilon_t^B \tag{35}$$

$$D^F(k) = -\prod_{t=1}^{k} \det V_t^F \tag{36}$$

$$D^B(k) = -\prod_{t=k+1}^{N} \det V_t^B \tag{37}$$

The log likelihood is then given by

$$\begin{aligned} \log p(Y_1^N|k) &= -\frac{N}{2}\log\pi - \frac{m}{2}\log\sigma + \log\Gamma\left(\frac{m+N}{2}\right) - \log\Gamma\left(\frac{m}{2}\right) && (38)\\ &\quad -\frac{1}{2}\log D^F(k)D^B(k) - \frac{m+N}{2}\log\left(\sigma + S^F(k) + S^B(k)\right) && (39)\end{aligned}$$

if the prior of λ is $W^{-1}(m,\sigma)$ and

$$\begin{aligned} \log p(Y_1^N|k) &= -\frac{N}{2}\log\pi + \log\Gamma\left(\frac{N-2}{2}\right) - \log 2 && (40)\\ &\quad -\frac{1}{2}\log D^F(k)D^B(k) - \frac{N-2}{2}\log\left(S^F(k) + S^B(k)\right) && (41)\end{aligned}$$

if the prior of λ is flat. Furthermore, the a posteriori *distribution for λ, given the jump instant k, is $W^{-1}(m+N, \sigma + S^F(k) + S^B(k))$. Here $m = -2$ and $\sigma = 0$ if the non-informative prior is used.*

Proof: We have

$$\begin{aligned} p(Y_1^N|k) &= \int p(Y_1^N|k,\lambda)p(\lambda|k)d\lambda \\ &= \int p(Y_1^N|k,\lambda)p(\lambda)d\lambda. \end{aligned}$$

The likelihood in the integral is given by Algorithm 4.1 as

$$p(Y_1^N|k,\lambda) = \left(\frac{1}{2\pi\lambda}\right)^{N/2} (D^F(k)D^B(k))^{-1/2} \exp\left(-\frac{S^F(k)+S^B(k)}{2\lambda}\right).$$

Starting with the case of $p(\lambda) = 1$ we get

$$\begin{aligned} p(Y_1^N|k) &= \left(\frac{1}{2\pi}\right)^{N/2} (D^F(k)D^B(k))^{-1/2} \frac{\Gamma\left(\frac{N-2}{2}\right)2^{(N-2)/2}}{(S^F(k)+S^B(k))^{(N-2)/2}} \\ &\quad \int \frac{\left(S^F(k)+S^B(k)\right)^{(N-2)/2}}{\Gamma\left(\frac{N-2}{2}\right)2^{(N-2)/2}} \frac{\exp\left(-\frac{S^F(k)+S^B(k)}{2\lambda}\right)}{\lambda^{(N-2+2)/2}} d\lambda \\ &= \left(\frac{1}{2\pi}\right)^{N/2} (D^F(k)D^B(k))^{-1/2} \frac{\Gamma\left(\frac{N-2}{2}\right)2^{(N-2)/2}}{(S^F(k)+S^B(k))^{(N-2)/2}}. \end{aligned}$$

The integral equals one because the integrand is a probability density function, namely the inverse Wishart distribution (33). The case when $p(\lambda)$ is given by (33) is treated similarily,

$$\begin{aligned} p(Y_1^N|k) &= \left(\frac{1}{2\pi}\right)^{N/2} (D^F(k)D^B(k))^{-1/2} \frac{\sigma^{m/2}}{\Gamma\left(\frac{m}{2}\right)2^{m/2}} \frac{\Gamma\left(\frac{N+m}{2}\right)2^{(N+m)/2}}{(\sigma+S^F(k)+S^B(k))^{(N+m)/2}} \\ &\quad \int \frac{\left(\sigma+S^F(k)+S^B(k)\right)^{(N+m)/2}}{\Gamma\left(\frac{N+m}{2}\right)2^{(N+m)/2}} \frac{\exp\left(-\frac{\sigma+S^F(k)+S^B(k)}{2\lambda}\right)}{\lambda^{(N+m+2)/2}} d\lambda \\ &= \left(\frac{1}{2\pi}\right)^{N/2} (D^F(k)D^B(k))^{-1/2} \frac{\sigma^{m/2}}{\Gamma\left(\frac{m}{2}\right)2^{m/2}} \frac{\Gamma\left(\frac{N+m}{2}\right)2^{(N+m)/2}}{(\sigma+S^F(k)+S^B(k))^{(N+m)/2}}. \end{aligned}$$

The desired results follows by taking the logarithm of the above expressions.

The *a posteriori* distribution for λ follows from Bayes' law

$$p(\lambda|Y_1^N) = \frac{p(Y_1^N|\lambda)p(\lambda)}{p(Y_1^N)}$$

and by direct identification with (33). $\square$

Before we comment on the result, the likelihood is given for the case when the noise variance is different before and after the jump.

Theorem 5.2 *Consider the same detection problem as in Theorem 5.1 but with a noise variance changing at the jump instant,*

$$\lambda = \begin{cases} \lambda_1 & , \text{if } t \le k \\ \lambda_2 & , \text{if } k < t \le N \end{cases}$$

The log likelihood is then given by

$$\begin{aligned} p(Y_1^N|k) &= -\frac{N}{2}\log\pi - \frac{m}{2}\log\sigma - \frac{1}{2}\log D^F(k)D^B(k) && (42)\\ &+\log\Gamma\left(\frac{m+k}{2}\right) + \log\Gamma\left(\frac{m+N-k}{2}\right) - 2\log\Gamma\left(\frac{m}{2}\right) && (43)\\ &-\frac{m+k}{2}\log\left(\sigma + S^F(k)\right) - \frac{m+N-k}{2}\log\left(\sigma+S^B(k)\right) && (44) \end{aligned}$$

if the prior of λ is $W^{-1}(m,\sigma)$ and

$$\begin{aligned} p(Y_1^N|k) &= -\frac{N}{2}\log\pi - 2\log 2 - \frac{1}{2}\log D^F(k)D^B(k) && (45)\\ &+\log\Gamma\left(\frac{k-2}{2}\right) + \log\Gamma\left(\frac{N-k-2}{2}\right) && (46)\\ &-\frac{k-2}{2}\log\left(S^F(k)\right) - \frac{N-k-2}{2}\log\left(S^B(k)\right) && (47) \end{aligned}$$

if the prior of λ is flat. Furthermore, the a posteriori *distribution for λ_1, given the jump instant k, is $W^{-1}(m+k, \sigma+S^F(k))$ and for λ_2 it is $W^{-1}(m+N-k, \sigma+S^B(k))$. Here $m=-2$ and $\sigma=0$ if the non-informative prior is used.*

Proof: The proof resembles that of Theorem 5.1. The difference is that the integral over λ is here split into two integrals,

$$\begin{aligned} p(Y_1^N|k) &= p(Y_1^k)p(Y_{k+1}^N) \\ &= \int p(Y_1^k|\lambda)p(\lambda)d\lambda \int p(Y_{k+1}^N|\lambda)p(\lambda)d\lambda. \end{aligned}$$

Each integral is evaluated exactly as before. □

It is interesting to compare the likelihood in Algorithm 4.1 with the ones in Theorems 5.1 and 5.2.

- The model and data dependent quantities $S(k)$ and $D(k)$ are all given by the two filter algorithm 4.1. The decision for which likelihood is to be used can be deferred to after these quantities are computed. Especially, all three possibilities can be examined without much extra computations.

- The dominating terms are $S^F(k)$ and $S^B(k)$, as simulations have shown. When the noise is unknown, $(S^F(k)+S^B(k))/\lambda$ is essentially replaced by $N\log(S^F(k)+S^B(k))$ and $k\log S^F(k) + (N-k)\log S^B(k)$, respectively. This leads to a more cautious estimator, which diminishes the influence of the innovations.

- The term $-\frac{1}{2}\log D^F(k)D^B(k)$ appears in all likelihoods. This term does not change much for different jump instants $k = 1, 2, .., N-1$. However, it is very important for the no jump hypothesis $k = N$, because it is much larger in this case. This is why the single ML estimator of k is able to estimate $\hat{k} = N$.

- The conditional *a posteriori* distribution for λ can be used to compute the unconditional one,

$$\begin{aligned} p(\lambda|Y_1^N) &= \sum_{k=1}^{N} p(k|Y_1^N)p(\lambda|Y_1^N, k) \\ &= \sum_{k=1}^{N} \frac{p(Y_1^N|k)p(k)}{p(Y_1^N)} p(\lambda|Y_1^N, k). \end{aligned}$$

  All quantities are known so, for instance, the mean and variance for λ can be evaluated.

These methods are tested in the simulation section.

6 Simulation results

In this section the following detection methods are compared:

1. Maximizing $p(Y_1^N|k, \nu)$ for ν and $k = 1, 2, ..., N-1$. The likelihood is computed through the GLR test by one Kalman filter and $N-1$ matched filters, as given in Algorithm 3.1.

2. Maximizing $p(Y_1^N|k)$ for $k = 1, 2, ..., N$. The likelihood is computed through the GLR test by one Kalman filter and $N-1$ matched filters, as given in Algorithm 3.1.

3. Maximizing $p(Y_1^N|k)$ for $k = 1, 2, ..., N$. The likelihood is computed by an on-line implementation using N Kalman filters, as given in Algorithm 2.1.

4. Maximizing $p(Y_1^N|k)$ for $k = 1, 2, ..., N$. The likelihood is computed by an off-line implementation using 2 Kalman filters, as given in Algorithm 4.1.

5. Maximizing $p(Y_1^N|k)$ for $k = 1, 2, ..., N$. The likelihood is computed by an off-line implementation using 2 Kalman filters, as given in Theorem 5.1. The noise variance is considered as an unknown stochastic variable constant over time.

6. Maximizing $p(Y_1^N|k)$ for $k = 1, 2, ..., N$. The likelihood is computed by an off-line implementation using 2 Kalman filters, as given in Theorem 5.2. The noise variance is considered as an unknown stochastic variable, changing its value when the jump occurs.

7. Recursively maximizing $p(Y_1^t|k,\nu)$ with the restriction $k = t-M,..,t-1$. This windowing approach to approximate the GLR test is suggested in [10].

The following state space model will be used:

$$
\begin{aligned}
x_{t+1} &= \begin{pmatrix} 1 & -0.25 \\ 0 & 1 \end{pmatrix} + v_t + \delta(t-N/2)\nu \\
y_t &= (1\ 0)x_t + e_t
\end{aligned}
$$

where

$$
\begin{aligned}
v_t &\in \mathrm{N}(0, I) \\
e_t &\in \mathrm{N}(0, \lambda) \\
\nu &= (10,\ 10)^T.
\end{aligned}
$$

All filters are initialized assuming that $\hat{x}_{1|0} \in \mathrm{N}(0, 1000)$. The number of measurements is N and the state jump is in the middle of the data.

The first six methods will be compared by computing and plotting the likelihoods $p(Y_1^N|k,\hat{\nu})$ and $p(Y_1^N|k)$ for $k = 1, 2, .., N$, respectively. As in the GLR test, these likelihoods are normalized so $p(Y_1^N|k=N) = 1$.

6.1 Time Consumption

Figure 2 shows the time consumption in seconds for the different detection schemes as a function of the number of measurements, N, for $\lambda = 1$. The simulations were performed in MATLAB on a SUN SPARC station SLC. This figure is quite remarkable. First, it is commonly believed that the implementation in [10] with matched filters is very efficient. This is not true for this example, since the direct implementation is actually faster. However, both algorithms have a quadratic increase in the number of measurements. The big difference is for the two filter implementation. As expected, it shows only a linear increase in the computational complexity, where the last value in the figure equals 3.7 seconds.

The approximation by using a window for the considered jump instants is also investigated. The likelihood is computed for the last 10 time instants, so a total of 10 matched filters are used. As expected, the time consumption increases linearly with time and it is about five times slower than the two filter approach.

6.2 Likelihood Comparison

The original GLR test with matched filters as described in [10] (1 above) is not equivalent to the others, since it is a simultaneous estimate of the jump time *and* the jump magnitude, while the other ones estimate the jump time alone. Figure 3 shows the log likelihood of different jump times, averaged over 100 realizations. As seen, the difference is not large. Implementations 2-4 are equivalent as they should be. The likelihood $p(Y_1^N|k,\hat{\nu}(k))$ is quite similar, differing only for jump times

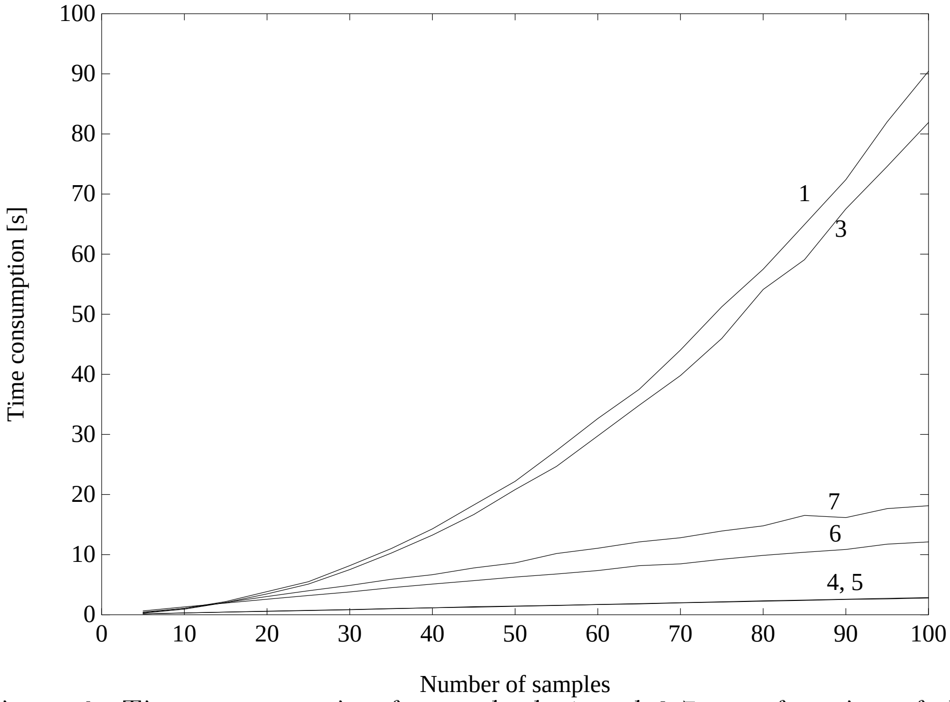

Figure 2: Time consumption for methods 1 and 3-7 as a function of the number of measurements.

near 1 and N. It should be remarked that this log likelihood is not normalized, so $\log p(Y_1^N|k = N) \approx -12$, in order to highlight the similarity to the other methods in the important interval.

For the cases when the noise variance is estimated the likelihoods look like 1-4, but the peaks are smaller. Thus, the ML estimators are more cautious when the noise variance is unknown. For the case of jumping variance, some peaks in the beginning and the end of the interval can be noted. These can be explained as follows. For an interval with very few measurements, its likelihood can be made very large by using a very small noise variance and choosing the other parameters appropiately. Thus, such peaks can always be expected and it is advisable to ignore them.

6.3 Robustness to Erroneous Noise Assumptions

Suppose now that the true noise variance is unknown, so an incorrect value is given to algorithms 1-4. Note that methods 5 and 6 is independent of the assumed noise level, so Figure 3 is still valid for these two methods. Figure 4 shows what happens when $\lambda = 10$ and Figure 5 the case $\lambda = 0.1$. We see that the case of an overestimated λ is the most serious one, because the peaks are rather small here. In fact, for methods 2-4 the peak is under the zero level, so $\hat{k} = N$. Thus, the jump is not detected and this is quite serious.

Next, it is examined what happens if there is no jump ($\nu = 0$) but the measurement noise variance increases from the assumed value of $\lambda = 1$ to $\lambda = 100$ at time $N/2$, for $N = 50$. Figure 6 shows the log likelihood averaged over 100 realizations.

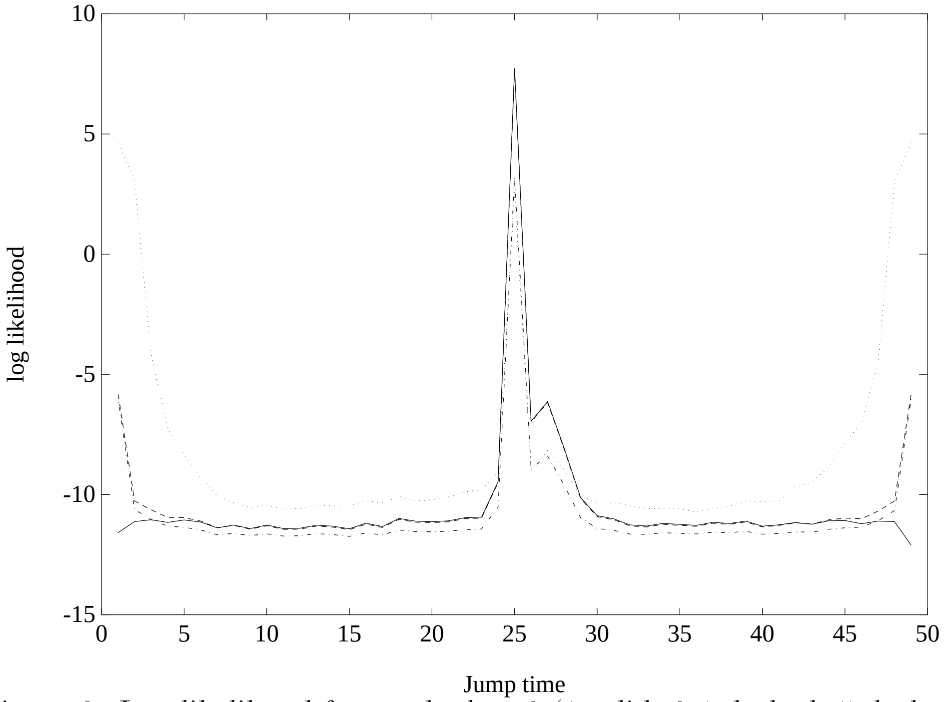

Figure 3: Log likelihood for methods 1-6 (1 solid, 2-4 dashed, 5 dashed-dotted, 6 dotted) averaged over 100 realizations.

After the increase in the noise, a lot of false peaks can be noted for methods 1-4, indicating a rather chaotic behaviour. Method 6 handles this case excellently with a peak at the true jump instant, while method 5 correctly estimates no jump since the noise variance is not allowed to change here.

The conclusion is that if the noise variance is approximately known, one of the methods 1-4 should be used. If it is completely unknown method 5 works well and, most important, if the noise variance is supposed to change method 5 or 6 must be used, dependent of if one is interested in detecting these noise changes in the noise level.

7 CONCLUSIONS

We have studied the problem of estimating the jump instant k in

$$\begin{aligned} x_{t+1} &= F_t x_t + w_t + \delta(k-t)\nu \\ y_t &= H_t^T x_t + e_t, \end{aligned}$$

given the measurements Y_1^N. There are two basically different approaches. One can compute the ML estimate of k alone or of (k, ν). We have given an explicit expression for the difference of these two approaches in Theorem 3.1, which indicates that the difference is minor and asymptotically diminishing.

The GLR test is equivalent to the ML estimator of (k, ν). This test has been re-examined here and the following properties were found, where the statements relate to a direct implementation involving N Kalman filters.

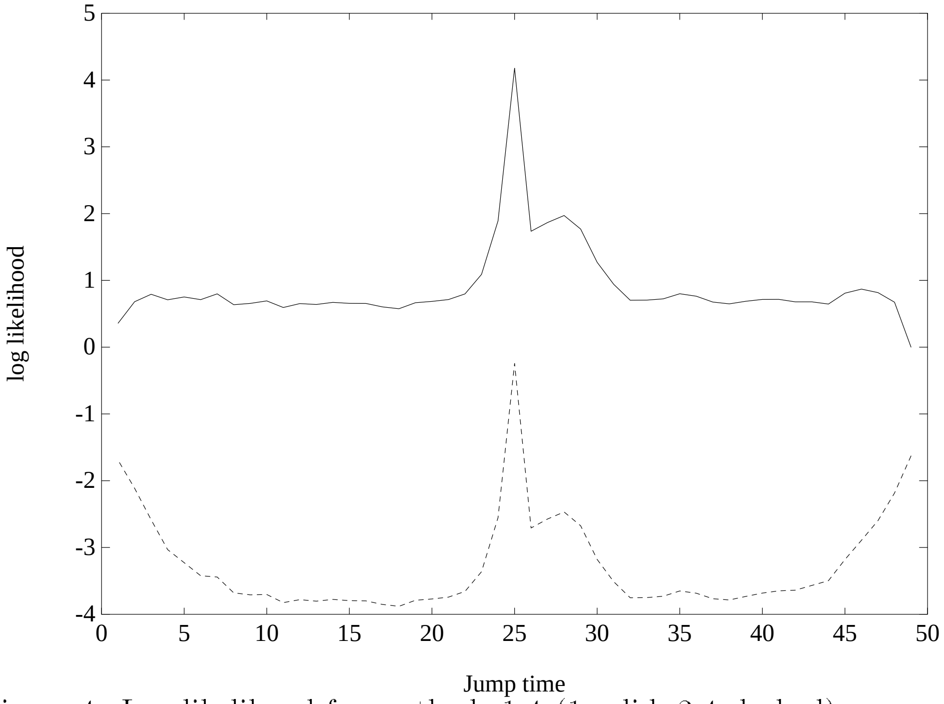

Figure 4: Log likelihood for methods 1-4 (1 solid, 2-4 dashed) averaged over 100 realizations in the case when the noise variance is overestimated a factor 10.

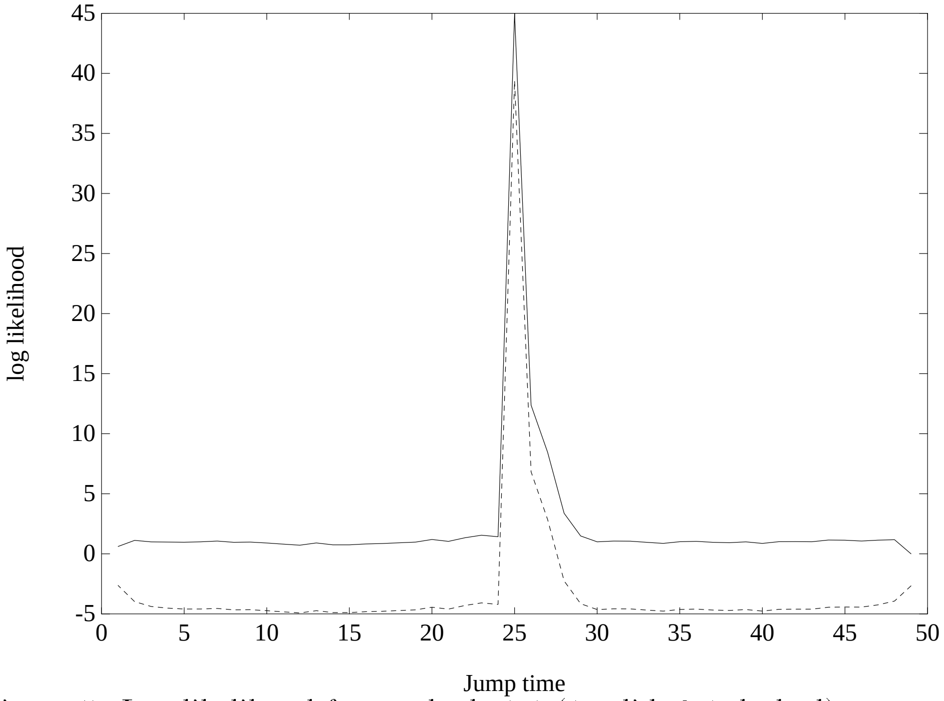

Figure 5: Log likelihood for methods 1-4 (1 solid, 2-4 dashed) averaged over 100 realizations in the case when the noise variance is underestimated a factor 10.

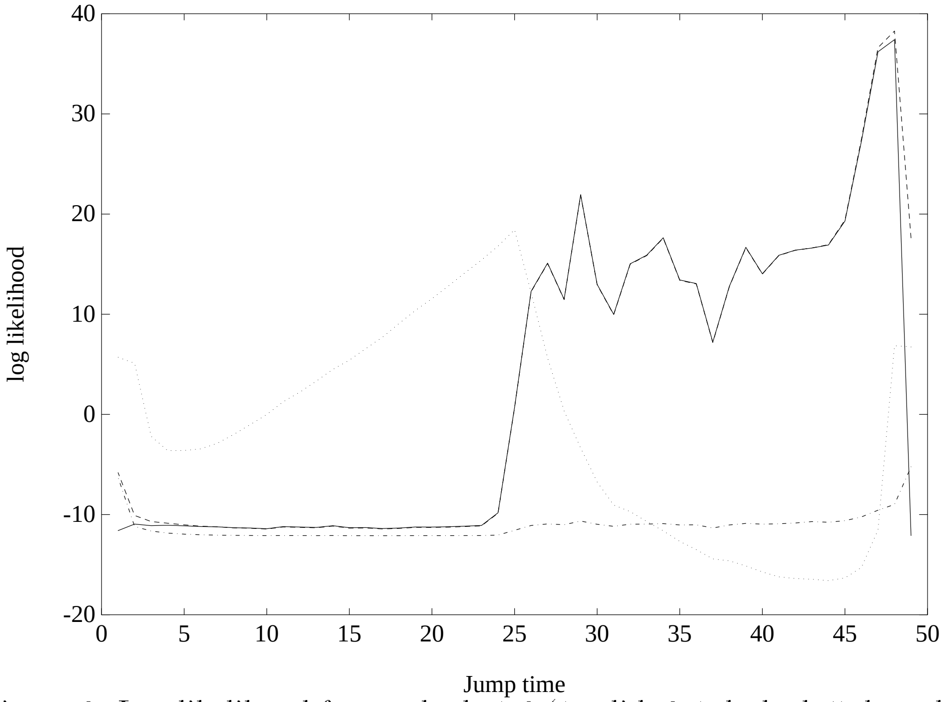

Figure 6: Log likelihood for methods 1-6 (1 solid, 2-4 dashed, 5 dotted, 6 dashed-dotted) averaged over 100 realizations when the noise variance is increased from 1 to 100 at time 25.

+ The only approach to compute the ML estimator of (k, ν).

+ This approach reveals a possibility to compare the ML estimators of k and (k, ν) explicitely. It was shown that the difference between these two likelihoods is small and tends to zero as $t - k$ tends to infinity. This result was supported by simulations.

– Requires an linearly increasing number of matched filters. Even with a proposed windowing approach, it is computationally more expensive compared to the two filter approach. Furthermore, the matched filters have turned out to be computationally more expensive than a Kalman filter in a simulation example. Thus, the method is not very efficient.

– Lack of robustness to incorrectly modeled noise variance.

– The problem formulation is overparameterized, since a scaling of all covariance matrices and the threshold will yield equivalent results.

– The GLR test is a multiple hypotheses test, so no total confidence level can be given.

– The (k, ν) estimator is dubious, because $\hat{k}_s = N$ is an impossible estimate. Instead, one has to rely on a chosen threshold, which "helps" the no jump hypothesis.

The drawbacks above are eliminated by the new result of two filter detection. In this approach we run one Kalman filter forwards in time, to get the forward innovations $\epsilon_t^F = y_t - H_t x_{t|t-1}$ and their covariance matrices $V_t^F = H_t P_{t|t-1} H_t^T + \Lambda_t$, and one Kalman filter backwards in time, to get the backward innovations $\epsilon_t^B = y_t - H_t x_{t|t+1}$ and their covariance matrices $V_t^B = H_t P_{t|t+1} H_t^T + \Lambda_t$. By using these quantities the likelihood of each jump instant can be calculated under various assumptions about the noise. Since the likelihoods are easily converted to *a posteriori* probabilities for each jump, a measure of the reliability can be assigned to every decison.

The log likelihood for k and a given noise variance is given by

$$\begin{aligned}
-2\log p(Y_1^N|k) &= N\log 2\pi + \sum_{t=1}^{k}\left(\log\det V_t^F + (\epsilon_t^F)^T (V_t^F)^{-1}\epsilon_t^F\right) \\
&\quad + \sum_{t=k+1}^{N}\left(\log\det V_t^B + (\epsilon_t^B)^T (V_t^B)^{-1}\epsilon_t^B\right).
\end{aligned}$$

Maximizing this likelihood essentially gives the same result as the GLR test. The main difference is that $\hat{k} = N$ is possible here, so no threshold has to be chosen. A simulation study shows that the likelihoods of a direct implementation and this two filter method is the same and approximately equal to the GLR test statistic.

The same two filter approach was shown to work in the case when the noise variance is unknown as well. The appropiate likelihood is now

$$\begin{aligned}
-2\log p(Y_1^N|k) &= C_1 + \sum_{t=1}^{k}\log\det V_t^F + \sum_{t=k+1}^{N}\log\det V_t^B \\
&\quad + (N-2)\log\left(\sum_{t=1}^{k}(\epsilon_t^F)^T (V_t^F)^{-1}\epsilon_t^F + \sum_{t=k+1}^{N}(\epsilon_t^B)^T (V_t^B)^{-1}\epsilon_t^B\right).
\end{aligned}$$

The case of a changing noise variance is particular tricky in filter-based detection methods. The herein proposed solution is to assume that the noise variance changes at the jump instant as well. This leads to the following likelihood,

$$\begin{aligned}
-2\log p(Y_1^N|k) &= C_2 + \sum_{t=1}^{k}\log\det V_t^F + (k-2)\log\left(\sum_{t=1}^{k}(\epsilon_t^F)^T (V_t^F)^{-1}\epsilon_t^F\right) \\
&\quad \sum_{t=k+1}^{N}\log\det V_t^B + (N-k-2)\log\left(\sum_{t=k+1}^{N}(\epsilon_t^B)^T (V_t^B)^{-1}\epsilon_t^B\right) \\
&\quad + 2\log\Gamma\left(\frac{k-2}{2}\right) + 2\log\Gamma\left(\frac{N-k-2}{2}\right).
\end{aligned}$$

We have here three expressions for the same likelihood under different noise assumptions. We stress that the data and model dependent quantities is given by one and the same algorithm which uses just two Kalman filters. All likelihoods can then be computed and compared. This was done for a simulated example in the simulation section. The conclusion is that if the noise variance is constant and approximately known, this value should be used in the first likelihood. If it is constant but very uncertain, then likelihood number two above is the appropiate choice. Finally, if the noise variance is changing, then likelihoods number two or three must be used. The choice depends on if one is interested in detecting these changes as well.

The only drawback with this two filter approach is that it is basically off-line. Nevertheless, it was shown that it could be used on-line as well, with a batchwise processing of the data, with less computational burden than the GLR test.

A Likelihood Expressions for Linear Regressions

In this appendix some different ways of computing likelihoods are given. The measurements are assumed to be generated by a linear regression,

$$y_t = \varphi_t^T \theta + e_t \tag{48}$$

where e_t is white Gaussian noise with variance Λ_t. The parameter estimate is given by the well known RLS scheme,

$$\begin{aligned}
\hat{\theta}_{t+1} &= \hat{\theta}_t + P_t\varphi_t \left(\varphi_t^T P_t \varphi_t + \Lambda_t\right)^{-1} (y_t - \varphi_t^T \hat{\theta}_t) \\
P_{t+1} &= P_t - P_t\varphi_t \left(\varphi_t^T P_t \varphi_t + \Lambda_t\right)^{-1} \varphi_t P_t.
\end{aligned} \tag{49}$$

First on-line and off-line expressions for a Gaussian prior on the parameters θ are given, then the case of non-informative prior is considered.

Lemma A.1 *The likelihood of the sequence $Y_1^N = \{y_k\}_{k=1}^N$ from (48) is computed on-line by*

$$\begin{aligned}
\log p(Y_1^N) &= \log p(Y_1^{N-1}) - \frac{1}{2}\log 2\pi - \frac{1}{2}\log\det\left(\varphi_N^T P_{N-1}\varphi_N + \Lambda_N\right) \\
&\quad -\frac{1}{2}(y_N - \varphi_N^T\hat{\theta}_{N-1})^T \left(\varphi_N^T P_{N-1}\varphi_N + \Lambda_N\right)^{-1} (y_N - \varphi_N^T\hat{\theta}_{N-1}) \\
&= -N\frac{1}{2}\log 2\pi - \sum_{t=1}^{N}\frac{1}{2}\log\det\left(\varphi_t^T P_{t-1}\varphi_t + \Lambda_t\right) \\
&\quad -\frac{1}{2}\sum_{t=1}^{N}(y_t - \varphi_t^T\hat{\theta}_{t-1})^T \left(\varphi_t^T P_{t-1}\varphi_t + \Lambda_t\right)^{-1} (y_t - \varphi_t^T\hat{\theta}_{t-1})
\end{aligned}$$

and off-line by

$$\begin{aligned}
\log p(Y_1^N) &= -N\frac{1}{2}\log 2\pi + \frac{1}{2}\log\det P_N - \frac{1}{2}\log\det P_0 - \frac{1}{2}\sum_{t=1}^{N}\log\det\Lambda_t \\
&\quad - \frac{1}{2}\sum_{t=1}^{N}(y_t - \varphi_t^T\hat{\theta}_N)^T\Lambda_t^{-1}(y_t - \varphi_t^T\hat{\theta}_N) + \frac{1}{2}(\hat{\theta}_N - \theta_0)^T P_0^{-1}(\hat{\theta}_N - \theta_0).
\end{aligned}$$

Here $\hat{\theta}_t$ and P_t are given by the RLS scheme (49) with initial conditions θ_0 and P_0.

Proof: Iteratively using Bayes' law gives

$$p(Y_1^N) = \prod_{t=1}^{N} p(y_t|Y_1^{t-1}).$$

Since $\theta_t|Y_1^{t-1} \in \mathrm{N}(\hat{\theta}_{t-1}, P_{t-1})$, where $\hat{\theta}_{t-1}$ and P_{t-1} is the RLS estimate and covariance matrix respectively, it follows that

$$y_t|Y_1^{t-1} \in \mathrm{N}(\varphi_t^T\hat{\theta}_{t-1}, \varphi_t^T P_{t-1}\varphi_t + \Lambda_t),$$

and the on-line expression follows from the definition of the Gaussian PDF.

The equality

$$\begin{aligned}
&\sum_{t=1}^{N}(y_t-\varphi_t^T\hat{\theta}_{t-1})^T\left(\varphi_t^T P_{t-1}\varphi_t+\Lambda_t\right)^{-1}(y_t-\varphi_t^T\hat{\theta}_{t-1})\\
=\;&\sum_{t=1}^{N}(y_t-\varphi_t^T\hat{\theta}_N)^T\Lambda_t^{-1}(y_t-\varphi_t^T\hat{\theta}_N)+(\hat{\theta}_N-\theta_0)^T P_0^{-1}(\hat{\theta}_N-\theta_0)
\end{aligned}$$

is proved in [6], page 434.

By using the alternative update formula for the covariance matrix, $P_t^{-1} = P_{t-1}^{-1}+\varphi_t\Lambda_t^{-1}\varphi_t^T$ and the equality $\det(I+AB)=\det(I+BA)$ we have

$$\begin{aligned}
&\sum_{t=1}^{N}\log\det\left(\varphi_t^T P_{t-1}\varphi_t+\Lambda_t\right)\\
=\;&\sum_{t=1}^{N}\log\det\Lambda_t+\log\det\left(I+\Lambda_t^{-1}\varphi_t^T P_{t-1}\varphi_t\right)\\
=\;&\sum_{t=1}^{N}\log\det\Lambda_t+\log\det\left(I+\varphi_t\Lambda_t^{-1}\varphi_t^T P_{t-1}\right)\\
=\;&\sum_{t=1}^{N}\log\det\Lambda_t-\log\det P_{t-1}^{-1}+\log\det\left(P_{t-1}^{-1}+\varphi_t\Lambda_t^{-1}\varphi_t^T\right)\\
=\;&\sum_{t=1}^{N}\log\det\Lambda_t-\log\det P_{t-1}^{-1}+\log\det P_t^{-1}\\
=\;&\sum_{t=1}^{N}\log\det\Lambda_t+\log\det P_0-\log\det P_N
\end{aligned}$$

and the off-line expression follows. □

Corollary A.1 (Likelihood with Gaussian prior on the parameters) *If $\theta\in N(\theta_0,P_0)$, the likelihood for the measurements Y_1^N from (48) can be be calculated by*

$$p(Y_1^N)=p(Y_1^N|\hat{\theta}_N)\left(\frac{\det P_N}{\det P_0}\right)^{1/2}\exp(-\frac{1}{2}(\hat{\theta}_N-\theta_0)^T P_0^{-1}(\hat{\theta}_N-\theta_0))$$

Proof: This follows immediately from the off-line expression in Lemma A.1. □

Corrolary A.1 describes a convenient way both to compute the likelihood and to compare the likelihoods $p(Y_1^N)=\int p(Y_1^N|\theta)p(\theta)d\theta$ and $p(Y_1^N|\hat{\theta}_N)=\max p(Y_1^N|\theta)$. The next lemma is the appropriate counterpart when the prior of θ is non-informative.

Lemma A.2 (Likelihood with non-informative prior on the parameters) *Consider the linear regression (48) with a non-informative prior on the parameter vector θ. Then the likelihood for the measurements can be calculated by*

$$p(Y_1^N)=p(Y_1^N|\hat{\theta}_N)\,(2\pi)^{n/2}\,(\det P_N)^{1/2} \tag{50}$$

where $n=\dim\theta$.

Proof: In matrix notation, the linear regression can be written $Y = \Phi\theta$. Then it is well known that if there is no prior on θ, then

$$\theta | Y \in \mathrm{N}\left(\hat{\theta}, P\right)$$

where

$$\begin{aligned}
\hat{\theta} &= \left(\Phi^T \Phi\right)^{-1} \Phi^T Y \\
P &= \left(\Phi^T \Phi\right)^{-1}.
\end{aligned}$$

First we derive the following relationship by completing squares.

$$\begin{aligned}
& (Y - \Phi\theta)^T (Y - \Phi\theta) - (Y - \Phi\hat{\theta})^T (Y - \Phi\hat{\theta}) \\
= \; & Y^T Y - Y^T \Phi\theta - \theta^T \Phi^T Y + \theta^T \Phi^T \Phi\theta \\
& -Y^T Y + 2Y^T \Phi(\Phi^T \Phi)^{-1} \Phi^T Y - Y^T \Phi(\Phi^T \Phi)^{-1} \Phi^T \Phi(\Phi^T \Phi)^{-1} \Phi^T Y \\
= \; & \theta^T \Phi^T \Phi\theta - Y^T \Phi\theta - \theta^T \Phi^T Y + \hat{\theta}^T \Phi^T \Phi\hat{\theta} \\
= \; & (\theta - \hat{\theta})^T \Phi^T \Phi(\theta - \hat{\theta}) + \hat{\theta}^T \Phi^T \Phi\theta + \theta^T \Phi^T \Phi\hat{\theta} - Y^T \Phi\theta - \theta^T \Phi^T Y \\
= \; & (\theta - \hat{\theta})^T \Phi^T \Phi(\theta - \hat{\theta})
\end{aligned}$$

From this equality it follows that

$$\begin{aligned}
p(Y_1^N) &= \int p(Y_1^N | \theta) d\theta \\
&= \int p(Y_1^N | \hat{\theta}) \exp\left(-\frac{1}{2}(\theta - \hat{\theta})^T \Phi^T \Phi(\theta - \hat{\theta})\right) d\theta \\
&= p(Y_1^N | \hat{\theta}) \, (2\pi)^{n/2} \left(\det P_N\right)^{1/2}
\end{aligned}$$

and that completes the proof.

References

[1] B. Anderson and J. Moore. *Optimal Filtering.* Information and System Sciences Series. Prentice Hall, New Jersey, 1979.

[2] M Basseville. Detecting changes in signals and systems — a survey. *Automatica*, 24:309–326, 1988.

[3] M. Basseville and A. Benveniste. Design and comparative study of some sequential jump detection algorithms for digital signals. *IEEE Transactions on Aucoustics, Speech and Signal Processing*, 31:521–535, 1983.

[4] M. Basseville, A. Benveniste, G. Moustakides, and A. Rougeé. Detection and diagnosis of changes in the eigenstructure of nonstationary multivariable systems. *Automatica*, 23:479–489, 1987.

[5] A. Benveniste, M. Basseville, and B.V. Moustakides. The asymptotic local approach to change detection and model validation. *IEEE Transactions on Automatic Control*, 32:583–592, 1987.

[6] L. Ljung and T Söderström. *Theory and Practice of Recursive Identification.* MIT Press, Cambridge MA, 1983.

[7] M. Niedzwiecki. Identification of time-varying systems with abrupt parameter changes. In *9th IFAC/IFORS Symposium on Identification and System Parameter Identification*, volume II, pages 1718–1723. IFAC, 1991.

[8] G. Verghese and T. Kailath. A further note on backwards markovian models. *IEEE Trans. Inform. Theory*, 25:121–124, 1979.

[9] A. Willsky. *Detection of Abrupt Changes in Dynamic Systems. In "Detection of Abrupt Changes in Signals and Dynamical Systems".* Lecture Notes in Control and Information Sciences. Springer-Verlag, 1986.

[10] A.S. Willsky and H.L. Jones. A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. *IEEE Transactions on Automatic Control*, pages 108–112, 1976.