THE MARGINALIZED LIKELIHOOD RATIO TEST FOR DETECTING ABRUPT CHANGES

Fredrik Gustafsson

Department of Electrical Engineering, Linköping University, S-58183, Sweden

Submitted to IEEE Transactions on Automatic Control

Revised 950307

ABSTRACT

The *generalized likelihood ratio* (GLR) test is a widely used method for detecting abrupt changes in linear systems and signals. In this paper the *marginalized likelihood ratio* (MLR) test is introduced for eliminating three shortcomings of GLR, while preserving its applicability and generality. Firstly, the need for a user-chosen threshold is eliminated in MLR. Secondly, the noise levels need not be known exactly and may even change over time, which means that MLR is robust. Finally, a very efficient exact implementation with linear in time complexity for batch-wise data processing is developed. This should be compared to the quadratic in time complexity of the exact GLR.

1 Introduction

The problem of detecting abrupt changes in linear systems and signals occurs in many applications. The practical and theoretical interest in this field are reflected in a large number of surveys, for instance [2, 3, 4, 6, 10, 11, 14]. One of the most powerful methods in change detection is the *generalized likelihood ratio* (GLR) test proposed in [15]. It applies to cases of abrupt changes in the state of an arbitrary linear system with known dynamics. Its general applicability has contributed to that GLR is now a standard tool in change detection. As summarized in [7], GLR has an appealing analytic framework, is widely understood by many researchers and is readily applicable to systems already utilizing a Kalman filter. Another advantage with GLR is that it partially solves the *diagnosis* problem in fault detection, that is, to locate the physical cause of the change. In [7], a number of drawbacks with GLR is pointed out as well. Among these, we mention problems with choosing decision thresholds and untenable computational burden.

The use of likelihood ratios in hypothesis testing is motivated by the Neyman–Pearson Lemma, see for instance Theorem 3.1 in [8]. In the application considered here, it says that the likelihood ratio is the optimal test statistic when the change magnitude is known and just one change time is considered. This is not the case here, but a sub-optimal extension is immediate: The test is computed for each possible change time, or a restriction to a sliding window, and if several tests indicate a change the most significant is taken as the estimated change time. In GLR, the actual change in the state of a linear system is estimated from data and then used in the likelihood ratio.

In this contribution, we propose to consider the change magnitude as a stochastic nuisance parameter. This is then eliminated not by estimation but by *marginalization*. Marginalization is well-known in estimation theory and is also used other detection problems, see for instance [13], though not yet in change detection. The resulting test will be denoted the *marginalized likelihood ratio* (MLR) test. The MLR test applies to all cases where GLR does, but we will point out three advantages with using the former:

- Tuning. Unlike GLR, there is no sensitive threshold to choose in MLR. One interpretation is that a reasonable threshold in GLR is chosen automatically.
- Robustness to modeling errors. The performance of GLR deteriorates in the case of incorrectly chosen noise variances. The noise level is in MLR allowed to be considered as another unknown nuisance parameter. This approach increases the robustness of MLR.
- Complexity. GLR requires a linearly increasing number of parallel filters. An approximation involving a sliding window technique is proposed in [15] to

obtain a constant number of filters, typically equivalent to 10–20 parallel filters. For off-line processing, the MLR test can be computed exactly from only two filters. This implementation is of particularly great impact on the design step, where the false alarm rate, robustness properties and detectability of different changes are evaluated using Monte-Carlo simulations. In fact, already the computation of one single exact GLR test for a realistic data size (> 1000) is infeasible.

It should be noted already here, that in a maximum likelihood framework, MLR provides the optimal estimate of the change time.

The problem formulation and notation is presented in Section 2. The GLR test is summarized in Section 3, while a detailed derivation using linear regression terminology is found in Appendix B. The same linear regression approach is used in Appendix C to derive the MLR. A direct on-line implementation of MLR and a comparison to GLR are given in Section 4. The two-filter off-line implementation is derived in Section 5, while the case on unknown scalings in the noise variances is treated in Section 6. Section 7 contains Monte-Carlo simulations for a tracking example and Section 8 summarizes the paper.

2 Problem formulation

2.1 Model

The model is assumed to be a linear state space model, where the abrupt change occurs in the state vector:

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

The observations are denoted y_t, the input u_t and the state x_t. For simplicity, the input term $G_t u_t$ will not be written out explicitly in the sequel. Here w_t, e_t and x_0 are assumed to be independent Gaussian variables;

$$\begin{aligned} w_t &\in \mathrm{N}(0, Q_t) \\ e_t &\in \mathrm{N}(0, R_t) \\ x_0 &\in \mathrm{N}(0, \Pi_0). \end{aligned}$$

Furthermore, they are assumed to be mutually independent. The state jump ν occurs at the unknown time instant k, and $\delta(j)$ is the pulse function that is one if

$j = 0$ and zero otherwise. The set of measurements $y_1, y_2, .., y_N$, each of dimension p, will be denoted y^N and y_t^N denotes the set $y_t, y_{t+1}, .., y_N$.

The Kalman filter equations for a jump $\nu \in \mathrm{N}(0, P_\nu)$ at a given time k are

$$\begin{aligned}
\hat{x}_{1|0}(k) &= x_0 \\
P_{1|0}(k) &= P_0 \\
\varepsilon_t(k) &= y_t - H_t \hat{x}_{t|t-1}(k) \\
S_t(k) &= H_t P_{t|t-1}(k) H_t^T + R_t \\
K_t(k) &= P_{t|t-1}(k) H_t^T S_t^{-1}(k) \\
\hat{x}_{t+1|t}(k) &= F_t \hat{x}_{t|t-1}(k) + F_t K_t(k) \varepsilon_t(k) \\
P_{t+1|t}(k) &= F_t \left(P_{t|t-1}(k) - K_t(k) H_t P_{t|t-1}(k) \right) F_t^T + Q_t + \delta(t-k) P_\nu .
\end{aligned} \tag{2}$$

The addressed problem is to modify these equations to the case where k and ν are unknown. The jump instant k is of primary interest, but also good state estimates may be desired.

In GLR, ν is an unknown constant, while it is considered as a stochastic variable in the MLR test. To start with, the jump will be assumed to have a Gaussian prior. Later on, a non-informative prior will be used that is sometimes called a prior of ignorance, see [8]. This prior is characterized by a constant density function, $p(\nu) = C$.

Example 1 *We can use Eq. (1) to detect abrupt changes in the mean of a sequence of stochastic variables by letting* $F_t = 1, H_t = 1, Q_t = 0, G_t = 0$. *Furthermore, if the mean before the change is supposed to be 0, a case often considered in literature (see [4]), we have* $x_0 = 0$ *and* $\Pi_0 = 0$.

Example 2 *An important special case of Eq. (1) arises if we put* $F_t = I$ *and* $H_t = (y_{t-1}, y_{t-2}, .., u_{t-1}, u_{t-2}, ..)$. *We then have a linear regression description of what is commonly called an ARX (Auto-Regressive Exogenous input) model, where* x_t *is the (time-varying) parameter vector and* H_t *the regressors. In this way, we can detect abrupt changes in the transfer function of ARX models. Note that the change occurs in the* dynamics *of the system in this case, and not in the system's* state.

2.2 Likelihood

The likelihood for the measurements up to time N given the jump ν at time k is denoted $p(y^N | k, \nu)$. The same notation is used for the conditional density function for y^N, given k, ν. For simplicity, $k = N$ is agreed to mean no jump. There are two principally different possibilities to estimate the jump time k.

- Joint ML estimate of k and ν,

$$\widehat{(k,\nu)} = \arg\max_{k\in[1,N],\nu} p(y^N|k,\nu). \tag{3}$$

  Here $\arg\max_{k\in[1,N],\nu} p(y^N|k,\nu)$ means the maximizing arguments of $p(y^N|k,\nu)$ where k is restricted to $[1,N]$.

- The ML estimate of just k using marginalization of the conditional density function $p(y^N|k,\nu)$;

$$\begin{aligned} p(y^N|k) &= \int p(y^N|k,\nu)p(\nu)d\nu && (4)\\ \hat{k} &= \arg\max_{k\in[1,N]} p(y^N|k). && (5)\end{aligned}$$

The likelihood for data given just k in (4) is the starting point in this approach.

A tool in the derivations is the so called flat prior, of the form $p(\nu) = C$, which is not a proper density function. The idea of using a flat prior, or non-informative prior, in marginalization is perhaps best explained by an example.

Example 3 *Suppose we have t observations from a Gaussian distribution; $y_t \in N(\mu,\lambda)$. Thus the likelihood $p(y^t|\mu,\lambda)$ is Gaussian. We want to compute the likelihood conditioned on just λ using marginalization. That is, $p(y^t|\lambda) = \int p(y^t|\mu,\lambda)p(\mu)d\mu$. Two alternatives of prior are a Gaussian, $\mu \in N(\mu_0, P_0)$, and a flat prior, $p(\mu) = C$. In both cases, we end up with an inverse Wishart density function, which will be defined in (25), with maximas*

$$\begin{aligned} \mu \in \mathrm{N}(\mu_0, P_0) \quad &\Rightarrow \quad \hat{\lambda} = \frac{1}{N-1}\left(\sum_{k=1}^{t}(y_t-\bar{y})^2 + \frac{(\mu_0-\bar{y})^2}{P_0}\right)\\ p(\mu) = C \quad &\Rightarrow \quad \hat{\lambda} = \frac{1}{N-1}\sum_{k=1}^{t}(y_t-\bar{y})^2 \end{aligned}$$

where $\bar{y}$ is the sample average.

Thus, a flat prior eliminates the bias induced by the prior. We remark that the likelihood interpreted as a conditional density function is proper and it does not depend on the constant C.

2.3 Likelihood ratio

In the context of hypothesis testing, the likelihood ratios rather than the likelihoods are used. The Likelihood Ratio (LR) test is a multiple hypotheses test, where the different jump hypotheses are compared to the no jump hypothesis pairwise. In the LR test, the jump magnitude is assumed to be known. The hypotheses under consideration are

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

The test is as follows. Introduce the log likelihood ratio for the hypotheses as the test statistic:

$$l_N(k,\nu) \triangleq 2\log\frac{p(y^N|H_1(k,\nu))}{p(y^N|H_0)} = 2\log\frac{p(y^N|k,\nu)}{p(y^N|k=N)} \tag{6}$$

The factor 2 is just for notational convenience. We use the convention that $H_1(N,\nu) = H_0$, so again $k = N$ means no jump. Then the LR estimate can be expressed as

$$\hat{k}^{ML} = \arg\max_k l_N(k,\nu), \tag{7}$$

when ν is known. Exactly as in (3) and (5) we have two possibilities how to eliminate the unknown nuisance parameter ν. Double maximization gives the GLR test, proposed for change detection in [15], and marginalization the MLR test suggested in this paper.

3 The GLR test

Since the GLR and MLR tests are so closely related, we start with giving the GLR test. Starting with the likelihood ratio in (6), the GLR test is a double maximization over k and ν,

$$\begin{aligned} \hat{\nu}(k) &= \arg\max_\nu 2\log\frac{p(y^N|k,\nu)}{p(y^N|k=N)} \\ \hat{k} &= \arg\max_k 2\log\frac{p(y^N|k,\hat{\nu}(k))}{p(y^N|N)}. \end{aligned}$$

where $\hat{\nu}(k)$ is the maximum likelihood estimate of ν, given a jump at time k. The jump candidate $\hat{k}$ in the GLR test is accepted if

$$l_N(\hat{k},\hat{\nu}(\hat{k})) > h. \tag{8}$$

The threshold h characterizes a hypothesis test and distinguishes the GLR test from the ML method (3). Note that (3) is a special case of (8), where $h = 0$. If the zero-jump hypothesis is rejected, the state estimate can easily be compensated for the detected jump.

The idea in the implementation of GLR in [15] is to make the dependence on ν explicit. This task is solved in Appendix B. The key point is that the innovations from the Kalman filter (2) with $k = N$ can be expressed as a linear regression in ν,

$$\varepsilon_t(k) = \varphi_t^T(k)\nu + \varepsilon_t$$

where $\varepsilon_t(k)$ are the innovations from the Kalman filter if ν and k were known. The GLR algorithm can be implemented as follows.

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

- *Calculate the innovations from the Kalman filter (2) assuming no jump.*
- *Compute the regressors $\varphi_t(k)$ using (36) and the linear regression quantities $R_t(k) = \sum_{i=1}^t \varphi_i(k) S_i^{-1} \varphi_i^T(k)$ and $f_t(k) = \sum_{i=1}^t \varphi_i(k) S_i^{-1} \varepsilon_i$ for each k, $1 \le k \le t$.*
- *At time $t = N$, the test statistic is given by $l_N(k, \hat{\nu}(k)) = f_N^T(k) R_N^{-1}(k) f_N(k)$.*
- *A jump candidate is given by $\hat{k} = \arg\max l_N(k, \hat{\nu}(k))$. It is accepted if $l_N(\hat{k}, \hat{\nu}(\hat{k}))$ is greater than some threshold h (otherwise $\hat{k} = N$) and the corresponding estimate of the jump magnitude is given by $\hat{\nu}_N(\hat{k}) = R_N^{-1}(\hat{k}) f_N(\hat{k})$.*

We make some remarks on the algorithm.

- It can be shown that the test statistic $l_N(k, \nu(k))$ under the null hypothesis is χ^2 distributed. Thus, given the confidence level on the test, the threshold h can be found from standard statistical tables. Note that this is a multiple hypothesis test performed for each $k = 1, 2, .., N-1$, so nothing can be said about the total confidence level.
- The regressor $\varphi_t(k)$ is called a *failure signature matrix* in [15].
- The regressors are pre-computable. Furthermore, if the system and the Kalman filter are time invariant, the regressor is only a function of $t-k$, which simplifies the calculations.
- The formulation in Algorithm 1 is off-line. Since the test statistic involves a matrix inversion of R_N, a more efficient on-line method is as follows. From (42) and (45) we get
$$l_t(k, \hat{\nu}(k)) = f_t^T(k)\hat{\nu}_t(k),$$
where t is used as time index instead of N. The Recursive Least Squares (RLS) scheme, see (27), can now be used to update $\hat{\nu}_t(k)$ recursively, eliminating the matrix inversion of $R_t(k)$. Thus, the best implementation requires t parallel RLS schemes and one Kalman filter.

The choice of threshold is difficult. It depends not only on the system's SNR but also on the actual noise levels as will be pointed out in Section 6.

4 The MLR test

In Appendix C the MLR test is derived using the quantities from the GLR test in Algorithm 1. This implementation does not support the ideas of this paper, but it gives a nice relation between GLR and MLR. In fact, they coincide for a certain choice of threshold.

Lemma 1 *If (1) is time invariant and ν is unknown, then the GLR test in Algorithm 1 gives the same estimated jump time as the MLR test in Theorem 5 as $N-k \to \infty$ and $k \to \infty$ if the threshold is chosen as*

$$h = p \log(2\pi) - \log \det \bar{R}_N(k)$$

where $\bar{R}_N(k) = \lim_{N-k\to\infty, k\to\infty} R_N(k)$, and $R_N(k)$ is defined in Algorithm 1.

Proof: In the MLR test a jump k is detected if $l_N(k) > l_N(N) = 0$ and in the GLR if $l_N(k, \nu(k)) > h$. From Theorem 5 we have $l_N(k) = l_N(k, \nu(k)) - \log \det R_N(k) + p \log(2\pi)$. Lemma 6 shows that $R_N(k)$ converges as $N \to \infty$, and so does $\log \det R_N(k)$. Since (1) is restricted to be time invariant the terms of $R_N(k)$ that depend on the system matrices and the Kalman gain are the same independently of k as $k \to \infty$ according to (36). $\Box$

The threshold is thus automatically included in the MLR test. If we want MLR to mimic a GLR test, we can of course include an external threshold $h_{MLR} = h_{GLR} - p\log(2\pi) - \log\det \bar{R}_N(k)$. In that case, we accept the jump hypothesis only if $l_N(k) > h_{MLR}$. The external threshold can also be included in an *ad-hoc* manner to tune the false alarm rate versus probability of correct detection.

Now, we will make a new derivation of the MLR test in a direct way using a linearly increasing number of Kalman filters. This derivation enables firstly the efficient implementation in the Section 5 and secondly the elimination of noise scalings in Section 6. Since the magnitudes of the likelihoods turn out to be of completely different orders, the log likelihood will be used in order to avoid possible numerical problems.

Theorem 1 *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 in (2). The log likelihood, conditioned on a jump at time k, can be recursively computed by*

$$\log p(y^t|k) = \log p(y^{t-1}|k) - \frac{p}{2}\log 2\pi - \frac{1}{2}\log\det S_t(k) - \frac{1}{2}\varepsilon_t^T(k)S_t^{-1}(k)\varepsilon_t(k). \quad (9)$$

Proof: By Bayes' rule we have

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

It is a well-known property of the Kalman filter, see for instance [1], that

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

and the result follows from the definition of the Gaussian density function. □

This approach requires a linearly growing number with N of Kalman filters.

5 A two-filter implementation

In order to compute the likelihood ratios efficiently, two statistical tricks are needed:

- Use a flat prior on the jump magnitude ν.
- Use some of the last observations for calculating proper distributions.

The point with the first one is that the measurements after the jump are independent of the measurements before the jump, and the likelihood can be computed as a product of the likelihoods before and after the jump. However, this leads to a problem. The likelihood is not uniquely defined immediately after a jump of infinite variance. Therefore, a small part of the data is used for initialization. We also have to assume that F_t in (1) is invertible.

5.1 Reversed time model

First, a Kalman filter running backwards in time is needed. This is a well-known problem in the context of fixed-interval smoothing. One solution here, which suits our purposes, is to use the reversed time version of the linear model (1) and then just apply the usual Kalman filter to this model. This will give the conditional mean $\hat{x}^B_{t|t+1} = \mathrm{E}\ (x_t|y^N_{t+1})$ and the conditional covariance $P^B_{t|t+1} = \mathrm{Cov}\ (x_t|y^N_{t+1})$ of the Gaussian variable x_t, given the measurements y^N_{t+1}.

The following lemma gives the desired backwards Markovian model that is sample path equivalent to (1). The Markov property implies that the noise process $\{w_t\}$ and the final value of the state vector x_N are independent. Not only are the first and second order statistics equal for these two models, but they are indeed sample path equivalent, since they both produce the same state and output vectors.

Lemma 2 *The following model is sample path equivalent to (1) and is its corresponding backward Markovian model*

$$\begin{aligned} x_t &= F^B_t x_{t+1} + w^B_{t+1} \\ y_t &= H_t x_t + e_t. \end{aligned} \tag{10}$$

Here

$$F^B_t = \Pi_t F^T_t \Pi^{-1}_{t+1} = F^{-1}_t - F^{-1}_t Q_t \Pi^{-1}_{t+1} \tag{11}$$
$$Q^B_t = \Pi_t - \Pi_t F^T_t \Pi^{-1}_{t+1} F_t \Pi_t = F^{-1}_t Q_t F^{-T}_t - F^{-1}_t Q_t \Pi^{-1}_{t+1} Q_t F^{-T}_t \tag{12}$$

where $\Pi_t = E[x_t x^T_t]$ *is the* a priori *covariance matrix of the state vector, computed recursively by* $\Pi_{t+1} = F_t \Pi_t F^T_t + Q_t$*. The last equalities of (11) and (12) hold if* F_t *is invertible.*

Proof: See [12]. □

Thus, the "*a posteriori*" distribution for the state vector is

$$x_t|y^N_{t+1} \in \mathrm{N}(\hat{x}^B_{t|t+1}, P^B_{t|t+1}).$$

The problem here, which is not apparent in smoothing, is that the prior $\Pi_N = \mathrm{E}[x_N x^T_N]$ generally depends on k, so we must be careful in using a common Kalman filter for all hypotheses. For this reason, the assumption on infinite variance of the jump magnitude is needed, so Π_N is infinite for all k as well. By infinite is meant that $\Pi^{-1}_N = 0$. The recursion $\Pi_{t+1} = F\Pi_t F^T + Q$ gives $\Pi^{-1}_{k+1} = 0$. Hence, (10) becomes

$$\begin{aligned} x_t &= F^{-1}_t x_{t+1} + F^{-1}_t w_t = F^{-1}_t x_{t+1} + w^B_t \\ y_t &= H_t x_t + e_t. \end{aligned} \tag{13}$$

Here $Q^B_t = \mathrm{E}[w^B_t (w^B_t)^T] = F^{-1}_t Q_t F^{-T}_t$ and $\Pi^{-1}_N = 0$ where $\Pi_N = \mathrm{E}[x_N x^T_N]$.

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 $P^B_{t|t+1}$.

5.2 The two-filter implementation

In this and the next section the likelihoods rather than likelihood ratios will be derived. The last L measurements are used for normalization, which means that jumps after time $N-L$ are not considered. This is not a serious restriction, since it suffices to choose $L = \dim x$, and jumps supported by so few data can not be detected with any significance anyway. We are now ready for the main result of this section.

Theorem 2 *Consider the signal model (1) for the case of an invertible F_t. The likelihood for the measurements conditioned on a jump at time k and the last L measurements, can be computed by two Kalman filters as follows. First, the likelihoods are separated,*

$$p(y^{N-L}|y^N_{N-L+1},k) = \begin{cases} \frac{p(y^N)}{p(y^N_{N-L+1})} & \text{if } k = N \\ p(y^k)p(y^{N-L}_{k+1}|y^N_{N-L+1},k) & \text{if } k < N-L \end{cases} \tag{14}$$

The involved likelihoods are computed by

$$p(y^k) = \prod_{t=1}^{k} \gamma(y_t - H_t\hat{x}^F_{t|t-1}, H_tP^F_{t|t-1}H_t^T + R_t) \tag{15}$$

$$p(y^N_{N-L+1}) = \prod_{t=N-L+1}^{N} \gamma(y_t - H_t\hat{x}^N_{t|t-1}, H_tP^N_{t|t-1}H_t^T + R_t) \tag{16}$$

$$p(y^{k+1}|y^N_{N-L+1},k) = \prod_{t=k+1}^{N-L} \gamma(y_t - H_t\hat{x}^B_{t|t+1}, H_tP^B_{t|t+1}H_t^T + R_t) \tag{17}$$

Here $\gamma(x-\mu,P)$ is the Gaussian probability density function. The quantities $\hat{x}^F_{t|t-1}$ and $P^F_{t|t-1}$ are given by the Kalman filter applied to the forward model (1) and $\hat{x}^B_{t|t+1}$ and $P^B_{t|t+1}$ are given by the Kalman filter applied on the backward model (13). The quantities $\hat{x}^N_{t|t-1}$ and $P^N_{t|t-1}$ used for normalization are given by the Kalman filter applied on the forward model initiated at time $t = N-L+1$ with $P_{N-L+1|N-L} = \Pi_{N-L+1}$.

Proof: Bayes' law gives

$$\begin{aligned} p(y^{N-L}|y^N_{N-L+1},k=N) &= \frac{p(y^{N-L}\cap y^N_{N-L+1}|k=N)}{p(y^N_{N-L+1}|k=N)} && (18)\\ &= \frac{p(y^N|k=N)}{p(y^N_{N-L+1}|k=N)} && (19)\\ p(y^{N-L}|y^N_{N-L+1},k<N-L) &= p(y^k|k<N-L)p(y^{N-L}_{k+1}|y^k,y^N_{N-L+1},k) && (20)\\ &= p(y^k)p(y^{N-L}_{k+1}|y^N_{N-L+1},k) && (21) \end{aligned}$$

In the last equality it is used that the jump at time k does not affect the measurements before time k by causality, so $p(y^k|k) = p(y^k)$, and that the infinite variance jump makes the measurement after the jump independent of those before.

The likelihood for a set y_m^n can be expanded either forwards or backwards using Bayes' chain rule:

$$p(y_m^n) = \prod_{t=m}^{n} p(y_t|y_m^{t-1}) \tag{22}$$

$$p(y_m^n) = \prod_{t=m}^{n} p(y_t|y_{t+1}^{n}). \tag{23}$$

Now $p(y^N|k = N)$ and $p(y^k)$ are computed using the forward recursion (22) and since x_t is Gaussian it follows immediately that $y_t|y^{t-1}$ is Gaussian with mean $H_t\hat{x}^F_{t|t-1}$ and covariance $H_tP^F_{t|t-1}H_t^T + R_t$ and (15) follows. Also $p(y^N_{N-L+1}|k = N)$ is computed in the same way, the difference is that the Kalman filter is initiated at time $N - L + 1$. Finally, $p(y^{N-L}_{k+1}|y^N_{N-L+1}, k)$ is computed using (23) where $y_t|y^N_{t+1}$ is Gaussian with mean $H_t\hat{x}^B_{t|t+1}$ and covariance $H_tP^B_{t|t+1}H_t^T + R_t$ and (17) follows. $\square$

As seen, all that is needed to compute the likelihoods are one Kalman filter running backwards in time, one running forwards in time and one processing the normalizing data at the end. 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. The notation introduced here will be used in the sequel of the paper.

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

Forward filter *for $t = 1, 1, .., N$:*

$$\begin{aligned}
\hat{x}^F_{1|0} &= x_0 \\
P^F_{1|0} &= \Pi_0 \\
V^F(0) &= 0 \\
D^F(0) &= 0 \\
\varepsilon^F_t &= y_t - H_t\hat{x}^F_{t|t-1} \\
S^F_t &= H_tP^F_{t|t-1}H_t + R_t \\
\hat{x}^F_{t+1|t} &= F_t\hat{x}^F_{t|t-1} + F_tP^F_{t|t-1}H_t(S^F_t)^{-1}\varepsilon^F_t \\
P^F_{t+1|t} &= F_t\left(P^F_{t|t-1} - P^F_{t|t-1}H_t^T(S^F_t)^{-1}H_tP^F_{t|t-1}\right)F_t^T + Q_t \\
\Pi_{t+1} &= F\Pi_tF^T + Q \\
V^F(t) &= V^F(t-1) + (\varepsilon^F_t)^T(S^F_t)^{-1}\varepsilon^F_t \\
D^F(t) &= D^F(t-1) + \log\det S^F_t
\end{aligned}$$

Normalization filter *for* $t = N-L+1, N-L+2, .., N$*:*

$$
\begin{aligned}
\hat{x}^N_{N-L+1|N-L} &= (\prod_{t=1}^{N-L} F_t)x_0 \\
P^N_{N-L+1|N-L} &= \Pi_{N-L} \\
V^N(N-L) &= 0 \\
D^N(N-L) &= 0 \\
\varepsilon^N_t &= y_t - H_t\hat{x}^N_{t|t-1} \\
S^N_t &= H_t P^N_{t|t-1} H_t + R_t \\
\hat{x}^N_{t+1|t} &= F_t\hat{x}^N_{t|t-1} + F_t P^N_{t|t-1} H_t (S^N_t)^{-1}\varepsilon^N_t \\
P^N_{t+1|t} &= F_t \left(P^N_{t|t-1} - P^N_{t|t-1} H_t^T (S^N_t)^{-1} H_t P^N_{t|t-1}\right) F_t^T + Q_t \\
V^N(t) &= V^N(t-1) + (\varepsilon^N_t)^T (S^N_t)^{-1}\varepsilon^N_t \\
D^N(t) &= D^N(t-1) + \log\det(S^N_t)
\end{aligned}
$$

Backward information filter *for* $t = N, N-1, .., N-L+1$*:*

$$
\begin{aligned}
\hat{a}^B_{N|N+1} &= 0 \\
(P^B_{N|N+1})^{-1} &= 0 \\
\hat{a}^B_{t|t} &= \hat{a}^B_{t|t-1} + H_t^T R_t^{-1} y_t \\
(P^B_{t|t})^{-1} &= (P^B_{t|t-1})^{-1} + H_t^T R_t^{-1} H_t \\
A &= F_t^T (P^B_{t|t})^{-1} F_t \\
B &= A F_t^{-1} G_t \left(G_t^T F_t^{-T} A F_t^{-1} G_t + Q_t^{-1}\right)^{-1} \\
\hat{a}^B_{t-1|t} &= \left(I - B G_t^T F_t^{-T}\right) F_t^T \hat{a}^B_{t|t} \\
(P^B_{t-1|t})^{-1} &= \left(I - B G_t^T F_t^{-T}\right) A
\end{aligned}
$$

Backward Kalman filter *for* $t = N-L, N-L-1, .., 1$*:*

$$
\begin{aligned}
P^B_{N-L|N-L+1} & \quad \text{from backward information filter} \\
\hat{x}^B_{N-L|N-L+1} &= P^B_{N-L|N-L+1}\hat{a}^B_{N-L|N-L+1} \\
V^B(N-L+1) &= 0 \\
D^B(N-L+1) &= 0 \\
\varepsilon^B_t &= y_t - H_t\hat{x}^B_{t|t+1} \\
S^B_t &= H_t P^B_{t|t+1} H_t + R_t \\
\hat{x}^B_{t-1|t} &= F_t^{-1}\hat{x}^B_{t|t+1} + F_t^{-1} P^B_{t|t+1} H_t (S^B_t)^{-1}\varepsilon^B_t \\
P^B_{t-1|t} &= F_t^{-1}\left(P^B_{t|t+1} - P^B_{t|t+1} H_t^T (S^B_t)^{-1} H_t P^B_{t|t+1}\right) F_t^{-T} + F_t^{-1} Q_t F_t^{-T} \\
V^B(t) &= V^B(t+1) + (\varepsilon^B_t)^T (S^B_t)^{-1}\varepsilon^B_t \\
D^B(t) &= D^B(t+1) + \log\det S^B_t
\end{aligned}
$$

Compute the likelihood ratios

$$
\begin{aligned}
l_N(k) &= V^F(N) - V^N(N) - V^F(k) - V^B(k+1) + \\
&\quad D^F(N) - D^N(N) - D^F(k) - D^B(k+1)
\end{aligned}
$$

We make some remarks on the algorithm.

- The normalization filter and the backward information filter play a minor role for the likelihood and might be omitted in an approximate algorithm.
- The jump magnitude, conditioned on the jump time, can be estimated from available information using the signal model (1) and the filter state estimates:

$$
\hat{\nu}(k) = \hat{x}^B(k+1|k+1) - F_k \hat{x}^F(k|k) - G_k u_k
$$

- The *a posteriori* probability of k is easily computed by using Bayes' law. Assuming $p(k) = C$,

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

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

- The relation to fixed-interval smoothing is as follows. The smoothed estimates under the no jump hypothesis can be computed by

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

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

- If the data are collected in batches, the two-filter algorithm can be applied after each batch saving computation time.

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 (14).

A related two-filter idea is found in [9], where a sub-optimal two-filter detection algorithm is proposed for detecting changes in the parameters of a finite impulse response model.

6 Marginalization of the noise level

6.1 Introduction

Knowledge of the covariance matrices is crucial for the performance of model based detectors. The amount of prior information can be substantially relaxed, by eliminating unknown scalings from the covariance matrices in the state space model (1).

$$\overline{R} = \lambda R, \ \ \overline{P}_0 = \lambda P_0, \ \ \overline{Q} = \lambda Q. \tag{24}$$

Here the matrices without bar are chosen by the user, and the ones with bar are the "true" ones or at least give good performance. This means that one chooses the tracking ability of the filter, that is known to be insensitive to scalings, see [1]. The estimator then estimates the actual level, which is decisive for the likelihoods. The assumption (24) implies $\overline{P}_t = \lambda P_t$ and $\overline{S}_t = \lambda S_t$ and from (44) it follows that

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

For the GLR test, 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 in GLR. In this sense, the problem formulation is over-parameterized, since both λ and the threshold have to be chosen by the user.

Equation (24) is an interesting assumption from the practical point of view. Scaling does not influence the filtered state estimates. It is relatively easy for the user to tune the tracking ability, but the sizes of the covariances are harder to judge. The robustness to unknown scalings is one of the advantages with the MLR test, as will be shown.

Another point is that a changing measurement noise variance is known to cause problems to many proposed detection algorithms. This is treated here by allowing the noise covariance scaling to be abruptly changing.

A summary of the MLR's is given in Section 6.4.

6.2 State jump

In this section, the two filter detector in Theorem 2 will be derived for the unknown scaling assumption in (24). If F is not invertible as assumed in Theorem 2, the direct implementation of MLR in Theorem 1 can also be modified in the same way. The following Theorem is the counterpart to the two filter detection method in Algorithm 2, for the case of an unknown λ.

Theorem 3 *Consider the signal model (1) in the case of an unknown noise variance and suppose that (24) holds. With notation as in the two filter detector in Algorithm 2, the log likelihood ratios are given by*

$$
\begin{aligned}
l_N(k) &= (Np-2)\left(\log(V^F(N)-V^N(N)) - \log(V^F(k)+V^B(k+1))\right) \\
&\quad + D^F(N) - D^N(N) - D^F(k) - D^B(k+1)
\end{aligned}
$$

with a flat prior on λ.

Proof: The flat prior on λ here means $p(\lambda|y_{N-L+1}^N) = C$, corresponding to being completely unknown. We get by marginalization if $k < N-L$

$$
\begin{aligned}
& p(y^{N-L}|y_{N-L+1}^N, k) \\
=& \int_0^\infty p(y^{N-L}|y_{N-L+1}^N, k, \lambda) p(\lambda|y_{N-L+1}^N) d\lambda \\
=& C\int_0^\infty p(y^k|\lambda) p(y_{k+1}^{N-L}|y_{N-L+1}^N, k, \lambda) d\lambda \\
=& C\int_0^\infty (2\pi)^{-Np/2} \exp\left(-\frac{1}{2}(D^F(k)+D^B(k))\right) \lambda^{-Np/2} \exp\left(-\frac{V^F(k)+V^B(k)}{2\lambda}\right) d\lambda \\
=& C(2\pi)^{-Np/2} \exp\left(-\frac{1}{2}(D^F(k)+D^B(k))\right) 2^{\frac{Np-2}{2}} \Gamma\left(\frac{Np-2}{2}\right) (V^F(k)+V^B(k))^{-\frac{Np-2}{2}} \\
& \times \int_0^\infty \frac{(V^F(k)+V^B(k))^{\frac{Np-2}{2}}}{2^{(Np-2)/2}\Gamma\left(\frac{Np-2}{2}\right)} \frac{\exp\left(-\frac{V^F(k)+V^B(k)}{2\lambda}\right)}{\lambda^{(Np-2+2)/2}} d\lambda \\
=& C(2\pi)^{-Np/2} \exp\left(-\frac{1}{2}(D^F(k)+D^B(k))\right) 2^{\frac{Np-2}{2}} \Gamma\left(\frac{Np-2}{2}\right) (V^F(k)+V^B(k))^{-\frac{Np-2}{2}}.
\end{aligned}
$$

The gamma function is defined by $\Gamma(a) = \int_0^\infty e^{-t} t^{a-1} dt$. The last equality follows by recognizing the integrand as a density function, namely the inverse Wishart distribution

$$
f(\lambda) = \frac{\sigma^{m/2}}{2^{m/2}\Gamma(m/2)} \frac{\exp\left(-\frac{\sigma}{2\lambda}\right)}{\lambda^{(m+2)/2}} \tag{25}
$$

which integrates to one. In the same way, we have for $k = N$

$$
\begin{gathered}
p(y^{N-L}|y_{N-L+1}^N, k=N) = \\
C(2\pi)^{-Np/2} \exp\left(-\tfrac{1}{2}(D^F(N)-D^N(N))\right) 2^{\frac{Np-2}{2}} \Gamma\left(\tfrac{Np-2}{2}\right) (V^F(N)-V^N(N))^{-\frac{Np-2}{2}}
\end{gathered}
$$

and the result follows. □

Here we remark that the *a posteriori* distribution for λ, given the jump instant k, is $W^{-1}(Np, V^F(k)+V^B(k))$, where W^{-1} denotes the inverse Wishart distribution (25).

6.3 State and variance jump

In this section, the likelihood is given for the case when the noise variance is different before and after the jump. This result is of great practical relevance, since variance changes are very common in real signals.

Theorem 4 *Consider the same detection problem as in Theorem 3 but with a noise variance changing at the jump instant,*

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

With notation as in Algorithm 2, the log likelihood ratios for $2/p < k < N - 2/p$ are given by

$$\begin{aligned} l_N(k) &= (Np-2)\log(V^F(N) - V^N(N)) - (kp-2)\log V^F(k) \\ &\quad -(Np-kp-2)\log V^B(k+1)) + D^F(N) - D^N(N) - D^F(k) - D^B(k+1) \end{aligned}$$

if the prior on λ is flat.

Proof: The proof resembles that of Theorem 3. The difference is that the integral over λ is split into two integrals,

$$\begin{aligned} &p(y^{N-L}|y^N_{N-L+1}, k) = p(y^k)p(y^N_{k+1}|y^N_{N-L+1}, k) = \\ &\int p(y^k|\lambda_1)p(\lambda_1|y^N_{N-L+1})d\lambda_1 \int p(y^N_{k+1}|y^N_{N-L+1}, k, \lambda_2)p(\lambda_2|y^N_{N-L+1})d\lambda_2. \end{aligned} \quad (26)$$

Each integral is evaluated exactly as in Theorem 3. □

Remark 1 *For this particular prior on λ, the integrals in (26), and thus also the likelihood, are not defined for $kp \leq 2$ and $Np - kp \leq 2$. This is logical because too few data are available to evaluate the noise variance.*

In this case the *a posteriori* distribution for λ_1, given the jump instant k, is $W^{-1}(kp, V^F(k))$ and for λ_2 it is $W^{-1}((N-k)p, V^B(k))$.

6.4 Summary

We can conveniently summarize the results in the three different cases as follows: The MLR's in Theorems 3, 4 and Algorithm 2 are given by

$$l_N(k) = D^F(N) - D^N(N) - D^F(k) - D^B(k+1) +$$

$$\begin{cases} +V^F(N) - V^N(N) - V^B(k) - V^B(k+1) & (i) \\ +(Np-2)\log(V^F(N) - V^N(N)) - (Np-2)\log(V^F(k) - V^B(k+1)) & (ii) \\ +(Np-2)\log(V^F(N) - V^N(N)) - (kp-2)\log V^F(k) \\ \quad -(Np-kp-2)V^B(k+1) & (iii) \end{cases}$$

in the cases known lambda (i), unknown constant lambda (ii) and unknown changing lambda (iii), respectively.

- The model and data dependent quantities $V(k)$ and $D(k)$ are all given by the two filter algorithm 2. 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 $V^F(k)$ and $V^B(k)$, as simulations have shown. When the noise is unknown, $(V^F(k) + V^B(k))/\lambda$ is essentially replaced by $N\log(V^F(k) + V^B(k))$ and $k\log V^F(k) + (N-k)\log V^B(k)$, respectively. This leads to a more cautious estimator, that diminishes the influence of the innovations.
- The term $D^F(N) - D^N(N) - D^F(k) - D^B(k+1)$ appears in all likelihood ratios. It is positive and does not vary much for different jump instants $k = 1, 2, .., N-1$. This term corresponds to the threshold in the GLR test.

These methods are compared in the simulation section.

7 Simulation results

The applicability of GLR is well-known as mentioned in the introduction. The MLR uses the same model and, thus, can be applied to the same problems as GLR. Therefore, the purpose of the current section is to show what can be gained in robustness and computational burden by using MLR instead of GLR illustrated by a quite simple example. A rather short data length will be used, that allows us to compute the exact GLR test.

A sampled double integrator will be examined in this comparative study of the different methods. For instance, it can be thought of as a model for the position of an object influenced by a random force. The state space model for sample interval 1 is

$$x_{t+1} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} x_t + \begin{pmatrix} 0.5 \\ 1 \end{pmatrix} v_t + \delta(t-k)\nu$$

$$y_t = (1\ 0)x_t + e_t$$

where

$$\begin{aligned} v_t &\in \mathrm{N}(0, \lambda) \\ e_t &\in \mathrm{N}(0, \lambda R) \end{aligned}$$

The jump change corresponds to a sudden force disturbance caused by a manoeuvre for example. All filters are initialized assuming that $\hat{x}_{1|0} \in \mathrm{N}(0, 1000\lambda I)$ and the number of measurements is N. Default values are given by

$$\begin{aligned} \nu &= (5,\ 10)^T \\ k &= 25 \\ N &= 50 \\ \lambda &= 1 \\ R &= 1 \end{aligned}$$

The default values on λ and R are used to compute the change detectors. The detectors are kept the same in all cases and the data generation is varied in order to examine the robustness properties.

7.1 A Monte Carlo simulation

The following detection methods are compared:

- GLR in Algorithm 1 and using a sliding window of size 10 (referred to as GLR(10)).
- MLR in Algorithm 2 for an assumed known scaling $\lambda = 1$ (MLR 1), Theorem 3 for unknown scaling (MLR 2) and Theorem 4 for unknown and changing scaling (MLR 3), respectively.

For the two-filter methods MLR 1-3, five extra data points were simulated and used for initialization. Table 1 shows the alarm rates for no jump and jump, respectively, while Table 2 shows the estimated jump time in the cases of a jump. The left column indicates what has been changed from the perfect modeling case.

We note that the sliding window approximation of GLR is indistinguishable from the exact implementation. The alarm rates of GLR for the perfect modeling case are slightly larger than for MLR with the chosen threshold. Taking this fact into

Case	GLR	GLR(10)	MLR 1	MLR 2	MLR 3
Perfect modeling	0.08	0.08	0.01	0.01	0.04
Perfect modeling, jump	0.99	0.99	0.97	0.92	0.95
Perfect modeling, jump [1;2]	0.10	0.10	0.02	0.01	0.04
Perfect modeling, jump at $t = 40$	1	1	0.94	0.88	0.91
Perfect modeling, jump at $t = 10$	1	1	0.96	0.91	0.97
10 times increase in λ	1	1	0.97	0.10	0.99
10 times increase in λ, jump	1	1	0.99	0.14	1
100 times increase in λ	1	1	1	0.14	1
100 times increase in λ, jump	1	1	1	0.15	1
$\lambda = 100$	1	1	1	0.01	1
$\lambda = 100$, jump	1	1	1	0.01	1
$\lambda = 10$	1	1	1	0.01	0.43
$\lambda = 10$, jump	1	1	1	0.02	0.49
$\lambda = 2$	0.50	0.50	0.17	0.01	0.08
$\lambda = 2$, jump	0.99	0.99	0.96	0.61	0.79
$\lambda = 0.5$	0.02	0.02	0	0.03	0.05
$\lambda = 0.5$, jump	1	1	0.96	0.98	0.99
$\lambda = 0.1$	0.02	0.02	0	0.12	0.10
$\lambda = 0.1$, jump	0.99	0.99	0.96	1	1
$\lambda = 0.01$	0	0	0	0.23	0.20
$\lambda = 0.01$, jump	0.99	0.99	0.97	1	1
$\lambda = 0.01$, jump [2;4]	0.12	0.12	0.02	0.50	0.48
$R = 0.1$	0	0	0	0.01	0
$R = 0.1$, jump	1	1	1	1	1
$R = 0.5$	0	0	0	0.01	0.01
$R = 0.5$, jump	1	1	0.99	1	1
$R = 2$	0.77	0.77	0.35	0.01	0.10
$R = 2$, jump	0.99	0.99	0.91	0.45	0.66
$R = 10$	1	1	0.99	0.02	0.59
$R = 10$, jump	1	1	0.99	0.03	0.63

Table 1: Alarm rate for 1000 Monte Carlo simulations for different cases of modeling errors

Case	GLR	GLR(10)	MLR 1	MLR 2	MLR 3
Standard jump	25	25	25	25	25
Jump [1;2]	29	29	25	23	22
Jump at $t = 40$	40	40	40	40	40
Jump at $t = 10$	10	10	10	10	10
10 times increase in λ	31	31	33	30	25
100 times increase in λ	37	37	38	39	25
$\lambda = 100$	25	25	27	25	21
$\lambda = 10$	26	26	26	27	23
$\lambda = 2$	25	25	25	25	25
$\lambda = 0.5$	25	25	25	25	25
$\lambda = 0.1$	25	25	25	25	25
$\lambda = 0.01$, jump $[2; 4]$	25	25	27	22	20
$R = 0.1$	25	25	25	25	25
$R = 0.5$	25	25	25	25	25
$R = 2$	25	25	25	25	25
$R = 10$	25	25	27	22	20

Table 2: Estimated change time for 1000 Monte Carlo simulations for different cases of modeling errors

account, there is no significant difference between GLR and MLR 1. MLR 2 and 3 give somewhat larger false alarm rate and smaller detection probabilities in the perfect modeling case. This is no surprise since less prior information is used, which becomes apparent in these short data sets. The cases where $\lambda < 1$ or $R < 1$, both implying that the measurement noise variance is smaller than expected, cause no problems for GLR and MLR 1. Note, however, how MLR 2 and 3 take advantage of this situation and here can detect smaller changes than GLR and MLR can. In the case $\lambda = 0.01$ and $\nu = [2; 4]$ the probability of detection is 50%, while MLR 1 only detect 2% of these jumps.

The real problem is of course the cases where the measurement noise variance is larger than modeled. Here the false alarm rate for GLR and MLR 1 is close to one. On the other hand, MLR 2 has a very small and MLR 3 a fairly small false alarm rate. Of course, it becomes harder to detect the fixed size jump that is hidden in large noise.

The cases of a suddenly increasing noise scaling is excellently handled by MLR 2 and 3. The former gives no alarm, because this kind of change is not included in the model, and the latter quite correctly estimates a change at time 25.

We can also illustrate the difference in performance by plotting the average log likelihood ratios $2\log p(y^N|k,\hat{\nu})\ /p(y^N|k=N)$ and $2\log p(y^N|k)/p(y^N|k=N)$, respectively, as a function of jump time $k = 1, 2, .., N$. This is done in Figure 1 for GLR and MLR 1,2,3. A change is detected if the peak value of the log likelihood ratio is larger than zero for MLR and larger than $h = 6$ for GLR. Remember that the GLR log likelihood ratio is always positive.

The first plot shows the perfect modeling case, and the peak values are well above the respective thresholds. Note that MLR 1 and GLR are very similar in their shape except for a constant offset as stated already in Lemma 1.

The second plot illustrates what happens after an abrupt change in noise scaling. GLR and MLR 1 become large for all $t > 25$ and the estimated change times are distributed over the interval $[25, 50]$. MLR 2, which assumes an unknown and constant scaling, handles this case excellently without any peak, while MLR 3 quite correctly has a peak at $t = 25$ where a change in scaling is estimated accurately.

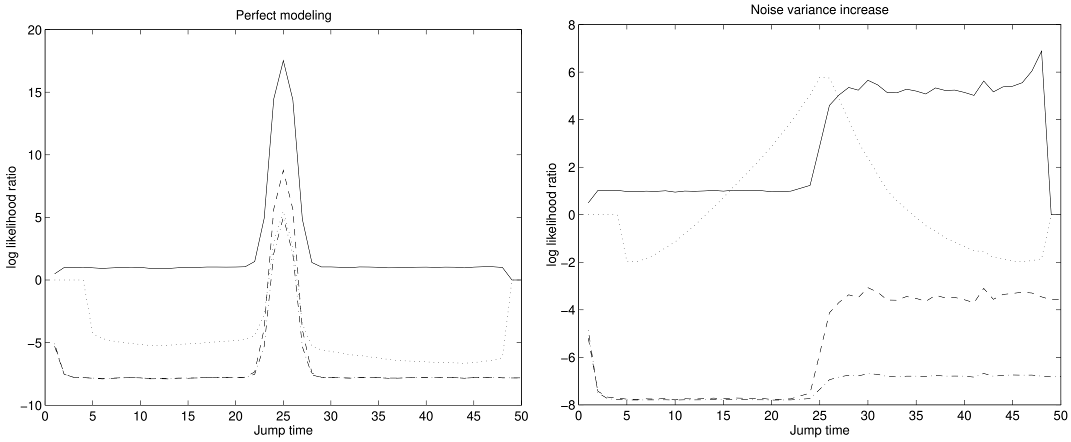

Figure 1: Log likelihood ratios for GLR (solid) and MLR 1,2,3 (dashed, dashed-dotted and dotted, respectively) averaged over 100 realizations. Left plot for perfect modeling and a jump, right plot for the case of no jump but the noise variance changes abruptly from 1 to 100 at time 25.

7.2 Complexity

Figure 2 shows the the complexity as a function of the number of observations N, counted in the number of used flops, for the following methods: GLR (M1) and GLR with sliding window (M7), MLR using GLR quantities as in Theorem 5 (M2), the direct implementation of MLR in Theorem 1 (M3), MLR 1 (M4), MLR 2 (M5) and MLR 3 (M6). It should be remarked, that the algorithms are intended to be implemented as efficiently as possible and identical Kalman filter implementations are used.

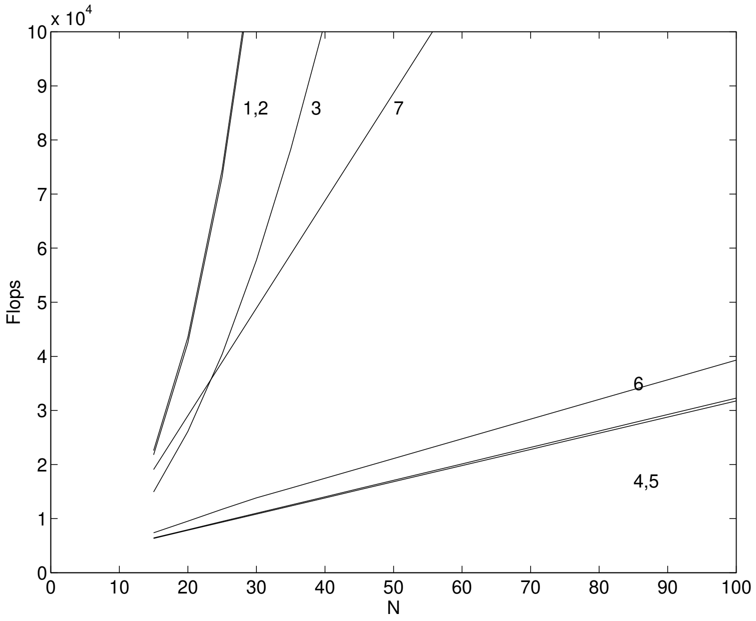

Figure 2: Complexity in flops for different implementations of GLR (M1 and M7) and MLR (M2-M6) as a function of the number of measurements.

The implementation in [15] with matched filters is not very efficient, since the direct implementation with Kalman filters instead of RLS schemes is actually faster. This is due to the computation of the regressors. 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.

The time consumption for GLR with a sliding window of size 10 increases linearly with time and it is about five times slower than the two filter approach.

8 Conclusions

The MLR test has been introduced as an alternative to GLR. A quadratic in time complexity on-line implementation was given, which is slightly faster than the exact GLR, as well as an linear in time complexity off-line implementation. Although on-line change detection is practically more relevant, the very fast off-line implementation is of great use in Monte-Carlo simulations that is almost inevitable for evaluating the change detection design with respect to robustness to modeling errors, detection probabilities for different changes and so on.

In contrast to classical Kalman filter design where only the signal to noise ratio has to be known, the actual magnitudes of all noise variances have to be known in change detection. The MLR test was derived for three different cases, known variances, all variances contain an unknown but constant scaling (not affecting the signal to noise ratio) and the case of an abruptly changing noise variance. Although the cases with unknown scalings were derived only for the off-line two-filter implementation, the derivation for the on-line case is straightforward. A simulation study supported the motivation of the MLR approach:

- The MLR test works well without the need of designing a threshold. It should be remarked that the user can, if required, include an *ad-hoc* threshold straightforwardly.
- The MLR test with assumed unknown noise scaling performs superiorly in cases where the measurement noise variance is under-estimated or suddenly increases. In these cases, the false alarm rate is much smaller than for GLR or MLR with a fixed scaling. When the real noise variance is smaller than expected, the MLR test with unknown scaling adopts to the new situation and is able to detect smaller changes.
- The MLR test with assumed changing noise scaling is able to detect increases in measurement noise and performs fairly well when the noise variance is under-estimated.

That is, the MLR approach greatly facilitates the design of a change detector by relaxing the amount of prior information and enabling interactive Monte-Carlo evaluation and MLR also improves the robustness properties with respect to incorrectly chosen or changing noise variances.

A Comparing on-line and off-line expressions

We will here derive two equivalent ways of computing likelihoods in a linear regression framework: one on-line and one off-line. First, some least squares relations are given.

Consider the linear regression $y_t = \varphi_t^T \theta + e_t$ and the loss function $V_t(\theta) = \sum_{k=1}^t (y_k - \varphi_k^T \theta)^T R_k^{-1} (y_k - \varphi_k^T \theta)$. Assume that the RLS estimate at time $t-1$ is $\hat{\theta}_{t-1}$ with covariance matrix P_{t-1}. Then a new measurement gives the update formulas:

$$\begin{aligned} \hat{\theta}_t &= \hat{\theta}_{t-1} + P_{t-1}\varphi_t[\varphi_t^T P_{t-1}\varphi_t + R_t]^{-1}(y_t - \varphi_t^T \hat{\theta}_{t-1}) \\ P_t &= P_{t-1} - P_{t-1}\varphi_t[\varphi_t^T P_{t-1}\varphi_t + R_t]^{-1}\varphi_t^T P_{t-1}. \end{aligned} \tag{27}$$

Here the initial conditions are given by the prior, $\theta \in \mathrm{N}(\theta_0, P_0)$. The a posteriori distribution of the parameter vector is

$$\theta_t \in \mathrm{N}(\hat{\theta}_t, P_t).$$

The probability density function for the observations is given in the following lemma. The likelihood for data given for instance the initial value ν is simply the conditional density function $p(y^N|\nu)$.

Lemma 3 *The density function of the sequence $y^N = \{y_k\}_{k=1}^N$ from $y_t = \varphi_t^T\theta + e_t$ is computed recursively by*

$$p(y^N) = p(\varepsilon^N) \tag{28}$$

where $\varepsilon_t = y_t - \varphi_t^T\hat{\theta}_{t-1} \in \mathrm{N}(0, \varphi_t^T P_{t-1}\varphi_t + R_t)$. Here $\hat{\theta}_t$ and P_t are given by the RLS scheme in (27) with initial conditions θ_0 and P_0. Furthermore, $e_t \in \mathrm{N}(0, R_t)$ and φ_t is a known sequence.

Proof: Iteratively using Bayes' law gives

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

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

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

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

Lemma 4 *The off-line counterpart to (28) is*

$$p(y^N) = p(y^N|\hat{\theta}_N)p_\theta(\hat{\theta}_N)\left(\det P_N\right)^{1/2}(2\pi)^{\frac{d}{2}} \tag{29}$$

if the prior on θ is Gaussian or

$$p(y^N) = p(y^N|\hat{\theta}_N)\left(\det P_N\right)^{1/2} \tag{30}$$

if the prior is non-informative. Here $d = \dim\theta$.

Proof: See [5], Lemmas B.2 and B.3.

Since the data sequence is arbitrary we also have

$$\sum_{t=1}^{N}(y_t - \varphi_t^T\hat{\theta}_{t-1})^T \left(\varphi_t^T P_{t-1}\varphi_t + R_t\right)^{-1}(y_t - \varphi_t^T\hat{\theta}_{t-1}) = \tag{31}$$

$$= \sum_{t=1}^{N}(y_t - \varphi_t^T\hat{\theta}_N)^T R_t^{-1}(y_t - \varphi_t^T\hat{\theta}_N) + (\theta_0 - \hat{\theta}_N)^T P_0^{-1}(\theta_0 - \hat{\theta}_N) \tag{32}$$

$$\sum_{t=1}^{N}\log\det\left(\varphi_t^T P_{t-1}\varphi_t + R_t\right) = C - \log\det P_N \tag{33}$$

Note that the right-hand sides involve just the final values $\hat{\theta}_N$ and P_N.

B Derivation of the GLR test

B.1 Regression model for the jump

First, process the measurements through a Kalman filter acting under the hypotheses of no jump. With notation as in (2), denote its state estimate, gain, prediction error and prediction error covariance by

$$\hat{x}_{t|t}, \quad K_t = P_{t|t-1}H_t S_t^{-1}, \quad \varepsilon_t = y_t - H_t\hat{x}_{t|t-1}, \quad S_t = R_t + H_t P_{t|t-1}H_t^T.$$

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,

$$\hat{x}_{t|t}(k) = \hat{x}_{t|t} + \mu_t(k)\nu \tag{34}$$

$$\varepsilon_t(k) = \varepsilon_t + \varphi_t^T(k)\nu. \tag{35}$$

Here $\hat{x}_{t|t}(k)$ and $\varepsilon_t(k)$ are the quantities one would have obtained from a Kalman filter applied under the assumption on a jump at time k. These Kalman filters will however not be used explicitly, and that is the key point. Again, we follow the convention that $k = t$ means no jump. We have the following update formulas for the $n \times n$ matrix $\mu_t(k)$ and the $n \times 1$ vector $\varphi_t(k)$, first given in [15].

Lemma 5 *Consider the residuals from the Kalman filter applied to the model (1) assuming no jump. 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*

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

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

$$\begin{aligned}
\varphi_{t+1}^T(k) &= H_{t+1}\left(\prod_{i=k}^{t} F_i - F_t\mu_t(k)\right) && (36)\\
\mu_{t+1}(k) &= F_t\mu_t(k) + K_{t+1}\varphi_{t+1}^T(k), && (37)
\end{aligned}$$

with the initial conditions $\mu_k(k) = 0$ and $\varphi_k(k) = 0$. Here K_t is the Kalman gain at time t.

Proof: The result is proved by induction. The initial condition is a direct consequence of the signal model (1). The induction step follows from the Kalman equations, (34) and (35). First we have the relation $x_t(k) = x_t + \prod_{i=k}^{t-1} F_i\nu$. This and equations (34) and (35) give

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

and

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

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.

B.2 The GLR test

We begin by deriving the classical GLR test, first given in [15], where the jump magnitude is considered as deterministic. Lemma 3 gives

$$p(y^N) = p(\varepsilon^N).$$

That is, the measurements and the residuals have the same probability density function. This equality can also be stated directly, since the transformation from measurement to residuals is linear with a Jacobian equal to one so the density is unchanged under this transformation. Given a jump ν at time k, the residuals get a bias,

$$\varepsilon_t \in \mathrm{N}(\varphi_t^T(k)\nu, S_t). \tag{38}$$

Thus, the test statistic (6) can be written as

$$l_N(k,\hat{\nu}(k)) = 2\log\frac{p(\varepsilon_1^N|k,\hat{\nu}(k))}{p(\varepsilon_1^N|k=N)}. \tag{39}$$

Introduce the well known compact quantities of the LS estimator

$$f_N(k) = \sum_{t=1}^{N}\varphi_t(k)S_t^{-1}\varepsilon_t \tag{40}$$

$$R_N(k) = \sum_{t=1}^{N}\varphi_t(k)S_t^{-1}\varphi_t^T(k). \tag{41}$$

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

$$\hat{\nu}(k) = R_N^{-1}(k)f_N(k). \tag{42}$$

We get

$$\begin{aligned}
l_N(k,\hat{\nu}(k)) &= 2\log\frac{p(\varepsilon_1^N|k,\hat{\nu}(k))}{p(\varepsilon_1^N|k=N)} && (43)\\
&= \sum_{t=k+1}^{N}\varepsilon_t^T S_t^{-1}\varepsilon_t - (\varepsilon_t-\varphi_t^T(k)\hat{\nu}(k))^T S_t^{-1}(\varepsilon_t-\varphi_t^T(k)\hat{\nu}(k)) && (44)\\
&= f_N^T(k)R_N^{-1}(k)f_N(k), && (45)
\end{aligned}$$

where the second equality follows from (38) and the Gaussian probability density function. The third equality follows from straightforward calculations using (42), (40) and (41). This simple expression for the test statistic is a appealing property of the GLR test.

C LS based derivation of the MLR test

The idea here is to consider the jump magnitude as a stochastic variable. In this way the maximization over ν in $l_N(k)$ (6) can be avoided. Instead, the jump magnitude is eliminated by integration,

$$p(\varepsilon_1^N) = \int p(\varepsilon_1^N|\nu)p(\nu)d\nu.$$

In the context of likelihood ratio tests, this possibility to integrate out the nuisance parameter ν is also discussed in [13]. Here, there are two choices. Either ν is assumed to have a Gaussian prior or it is considered to have infinite variance, so the (improper) prior is constant. The log likelihood ratio is given in the following theorem.

Theorem 5 *Consider the GLR test in Algorithm 1 as an estimator of k and ν, $k = 1, 2, ..., N-1$, for the signal model (1), given the measurements y^N. The LR, corresponding to the estimator of k alone, is given by*

$$\begin{aligned} l_N(k) &= 2\log\frac{p(y^N|k)}{p(y^N)} \\ &= \arg\max_{\nu} 2\log\frac{p(y^N|k,\nu)}{p(y^N)} - \log\det R_N(k) + C_{prior}(k) \\ &= l_N(k,\hat{\nu}(k)) - \log\det R_N(k) + C_{prior}(k) \end{aligned}$$

where $l_N(k,\hat{\nu}(k)) = f_N^T R_N^{-1} f_N$ is given in Algorithm 1. Here $C_{prior}(k)$ is a prior dependent constant that equals

$$C_{prior}(k) = p\log(2\pi) \tag{46}$$

if the prior is chosen to be non-informative and

$$C_{prior}(k) = -\log\det P_\nu + (\hat{\nu}_N(k) - \nu_0)^T P_\nu^{-1}(\hat{\nu}_N(k) - \nu_0) \tag{47}$$

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 (42) and (41), respectively.

Proof: We begin with the Gaussian case.

$$\begin{aligned} 2\log\frac{p(y^N|k)}{p(y^N)} &= 2\log\frac{p(y^k)p(y_{k+1}^N|y^k,k)}{p(y^k)p(y_{k+1}^N|y^k)} \\ &= \sum_{t=k+1}^{N} \varepsilon_t^T S_t^{-1}\varepsilon_t - (\varepsilon_t - \varphi_t^T(k)\hat{\nu}_N(k))^T S_t^{-1}(\varepsilon_t - \varphi_t^T(k)\hat{\nu}_N(k)) \\ &\quad + \log\det P_N(k) - \log\det P_\nu + (\hat{\nu}_N(k) - \nu_0)^T P_\nu^{-1}(\hat{\nu}_N(k) - \nu_0) \\ &= l_N(k,\hat{\nu}(k)) - \log\det R_N(k) + C_{prior}(k) \end{aligned}$$

In the second equality the off-line expression from Lemma 4 is used for $p(y_{k+1}^N)$.

The case of the non-informative prior is proved as above, again Lemma 4 is used for $p(y_{k+1}^N)$.

$$2\log\frac{p(y^N|k)}{p(y^N)} = 2\log\frac{p(y^k)p(y_{k+1}^N|y^k,k)}{p(y^k)p(y_{k+1}^N|y^k)}$$

$$
\begin{aligned}
&= \sum_{t=k+1}^{N} \varepsilon_t^T S_t^{-1}\varepsilon_t - (\varepsilon_t - \varphi_t^T(k)\hat{\nu}_N(k))^T S_t^{-1}(\varepsilon_t - \varphi_t^T(k)\hat{\nu}_N(k)) \\
&\quad + \log\det P_N(k) + p\log(2\pi) \\
&= l_N(k, \hat{\nu}(k)) - \log\det R_N(k) + C_{prior}(k)
\end{aligned}
$$

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, the likelihood ratios are asymptotically equivalent except for a constant. This constant can be interpreted as different thresholds as done in Lemma 1.

In this constant, the term $(\hat{\nu}_N - \nu_0)^T P_\nu^{-1}(\hat{\nu}_N - \nu_0)$ is negligible if the prior uncertainty P_ν is large or, since $\hat{\nu}_N(k)$ converges to ν_0 ($\hat{\nu}_N(k) \to \nu_0$ as $N-k \to \infty$ because the Kalman filter eventually tracks the abrupt change in the state vector), if $N-k$ is large. As will be seen in the simulation section, the term $\log\det R_N(k)$ does not alter the likelihood significantly. That it is asymptotically constant is formally proved in the following lemma.

Lemma 6 *Let $\varphi_t(k)$ be as given in Result 5. Then, if the signal model (1) is stable and time-invariant,*

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

converges as $N-k$ tends to infinity.

Proof: Rewriting (37) using (36) gives

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

where $\overline{K}$ is the stationary 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 $F\overline{K}$ here). 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 $l > m$

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

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

We have now proved that the log likelihoods for the two variants of parameter vectors (k, ν) and k are approximately equal for $k = 1, 2, .., N-1$, except for an unknown constant, so

$$l_N(k, \hat{\nu}_N(k)) \approx l_N(k) + C.$$

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), since $l_N(N, \hat{\nu}_N(N)) = l_N(N) = 0$. To get equivalence for this case as well, the threshold in the GLR test has to be chosen to this unknown constant C.

References

[1] B.D.O. Anderson and J.B. Moore. *Optimal Filtering.* Prentice Hall, Englewood Cliffs, NJ., 1979.

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

[3] M. Basseville and A. Benveniste. *Detection of Abrupt Changes in Signals and Dynamical Systems.* Lecture Notes in Control and Information Sciences. Springer-Verlag, 1986.

[4] M. Basseville and I.V. Nikiforov. *Detection of abrupt changes: theory and application.* Information and system science series. Prentice Hall, Englewood Cliffs, NJ., 1993.

[5] Fredrik Gustafsson. *Estimation of discrete parameters in linear systems.* Dissertations no. 271, Linköping University, Sweden, 1992.

[6] R Isermann. Process fault detection based on modeling and estimation methods - a survey. *Automatica*, 20:387–404, 1984.

[7] T. Kerr. Decentralized filtering and redundancy management for multisensor navigation. *IEEE Trans. on Aerospace and Electronic Systems*, 23:83–119, 1987.

[8] E.L. Lehmann. *Testing statistical hypothesis.* Statistical/Probability series. Wadsworth & Brooks/Cole, 1991.

[9] 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, Budapest, 1991. IFAC.

[10] I.V. Nikiforov. Sequential detection of changes in stochastic processes. In *9th IFAC/IFORS Symposium on Identification and System Parameter Identification*, pages 11–19, Budapest, 1991. IFAC.

[11] R. Patton, P. Frank, and R. Clark. *Fault diagnosis in dynamic systems.* Prentice Hall, 1989.

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

[13] A. Wald. *Statistical Decision Functions.* Wiley, New York, 1950.

[14] A.S. Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12:601–611, 1976.

[15] 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.