NTNU
Norwegian University of
Science and Technology

Adaptive Tuning of Acoustic Positioning Sensor

Fredrik Ulvin

Master of Science in Cybernetics and Robotics
Submission date: June 2017
Supervisor: Edmund Førland Brekke, ITK
Co-supervisor: Are Willumsen, Kongsberg Maritime AS

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

The candidate will study "Adaptive Tuning of Acoustic Positioning Sensor".

The work includes:

1. Comparison of different methods for tuning the measurement noise covariance.
2. Investigating the possibilities of tuning additional parameters such as process noise covariance.
3. Quantify estimability, observability and performance in general.

Abstract

This thesis compare methods of noise covariance estimation with the aim to improve the percision of Extended Kalman Filter (EKF) estimates of acoustic distance measurements. Because computational load is a priority, only suboptimal estimators have been considered; four relying on innovation covarianse matching and one sequential Maximum Likelihood Estimator. The methods will mainly be compared by the mean positional error, the filter state estimate covariance, and the estimators ability to hit the target covariance. A simulated HiPAP acoustic measurement signal was used as the measurement for the adaptive filters. The methods performance and sources of errors have been discussed. Simulations show that the process noise covariance is needed to be known or estimated to ensure performance during noise covariance estimation. Moreover simulations show that the proposed combination estimator outperforms the established methods, if trained on measurements under static noise conditions.

Sammendrag

I denne diplomoppgaven sammenlignes forskjellige metoder for å estimere kovarianser for støy, med mål å øke treffsikkerheten til Utvidet Kalman Filters (EKF) estimat av akustiske avstandsmålinger. Ettersom lav kjøretid er viktig faktor blir bare suboptimale metoder undersøkt; fire basert på innovasjons-kovarians ekvivalens, og en basert på sannsynlighets maksimering. Hovedsakelig sammenlignes metodenes estimerte posisjonsfeil, filterets estimat-kovarianse, og treffsikkerheten til støy-kovarianse estimatet. Simulerte HiPAP akustisk målesignal har blitt brukt som målinger for filterene. Filtrenes ytelse og feilkilder har blitt diskutert. Simuleringene viser at prosesstøy kovariansen må være kjent eller estimeres for å sikre ytelsen til for et filter med målestøy kovarianse estimaring.

Acknowledgements

Thank to my friends, family, girlfriend, and not least my supervisors, Edmund and Are. Thank you all for your boundless patience.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

ALMF	**A**daptive **L**imited **M**emory **F**ilter
MLE	**M**aximum **L**ikelihood **F**ilter
SH	**S**age-**H**usa estimator
TANIS	**T**ime-**A**veraged **I**nnovation **S**quared
HiPAP	**Hi**gh **P**ercision **A**coustic **P**ositioning
HAIN	**H**ydroacoustic **A**ided **I**nertial **N**avigation
KF	**K**alman **F**ilter
EKF	**E**xtended **K**alman **F**ilter

For/Dedicated to/To my...

CHAPTER 1

MOTIVATION

Filtering is required to remove measurement errors thereby improving the performance of acoustic sensors. Kalman Filter (KF) (and variations) are commonly used, which rely on knowledge of the noise statistical properties to produce estimates of the measurement free of noise. In cases with unknown noise covariance, KF cannot be garanteed to consistently give more accurate or percise results than the measurements themselves.

A method of estimating the noise covariance must then be used in conjunction with KF to ensure consistent performance in light of changing noise characteristics. The field of estimation is wide, and has been divided into four categories[1]; *bayesian estimation* and *maximum likelihood methods* finds the most likely estimates given the measurements; *correlation matching methods* find estimates which correlates the measurements; and *covariance matching* find covariance estimates that match observed covariance from the measurements. Filters that are extended with estimation of design parameters are called *adaptive filters*.

The estimators will be added to a EKF to form various adaptive filters. They will be tested on a simulated HiPAP[2] measurement. The HiPAP sensor is a part of the sensor package HAIN [3], produced by Kongsberg Maritim.

1.1 Objective

1. Do a litterary search for low-resource system non-specific methods for on-line covariance estimation.
2. Investigate problems related to the methods found.
3. Evaluate the methods performance using simulated HiPAP measurements.
4. Propose a method that can increase the accuracy of HiPAP measurements.

1.2 Outline

Chapter 2 provides a theoretical background. Chapter 3 present the different relevant methods tested. Chapter 4 contains plots of cost functions, which are used to evaluate the adaptive filters. costs for filters with and without estimation will be shown. Chapter 5 compares the methods based on their performance on simulated measurements. Lastly concluding remarks and suggestions for further work is found in chapter 6.

CHAPTER 2

THEORY

2.1 Test Signal

Later a signal that imitate HiPAP[2] measurements will be generated. The measurements give the position of a vessel, with some added noise.

The change in position is given by the target vessels dynamics $f(p, u)$, where p is the position and u is the vessels input. Since the noise covariance is of interest here, it is assumed the position is contant. Furthermore it cannot be assumed that the modeling is perfect, so the accumulated error is included as a zero mean white noise v_x.

The change in position is then given as

$$\dot{p} = f(p, u) + v_p \tag{2.1}$$

Over time the measurements will be corrupted by a increasing bias. It is modeled as a first order Gauss-Markov process.

$$\dot{b} = -\frac{1}{T}b + v_b \tag{2.2}$$

T is a time constant, determining how fast the bias is change. v_b is white noise with zero mean.

The resulting measurements are then the sum of the position p, bias b and measurement noise w. Collected this becomes

$$\begin{bmatrix} \dot{p} \\ \dot{b} \end{bmatrix} = \begin{bmatrix} f(p) \\ -\frac{1}{T}b \end{bmatrix} + \begin{bmatrix} v_x \\ v_b \end{bmatrix} \tag{2.3}$$

$$y = \begin{bmatrix} 1 & 1 \end{bmatrix} x + w \tag{2.4}$$

HiPAP determines the vessel position from the measured distance from its different sensors. Since this is not continous, the above equations needs to be discretized.
Δ_t denote the time step length.

$$\begin{bmatrix} p_k \\ b_k \end{bmatrix} = \begin{bmatrix} 1 + \frac{\partial f(p,u)}{\partial p}\Delta_t & 0 \\ 0 & 1 - \frac{\Delta_t}{T} \end{bmatrix} x_{k-1} + \begin{bmatrix} \frac{\partial f(p,u)}{\partial u}\Delta_t \\ 0 \end{bmatrix} u + \begin{bmatrix} v_x \\ v_b \end{bmatrix} \tag{2.5}$$

$$y_k = \begin{bmatrix} 1 & 1 \end{bmatrix} x_k + w \tag{2.6}$$

Because the target is stationary $f(p,u) = C \rightarrow \frac{\partial f(p,u)}{\partial p} = 0$. This means that there is no input $u = 0$.

2.2 Kalman Filter

The Kalman Filter (KF) is a true and tested estimation algorithm that finds estimates which minimize the state quadratic error from measurements corrupted by noise [4]. It can be used for smoothing, filtering out noise, or predicting, both for discrete and continous signals.

KF can be designed to work on continous or discreete signals. In this thesis filters for a discreete system has been concidered.

The algorithm consist of two main steps. In the prediction step the previous estimate and a model of the system dynamics is used to predict what the estimated variables are that iteration.

Next, in the Update step, the prediction is compared to the incomming measurement. The estimate is then updated using the difference modified by a gain K. The KF gain K is determined by the uncertainty of the measurement and the filter model.

KF was developed to work with linear systems. But HiPAP measure the position of some vessel, which dynamics are non-linear. Several modifications to KF has been made to accomidate non-linear systems. The most commonly used is the Extended Kalman Filter (EKF). EKF linerizes the dynamic equations around the current state estimates.

KF is optimal in the sense that it minimize the mean square error of the produced estimates. It is optimal when the system is modeled perfectly, the noise is uncorrelated, and the covariance of the noise is known. Since the linearization is a simplification, the EKF will be sub-optimal.

2.2.1 Equations

EKF has some assumtions.

- Initial values $P_0, \hat{x}_0$ are known.
- Measurement R and process noise Q covariances are known.
- The system dynamics can be linearlized.
- The noises are uncorrelated with each other and in time.

EKF has the following procedure.

Linerization

Calculate the state transition ϕ_k and output H_k matrix by taking the Jakobian of the model equations (2.3), (2.4).

Since the position of a stationary target is measured the matrices are quite simple and time invariant.

$$\phi = \left.\frac{\partial \dot{x}}{\partial x}\right|_{x=\hat{x}_{k-1}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 - \frac{1}{T}\Delta_t \end{bmatrix} \tag{2.7}$$

$$H = \left.\frac{\partial C}{\partial x}\right|_{x=\hat{x}_{k-1}} = \begin{bmatrix} 1 & 1 \end{bmatrix} \tag{2.8}$$

Prediction

Predict the state in this step.

$$\hat{x}_k^- = \phi \hat{x}_{k-1} \tag{2.9}$$

Predict the state covariance.

$$P_k^- = \phi P_{k-1} \phi^T + Q \tag{2.10}$$

Update

Calculate the innovation from the measurement and the state prediction.

$$\nu_k = y_k - H\hat{x}_k^- \tag{2.11}$$

Calculate the innovation covariance.

$$C_{\nu,k} = H P_k^- H^T + R \tag{2.12}$$

Calculate the KF gain.

$$K_k = P_k^- H^T C_{\nu,k}^{-1} \tag{2.13}$$

Calculate the estimate from the predicted states and innovation.

$$\hat{x_k} = \hat{x}_k^- + K_k \nu_k \tag{2.14}$$

Update the estimate covariance.

$$P_k = (I - K_k H) P_k^- \tag{2.15}$$

As usual, Q and R are the covariance matrices for process and measurement noise respectivly.

2.2.2 Consistency

EKF is designed to work for given noise characteristics. If it changes the initial assuptions are no longer valid, and the filter does not neccessarily produce good state estimates. The filter has then become inconsistent.

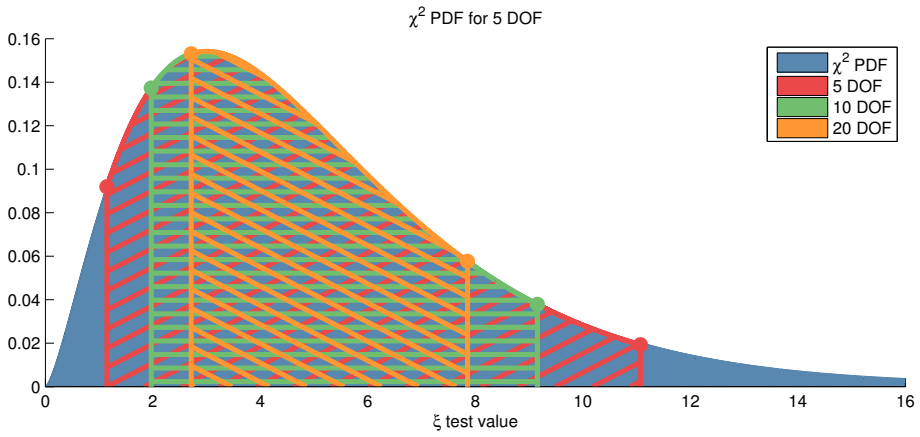

FIGURE 2.1: χ^2 PDF for 5 DOF. Significance region $5\% - 95\%$ for different DOFs projected onto the $\chi^2(5)$ distribution.

A consistent filter is able to produce results with the same accuracy over time. There are several measures of consistency, but only two can be used on-line. One which is of particular interest of the thesis is the Time-Averaged Normalized Innovation Squared (TANIS).

The measurement innovation ν is the differene between the expected and the real measurement. Its theoretical covariance C_ν is defined as

$$C_\nu = E[\nu_k \nu_k^T] = H P_k^- H^T + R \tag{2.16}$$

If ν is white with a zero-mean, $\nu^T C_\nu^{-1} \nu$ has a χ^2 distribution with DOF equal to the number of states in the system. As the sum of χ^2 distributed variables also is χ^2 distributed, the following test variable can be used for consistency testing.

The TANIS[5] test statistic is defined as

$$\xi(\boldsymbol{\nu}, S) = \frac{1}{N} \Sigma \nu_i^T C_\nu^{-1} \nu_i, \boldsymbol{\nu} = [\nu_1 \nu_2 \dots \nu_N] \tag{2.17}$$

Which has a χ^2 distribution with DOF = number of states * sample size N. This is a Goodness-to-Fit test which indicates how well the filter parameters fit the statistic ν.

The test sees if the test statistic $\xi(\boldsymbol{\nu}, S)$ lies within a confidence region determined by upper and lower probabilites given by the χ^2 PDF. The condidence region shrinks for larger DOF, as is shown in fig. 2.1. In this work the confidence region has limits set at 5% and 95%. Meaning that if $\xi(\boldsymbol{\nu}, S)$ lies outside the confidence region, there is a 90% chance the filter parameters are wrong.

2.3 Cost functions

As a measure of filter performance various cost functions can be employed. Most of these cost functions are derived from probability density functions (PDF) related to the KF. [6]

By taking the negative log of a PDF, and multiply it by a constant if neccesary creates a cost function which has a lower cost for higher probabilities.

2.3.1 Absolute State Error

The most simplistic cost function is the Absolute State Error. It looks only on how far the estimates are from the true states. As the covariance is not considered, this is usually not a good measure for a filter.

$$J^{abs} = \sum_{i=1}^{N} \mid x_i - \hat{x}_i \mid \tag{2.18}$$

2.3.2 State Estimation Error Squared

If the true value of the states are known, the PDF for the state estimated error can also be known. The State Estimation Error Squared shows the likelihood for the state estimate given measurements and tuning variables.

$$\rho(x|z,\theta) = \frac{1}{\sqrt{2\pi P}} e^{-(x-\hat{x})^T P^{-1}(x-\hat{x})} \tag{2.19}$$

Which result in the cost function

$$J^P = \sum_{i=1}^{N} [log_n(2\pi P_i) + (x_i - \hat{x}_i) P_i^{-1} (x_i - \hat{x}_i)^T] \tag{2.20}$$

The first term keep P down, while the second makes it fit to the state estimate error.

2.3.3 Measurement Prediction Error Squared (Innovation Squared)

Starting from the probability for a measurement, given estimated states and tuning parameters. Assuming ν has a Gaussian distribution, the PDF is

$$\rho(z|\hat{x},\theta) = \frac{1}{\sqrt{2\pi C_\nu}} e^{-\nu C_\nu^{-1} \nu^T} \tag{2.21}$$

Where θ is the filters tuning variables,C_ν and ν are as defined by eq. (2.11) and eq. (2.13). Gives the cost function

$$J^S = \sum_{i=1}^{N} [log(2\pi C_{\nu,i}) + \nu_i^T C_{\nu,i}^{-1} \nu_i] \tag{2.22}$$

Just as for J^P, the first term keeps C_ν from getting to large, while the second makes it large enought to fit ν.

2.3.4 Innovation Consistency

A cost function can be derived from the χ^2 PDF used in consistency testing (TANIS).

$$\chi^2(\xi(\boldsymbol{\nu}, C_\nu), k) = \frac{1}{2^{\frac{k}{2}}\Gamma(\frac{k}{2})} \xi(\boldsymbol{\nu}, C_\nu)^{\frac{k}{2}-1} e^{-\frac{\xi(\boldsymbol{\nu}, C_\nu)}{2}} \tag{2.23}$$

Where $\xi(\boldsymbol{\nu}, C_\nu)$ is the test value, k is degrees of freedom (DOF), and $\Gamma(\frac{k}{2})$ is the Gamma function.

Taking the negative log and removing constant terms results in the cost function

$$J^{\chi^2} = \sum_{i=1}^{N} [\xi(\boldsymbol{\nu}_i, C_{\nu,i}) - (k-2) log(\xi(\boldsymbol{\nu}_i, C_{\nu,i})) \tag{2.24}$$

2.4 Estimation

Measurements are often corrupted by noise, meaning that they cannot be used directly, and some scheme to 'see through' the noise is needed. This is the aim of estimation.

There are many methods depending on what is to be estimated, and certain requrements as on- or off-line use, specific computational restrictions, non-linearities, etc. In this thesis the aim is to get better position estimates under changing noise conditions. EKF already produce the position estimate, but as it assumes the noise covariances are known,

which does not hold if noise covariances change. The problem is then reduced to getting the correct noise covariances.

While covariance estimation is a difficult problem, it is also luckily well researched. Which has given rise to several estimation strategies. Most of these are used off-line iteration over all available data, and most cater to linear systems.

2.4.1 Covariance mathcing

One main types of covariance estimation uses covariance matching. The main idea is tuning the filter to match an observed covariance.

An unbiased approximation of covariance C is given as

$$\hat{C}_x = \frac{1}{N-1}\sum_{i=0}^{N}(x_i - \bar{x})(x_i - \bar{x})^T \tag{2.25}$$

Assuming zero mean, an biased estimate can be used in stead. The approximation then becomes

$$\hat{C}_x = \frac{1}{N}\sum_{i=0}^{N} x_i x_i^T \tag{2.26}$$

Measurement Noise Covariance

The measurement innovation ν (2.11) has the expected variance.

$$\nu = z - H\hat{x}^- \tag{2.27}$$
$$= H(x - \hat{x}^-) + v \tag{2.28}$$
$$C_\nu = HP_k^- H^T + R \tag{2.29}$$

Solving wrt. R, and using the calculated $\hat{C}_\nu$, R can be estimated by

$$\hat{R} = \hat{C}_\nu - HP_k^- H^T \tag{2.30}$$

Process Noise Covariance

Similarly looking at the estimate innovation $q = \hat{x} - \hat{x}^-$

$$q = \hat{x}_k - \hat{x^-}_k + x_k - x_k \tag{2.31}$$
$$= -(x_k - \hat{x}_k) + (x_k - \hat{x}^-) \tag{2.32}$$
$$= -(x_k - \hat{x}_k) + \phi_k(x_{k-1} - \hat{x}_{k-1}) + w_k \tag{2.33}$$
$$C_q = Cov[q] = -P_k + \phi_k P_{k-1}\phi_k^T + Q_k \tag{2.34}$$

Which shows that Q can be estimated as

$$\hat{Q}_k = \hat{C}_q - (\phi_k P \phi_k^T - P_k) \tag{2.35}$$

Some authors has used $q_k = K_k \nu_k \nu_k^T K_k^T$, which is equivilent to q as defined above.

2.4.2 Maximum Likelihood Estimation

Maximum likelihood methods seek to find the most probable states and parameters. As it is widely used, there are several different implementations. [7] They are garanteed to produce the most likely estimate available from the information used. However they are computationally expensive, and are mostly used off-line, iterating over the data several times. It is possible to use them on-line they are not as accurate then, especially for system with quickly changing characteristics.

2.4.3 Joint Estimation

If the covariances are unknown, the problem is extended to estimate them as well as the postition from the available information.

The most intuitive approach is finding $\hat{x}, \hat{\theta}$ that maximises the likelihood

$$\rho(x, \theta | z) \tag{2.36}$$

The PDF can be further expanded using Bayes rule to

$$max_{x,\theta}(\rho(x, \theta|z)) = max(\frac{\rho(z|x,\theta)\rho(x,\theta)}{\rho(z)}) \tag{2.37}$$
$$= max(\rho(z|x,\theta)\rho(x,\theta)) \tag{2.38}$$
$$= max(\rho(z|x,\theta)\rho(x|\theta)\rho(\theta)) \tag{2.39}$$
$$= max(\rho(z|x,\theta)\rho(x|\theta)) \tag{2.40}$$

In short, Joint Maximisation (JM) methods seeks after the tuning parameters θ that maximises the likelihood for both the measured signal and the state estimates.

2.4.4 Marginal Estimation

Estimating the variances and states from the same PDF can be difficult. JM have to search in several dimentions using only one measure.

The PDF can be extended however

$$\rho(x, \theta|z) = \rho(x|\theta, z)\rho(\theta|z) \tag{2.41}$$

To maximising the two terms in the right will also maximise the term on the left. These methods are called Marginal Likelihood Estimation in litterature.

Both $\rho(x|\theta, z)$ and $\rho(\theta|z)$ needs to be maximised to garantee optimality. Given an estimate of the variances, a KF will maximise the former PDF.

The latter PDF can be further extended using Bayes rule

$$\rho(\theta|z) = \frac{\rho(z|\theta)\rho(\theta)}{\rho(z)} \tag{2.42}$$

As there is no information about θ, $\rho(\theta)$ cannot be determinined. Therefore only $\rho(z|\theta)$ needs to be maximised.

In short Marginal Likelihood Estimation seeks to find tuning parameters that gives the highest likelihood for the measured signal. Then it finds the most likely state values given measurements and θ.

2.4.5 Optimisation methods

Bayesian methods seek to maximise the likelihood that the produced estimate is correct, using the given measurements. [6]

To this end cost functions are used, derived from PDFs based on what data is available and which parameters are to be estimated. This will be dealt with in the next section.

To get $\hat{x}_{k+1}$ closer to x than $\hat{x}_k$ using z_{k+1}, an update law is needed. As cost functions J are evaluated, this can be viewed as an optimization problem.[8]

$$\hat{x}_{k+1} = min_x J \tag{2.43}$$

This needs to be done iterativly. A common and effective method is gradient decent. Generally it has the form

$$\hat{x_{k+1}} = \hat{x_k} - K_J \frac{\epsilon}{\epsilon x} J \tag{2.44}$$

Where K_J is a gain. Ideally,

$$\hat{x_{k+1}} - \hat{x_k} \to 0,\ \hat{x} \to x,\ ask \to \inf \tag{2.45}$$

It is however difficult to set K_J. If set to low convergance might be too slow, set to high $\hat{x}$ might occilate around the optimum.

Newton-Ralphson Gain

The states can be described as

$$x = \hat{x} + \Delta \tag{2.46}$$

If $J(x)$ is expanded it will result in

$$J(x) = J(\hat{x}) + \frac{\delta^2}{\delta^2 x} J(\hat{x}) + \frac{\delta^2}{\delta^2 x} J(\hat{x})\Delta \tag{2.47}$$

The higher order terms are assumed small enough to be ignored. At optimum the cost for the true states will equal the cost of the estimated.

$$0 = \frac{\delta^2}{\delta^2 x} J(\hat{x}) + \frac{\delta^2}{\delta^2 x} J(\hat{x})\Delta \tag{2.48}$$

Solving for Δ gives

$$\Delta_{k+1} = (\frac{\delta^2}{\delta^2 x} J(\hat{x}_{k+1}))^{-1} \frac{\delta^2}{\delta^2 x} J(\hat{x}_{k+1}) \tag{2.49}$$

$$x_k \hat{+} 1 = \hat{x_k} + \Delta_{k+1} \tag{2.50}$$

The gain can be defined. [7]

$$K_J = (\frac{\delta^2}{\delta^2 x} J(\hat{x}_{k+1}))^{-1} \tag{2.51}$$

2.5 Sources of error

2.5.1 Observability

Defining the error between the states and the estimates$\tilde{x} = x - \hat{x}$, the system can be rewritten as

$$\tilde{x}_k = \phi \tilde{x}_{k-1} + v \tag{2.52}$$

$$\nu_k = H\tilde{x}_k + w \tag{2.53}$$

Basic control theory states that the oberservation matrix needs to have full rank for Q to be fully oberservable.

Definition 1: The system (2.5) is observable if

$$\mathcal{O} = \begin{bmatrix} H \\ H\phi \\ \dots H\phi^{n-1} \end{bmatrix} \tag{2.54}$$

has full rank. Systems where ϕ change might only be observable sometimes. In the the model used in this thesis, Q is observable for the assumptions made.

This means that the error created by the process noise can be determined from the measurements themselves. Which indicate that the process nosise covariance can itself be estimated.

2.5.2 Estimability

The estimability property say if the states of a system in state-space form, can be estimated.

The formal criteria for estimability is that the aposteriori estimate covariance P_k is lower than the apriori estimate coariance P_k^- [9].

Definition 2: The system (2.5) is estimable if

$$P_k < P_k^-, \qquad for\ \forall k \geq n, \qquad P_{k-1}^-, P_k \in \boldsymbol{R}^n \tag{2.55}$$

iff

$$g^T P_K g < g^T P_k^- g, \qquad for\ \forall k \geq n, \qquad g \neq 0 \in \boldsymbol{R}^n \tag{2.56}$$

As the HiPAP model already is in use, it is supposed that the system is estimable. In this thesis wether or not an adaptive filter using covariance matching remains estimable.

If an estimate of the covariances R and or Q is used, the states must continue to be estimable.

Inserting equations from the Kalman Filter to expand (2.55) results in

$$(\boldsymbol{I} - K_k H)P_k^- < P_k^- \tag{2.57}$$
$$(\boldsymbol{I} - P_k^- H^T C_\nu^{-1} H) < \boldsymbol{I} \tag{2.58}$$
$$(\boldsymbol{I} - P_k^- H^T (H P_k^- H^T + R)^{-1} H) < \boldsymbol{I} \tag{2.59}$$
$$\tag{2.60}$$

For clarity a one dimentional system is considered with $H = [1]$

$$\boldsymbol{I} - K_k H = \boldsymbol{I} - \frac{P_k^-}{P_k^- + R} \tag{2.61}$$

Looking first at the edge case where $R = 0$. When

$$\hat{R} \to 0 \tag{2.62}$$
$$K_k \to \boldsymbol{I} \tag{2.63}$$
$$P_k \to 0 \tag{2.64}$$
$$P_k^- = Q \tag{2.65}$$

Here the estimable properiy still hold, although it is assumed that all the errors result from the proccess noise.

If R should be negative, it is seen that

$$\hat{R} < 0 \rightarrow \frac{P_k^-}{P_k^- + R} > \boldsymbol{I} \tag{2.66}$$

$$\rightarrow P_k < 0 \tag{2.67}$$

$$\rightarrow P_{k+1}^- < 0 \tag{2.68}$$

$$\rightarrow \boldsymbol{I} - K_{k+1}H < 0 \tag{2.69}$$

$$\rightarrow 0 < P_k \not< P_{k+1}^- \tag{2.70}$$

$$\tag{2.71}$$

Meaning the difinition of estimability (2.55) no longer holds.

Here it has been assumed that Q is not so big as to make P_{k+1}^- positive. If that is the case, P and P^- will become more negative with each iteration, until the definition no longer holds.

If $\hat{Q} = 0, P$ will converge towards zero and P^- converge to Q. The kalman filter gain will go towards 1 and the measurements relied upon intierly.

Should $\hat{Q}$ be nagative, P will become negative, and a similar inconsistency to (2.62) will occur.

As covariances are positive semidifinite by definition this is hardly suprising, but this shows that the estimated covariances must be positive semidefinite.

Further remarks can be found in [9].

2.5.3 Concurrent R and Q estimation

R and Q cannot be estimated in the same step. As it has been mentioned in [10], they have to be estimated in alternate steps. This is because the effects of the change in R is not completly seen before the next step. This is clear from the equations used in Q estimation.

$$\hat{x}_k^- \tag{2.72}$$

$$= \phi(x_{k-1}^- + K_{k-1}\nu_{k-1}) \tag{2.73}$$

$$= \phi(x_{k-1}^- + (P_{k-1}^- H^T (H P_{k-1}^- H + \hat{R}_{k-1})\nu_{k-1}) \tag{2.74}$$

$$\hat{x}_k = x_k^- + (P_k^- H^T (H P_k^- H + \hat{R}_k)\nu_k \tag{2.75}$$

Information about Q comes from the measurements, but this assumes R is consistent. When $\hat{R}_k \neq \hat{R}_{k-1}$, the the Q estimate will be corrputed by the change in $\hat{R}$.

Do avoid this, R and Q are estimated on alternate steps.

Since both estimates are based on the measurements in some way, there is no guarantee that they will converge to the correct value.

2.5.4 Effect of badly tuned filter

These methods assumes the other filter parameters are known and correct. If they are not, a constant error is produced in the estimate. For instance, if the set Q is incorrect while estimating R.

$$z_k = Hx_k + v_k \tag{2.76}$$
$$= H\phi x_{k-1} + Hw_k + v_k \tag{2.77}$$

This shows that Q enters into the estimation of R.

$$C_\nu = HP_{k-1}^- H^T + R \qquad = H\phi P_{k-1}^- \phi^T H^T + HQH^T + R \tag{2.78}$$

This is well and good should Q be known. If unknown this cause some problems however. If the set Q has an error, it can be described as $Q = Q_T + \epsilon_Q$. Then

$$C_\nu = H\phi P_{k-1}\phi^T H^T + HQ_{true}H^T + H\epsilon_Q H^T + R \tag{2.79}$$

Defining $\hat{R}_e$ as the R estimate with a wrongly set Q, leads to

$$\hat{R}_e = \hat{C}_\nu - H\phi P_{k-1}\phi^T H^T + HQ_{true}H^T + H\epsilon_Q H^T \tag{2.80}$$
$$= \hat{R} + H\epsilon_Q H^T \tag{2.81}$$

It might not appear to have much of an impact, but it will affect the position error estimate. Considering that methods using J^S matches a quantity given by P, Q, and R, to an observed one, and that a good position estimate depends on a ratio between R and Q,

All the covariance matching methods described will have this problem, where the estimated R contains an error from Q, and have no good way reduce it without also estimating Q.

2.5.5 R/Q ratio impact on positional error

When most students first fiddle with KFs they are told that the ratio between R and Q is more important than getting them correct. This is a simplification, but true if only the state error is concidered.

Using the KF equations gives the state estimate error one step backwards as

$$|x_k - \hat{x}_k| = |x_k - (x_k^- + K_k\nu_k)| \tag{2.82}$$

$$= |x_k - (x_k^- + P_k^- H^T S^{-1}\nu_k)| \tag{2.83}$$

$$= |x_k - (x_k^- + (\phi P_{k-1}\phi^T + Q)H^T(H\phi P_{k-1}\phi^T H^T + HQH^T + R)^{-1}\nu_k)| \tag{2.84}$$

$$= |\ x_k - (x_k^- + (\phi P_{k-1}\phi^T + Q_T + \epsilon_Q)H^T \tag{2.85}$$

$$(H\phi P_{k-1}\phi^T H^T + H(Q_T + \epsilon_Q)H^T + R_T + \epsilon_R)^{-1}\nu_k)\ | \tag{2.86}$$

Here $Q = Q_T + \epsilon_Q$ and $R = R_T + \epsilon_R$, meaning that the filters set R Q are the sum of the true value and some error. If the filter has the correct R, $\epsilon_R = 0$. Looking at the KF first steps the state estimate error is

$$|x_1 - (x_1^- + \frac{(\phi P_0\phi^T + Q_T + \epsilon_{Q,1})H^T}{H\phi P_0\phi^T H^T + H(Q_T + \epsilon_{Q,1})H^T + R_T + \epsilon_{R,1}}\nu_1)| \tag{2.87}$$

Collecting the terms containing R, $Q, \hat{x}_0$, and P_0 into $\tilde{x}^-$, N_{RQ}, and D_{RQ} result in.

$$|\tilde{x}^- - \frac{N_{RQ} + \epsilon_{Q,1}H^T}{D_{RQ} + H\epsilon_{Q,1}H^T + \epsilon_{R,1}}\nu_1| \tag{2.88}$$

Refering to the EKF equations (2.9)-(2.15), it can be shown that the terms N_{RQ} and D_{RQ} become less important for later steps. Which means that the absolute state error can be assumed to be defined solely by the ratio between Q and R. The full equations are lengthy and complex, and because of their tangential relevance, has been left out.

CHAPTER 3

ALGORITHMS

3.1 Adaptive Limited Memory Filter (ALMF)

The ALMF algorithm is a covariance matching scheme that uses the average of the last N steps to produce the estimates. [11]

As outlined in subsection 2.4.1, ALMF calculate a covariance from measurements and subtract the apropriate filter parameters to produce an estimate.

Rather than recalculating the mean each iteration, it updates according to the difference between the incomming and outgoing values. This is attractive because it makes the algorithm computationally light.

Algorithm

It does not assume zero-mean for the innovations, so it uses an unbiased innovation covariance estimate.

$$\hat{C}_R = \frac{1}{N-1}\sum_{i=0}^{N}[(\nu_{k-i} - \bar{\nu})(\nu_{k-i} - \bar{\nu})^T \tag{3.1}$$

The following procedure is for estimating R. Estimating Q is done the same way, but with the apropriate variables.

It is initialised with a full LIFO stacks to handle the innovations and $HP^{-}H^{T}$, mean innovation ν_0, mean innovation covariance $\hat{C}_0$, and starting estimate $\hat{R}_0$. When a new measurement is recieved, ν_k and $HP_k^{-}H^{T}$ is pushed into their respective stacks. ν_{k-N} and $HP_{k-N}^{-}H^{T}$ is popped out.

The innovation mean is then updated.

$$\bar{\nu}_k = \bar{\nu}_{k-1} + \frac{1}{N}(\nu_k - \nu_{k-N}) \tag{3.2}$$

The estimate is then updated

$$\hat{R}_k = \hat{R}_{k-1} + \frac{1}{N-1}[(\nu_k - \bar{\nu}_k)^2 - (\nu_{k-N} - \bar{\nu}_k)^2 \tag{3.3}$$

$$+ \frac{1}{N}(\nu_k - \nu_{k-N})^2 - \frac{N-1}{N}(HP_k^{-}H^{T} \; - HP_{k-N}^{-}H^{T}) \tag{3.4}$$

It is clear that this is equal to

$$\hat{R}_k = \frac{1}{N-1}\sum_{i=0}^{N}[(\nu_{k-i} - \bar{\nu})(\nu_{k-i} - \bar{\nu})^{T}] - \frac{1}{N}\sum_{i=0}^{N}[HP_iH^{T}] \tag{3.5}$$

$$= \hat{C}_\nu - H\bar{P^{-}}H^{T} \tag{3.6}$$

Algorithm 3.1: ALMF algorithm for estimating R

Parameters: $\boldsymbol{\nu}$: Innovation stack.
$\boldsymbol{HP^-H^T}$: Filter parameter stack.
N: Size of stacks.

Input: ν_k: Innovation.
$HP_k^-H^T$: Filter parameters.

Output: $\hat{R}_k$: Estimate

$\nu_k \rightarrow \boldsymbol{\nu}$ Add latest innovation to stack
$HP_k^-H^T \rightarrow \boldsymbol{HP^-H^T}$ Add latest filter parameters to stack
if $k > N$ **then**
    $\nu_{k-N} \leftarrow \boldsymbol{\nu}$ Take out oldest innovation from stack
    $HP_{k-N}^-H^T \leftarrow \boldsymbol{HP^-H^T}$ Take out oldest filter parameters from stack
    Update $\bar{\nu}$ using eq. (3.2)
    Produce $\hat{R}_k$ from eq. (3.3)
else if $k = N$ **then**
    Calculate $\bar{\nu}_0$ from stack $\boldsymbol{\nu}$
    Calculate $\hat{R}_0$ with eq. (3.5)
else
    $\hat{R}_k = \hat{R}_0$
end

Design Parameters

ALMF has three design parameters, the initial estimate $\hat{R}_0$, initial mean innovation $\bar{\nu}_0$, initial state estimate covariance $\bar{P_0^-}$, and the size N of the stacks keeping the inputs.

$\bar{\nu}_0$ and $\bar{P_0^-}$ needs to be set close to their correct value, as not doing so result in a systematic error. When implementing the ALMF algorithm, they are set to the mean of the stacks when first filled. This ensure that they adhere to the overservations.

$\hat{R}_0$ needs to be close to the observed value to avoid constant error. It has been set by eq. 3.5 when the stack when first filled, and $\bar{\nu}_0$ and $\bar{P_0^-}$ have been calculated.

The stack or sample size N determine the ability of the estimate to react to change, and impact the estimates variance.

With smaller N, the estimate reacts quicker, but the its variance will be higher as the noise has a stronger impact. This can be seen in fig. 3.1.

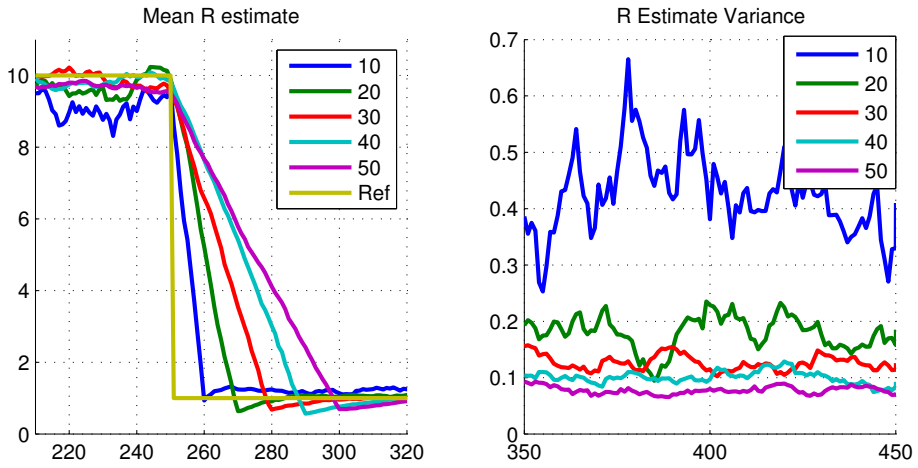

FIGURE 3.1: **Left:** Mean ALMF R estimate over 100 simulations for varying sample sizes N. The correct R goes from 10 to 1 at step 250.
Right: R estimate error variance $var(R_k - \hat{R}_k)$ for different sample sizes N. R is kept constant at 1.

Difficulties

There are some difficulties.

ALMF is sensitive to initial values. Since the difference between incomming and outgoing values is used, should the initial values be wrong, a constant error is introduced.

The implementation used lessened this problem by filling the stacks, and starting with the mean value of the stacks.

Furthermore, there is not a guarantee that $\hat{R}$ will be positive definite. It has been recommend resetting the estimate by taking the absolute value of the diagonal of the estimate each iteration [11]. This will cause different problems in turn.

The R estimate will fluctuate around the percieved correct value. So if the target R is percieved low it is quite likely $\hat{R}_k$ will be negative definite for some k. This will happen more often when Q is set too high, as it lowers the estimated R without lowering the estimation variance. This means that the estimate will be closer to zero, while it fluctuate more. If Q is set correct, and R is close to zero, this will happen less frequently as the estimation variance is related to the size of R.

Each time $\hat{R}$ is not positive definite $\hat{R}_k = |\hat{R}_k^-|$.

Should $\hat{R}_k^-$ be negative, an error will be introduced to mean estimate.

$$\epsilon_{R,k} = \hat{R}_k - \hat{R}_k^- . \tag{3.7}$$

Since the algorithm only uses the difference of the first and last values in the stack, it has no way to compensate for this error in future steps. This causes the algorithm to be inconsistent in some cases.

Since ALMF is included for comparison no attempt has been made to remedy this problem.

Some solutions readily appear, such as resetting the filter, setting a minimum value and recaulculating the mean values regularly. Or only returning the absolute value, while keeping the negative internaly for the next iteration.

3.2 Sage-Husa (SH)

The Sage-Husa algorithm [12] is also a covariance matching scheme. Mainly used for estimating covariances of IMUs during initialization, it assumes static conditions [10]. It is included to compare to ALMF, and because it has desirable characteristics. Namely quick initial convergance, high accuracy once settled, and that it does not have the same instability problems.

Algorithm

SH averages using a forgetting factor which weights the input and the previous estimate. It require less memory than ALMF, while also beeing slightly less computationally intensive.

The forgetting factor b define the weight d which is given to the input each iteration. d decrease each iteration, converging on a value determined from b.

$$d = \frac{1-b}{1-b^i} \tag{3.8}$$

i being the step number.

The estimates are produced from the incomming innovation ν_k, the current filter parameters $HP_k^-H^T$, and the previous estimate $\hat{R}_{k-1}$.

$$\hat{R}_k = (1-d)\hat{R}_k + d(\nu_k\nu_k^T - HP_k^-H^T) \tag{3.9}$$

Algorithm 3.2: Sage-Husa algorithm for estimating R

Parameters: i (Step counter)
b (Forgetting factor)
Input: ν_k (Innovation)
$HP_k^- H^T$ (Filter parameter)
Output: $\hat{R}_k$ (Estimate)
1 $d = (1-b)/(1-b^i)$
2 $\hat{R}_k = (1-d)\hat{R}_{k-1} + d(\nu_k \nu_k^T - HP_k^- H^T)$
3 Increment step counter i.

Design Parameters

The only design parameter is the forgetting factor b. It determine the weight that is given the input, and therefore will affect the ability to estimate a moving target, as well as the estimates variance. This is shown in fig. 3.2

b is reccomended to be set between 0.95 and 0.99, but must be between 0 and 1. It has been noted that the forgetting factor might be chosen dynamicly [10]. To reduce complexity b has been left constant at 0.99.

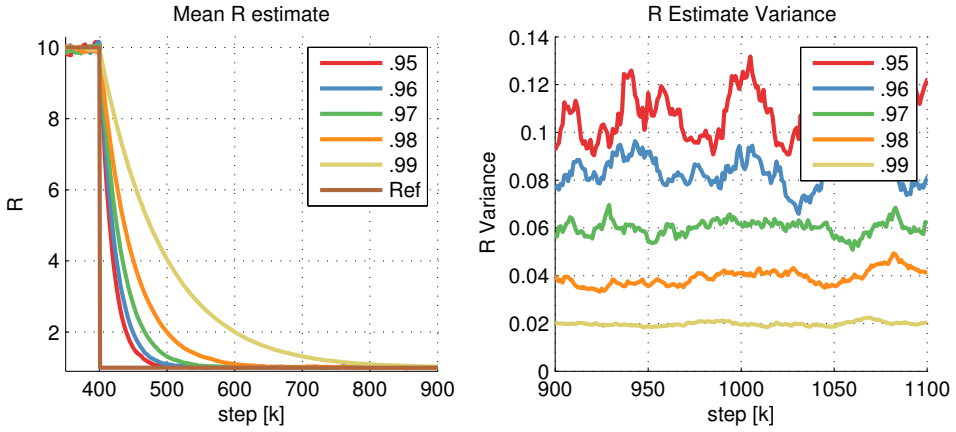

FIGURE 3.2: Sage-Husa estimate step response (left), and estimate variance under static conditions (right). R is estimated with different $b \in [.95, .99]$. The values shown are the average of a Monte Carlo simulation with 500 runs.

Difficulties

It is not given SH produce a good estimate initially, meaning the estimated covariance might not be positive definite. To subvert this the diagonal elements can be reset to their absolute value. Unlike in ALMF this does not cause a systematic error, as the estimate is not derived from the change in measurements, but the measurment themselves.

SH not handle quickly changing targets well. Apart from slow convergance, a dramatic change can introduce a error in the estimate. Because of the forgetting factor gives exponencially lower weight to previous input, they are never discarded. Meaning that after a change, the older estimates are still used which creates an error.

To circumvent, the user might reset the filter, thereby removing the error.

3.3 TANIS estimator (TANIS)

The TANIS estimator is a covariance matching method, that is based on filter consistency rather than mathcing directly.

Algorithm

It uses hypothesis testing to detect filter inconsistency compared to observed measurements. Two simple hypothesis are tested.

$\mathcal{H}_0$: $\xi(\boldsymbol{\nu}, C_{\nu,k}) < {\chi^2}_{min} \rightarrow$ Filter parameters are too high

$\mathcal{H}_1$: $\xi(\boldsymbol{\nu}, C_{\nu,k}) > {\chi^2}_{max} \rightarrow$ Filter parameters are too low

At first glance this might seem counterintuitive, but remembering that $\xi(\boldsymbol{\nu})$ (2.17) uses the inverse of the filter parameters should clear misgivings.

Assuming that Q is set correctly, only R can be changed to modify the filter. (See eq. (2.16)). To this end a parameter α is introduced to scale R to an appropriate size.

$$\hat{R}_k = \alpha R_0 \tag{3.10}$$

R_0 is the filters initial R.

If $\mathcal{H}_1$ holds α needs to be increased to remedy the inconsistency. Similarly if $\mathcal{H}_0$ holds α needs to be decreased. The two hypothesis are mutually exclucive. Should both fail, the filter is deemed consistent, and α remain as is.

How α changes is of importance. If it the estimate is continuous it is suseptible to the high estimate variance vs. high accuracy trade-off. As an estimate that follows change quickly while keeping variance low, is wanted more than accuracy, α is multiplied and divided by a constant when neccesary.

This will reduce the accuracy of the estimate. It was felt that addition and subtraction were too slow for a fixed step length, and complicating the matter unneccecarily if step length were to be chosen dynamicly.

Algorithm 3.3: TANIS algorithm for estimating R

Input: $\nu_k, HP_k^- H^T$
Output: $\hat{R}_k$
$\nu_k \rightarrow \boldsymbol{\nu}$ Add latest innovation to stack
$\nu_{k-N} \leftarrow \boldsymbol{\nu}$ Remove oldest innovation from stack
Get χ_k^2 from eq.(2.17)
if $\mathcal{H}_0$ **then**
    $\alpha_k = \frac{1}{2}\alpha_{k-1}$;
else if $\mathcal{H}_1$ **then**
    $\alpha_k = 2\alpha_{k-1}$;
else
    $\alpha_k = \alpha_{k-1}$;
end
$\hat{R}_k = \alpha_k R_0$

Design Parameters

As α has the form $\alpha = \alpha_0^n$, the choise of α_0 determine the amount the estimate change when deemed neccecary. This means that the estimate $\hat{R}$ closer match the target for lower values of α_0. However, as the estimate is changed by in- or decrementing n, a lower value for α_0 will lessen the estimates ability to track a changing target. If set too high on the other hand, will cause bad estimates. $\alpha_0 = 2$ was chosen, and has proven sufficient.

Another tuning parameter is sample size. As the test value ξ is the average NIS of N samples, any change will not be detected instantly. As with ALMF, larger N causes the estimate to change slower, while reducing the probability of false positives. Which in turn reduce the estimates variance.

A signifigance region needs to be chosen for the consistency test. It determine how unlikely the filter parameters, given the innovations, have to be for the hypothesis to pass. A smaller significance region causes fewer false positives, but the estimates will converge slower.

Problems

Defining a space that contains the target R as $\Omega_R = [0, \text{inf}]$. The estimator will only give estimates that are in the space $\Omega_{\hat{R}} = [(\alpha^n R_0), n \in [n_{min}, n_{max}]]$. It is clear that Ω_R is larger than $\Omega_{\hat{R}}$. This means the TANIS estimator searches a partitioned space, setteling at the partition which is closest to the target, rather than at the target itself. The underlying assumption is that the exact value is not needed, just that the estimate is closer to the target.

3.4 Ratio Balancer (β-MLE)

As stated in subsection 2.4.1 and later shown in section 5.1.3, $\hat{R}$ will be wrong if Q is not set correctly. This will make the ratio between Q and R wrong inturn, possibly worsen the state estimates error (see subsec. 2.5.5).

The following method seeks to rebalance this ratio by a sequential MLE. This means that this method sole use is finding a fitting Q while R is estimated.

Alogrithm

The filters Q must be estimated to ensure that it is closer to the true Q to get a better etstimate.

To this end a variable β is introduced, producing the estimated Q.

$$\hat{Q}_k = Q_0 + \beta I \tag{3.11}$$

The estimation methods hitertoo dicussed produce the estimate $\hat{R}$ by matching the filter parameters S with observed innovation covariance $\hat{C}_\nu$. Since S is given by R, P, and Q, β is subtracted from R to maintain the size of S resulting from the R estimation done.

Altogether β is introduced thusly.

$$\hat{Q}_k = Q_0 + \beta_k \boldsymbol{I} \tag{3.12}$$
$$\hat{R}_k^* = \hat{R}_k - \beta H H^T \tag{3.13}$$
$$\tag{3.14}$$

Defining C_ν^* as C_ν using $\hat{R}_k^*$ and $\hat{Q}_k$ in (2.13).

$$\begin{aligned}
C_{\nu,k}^* &= H P_k^- H^T + \hat{R}_k^* && (3.15)\\
&= H(\phi P_{k-1}\phi^T + \hat{Q}_k)H^T + \hat{R}_k^* && (3.16)\\
&= H\phi P_{k-1}\phi^T H^T + H\hat{Q}_k H^T + \hat{R}_k^* && (3.17)\\
&= H\phi P_{k-1}\phi^T H^T + H(Q_0 + \beta_k \boldsymbol{I})H^T + \hat{R}_k - \beta H H^T && (3.18)\\
&= H\phi P_{k-1}\phi^T H^T + H Q_0 H^T + H\beta_k \boldsymbol{I} H^T + \hat{R}_k - \beta H H^T && (3.19)\\
&= C_{\nu,k} + H\beta_k \boldsymbol{I} H^T - \beta H H^T && (3.20)\\
&= C_{\nu,k} && (3.21)
\end{aligned}$$

It is seen that the introduction of β does not change the innovation covariance used by the filter in the same step. Still, as shown later (in fig. 4.6), it is still possible to estimate Q from J^S (2.22).

To get β to appear the J^S equations (2.22) needs to be taken one step further back. Which gives

$$\nu_k^* = z_k - H\phi^2 \hat{x}_{k-2} - H\phi(\phi P_{k-2}\phi^T + \hat{Q})H^T C_{k-1}^{-1}(z_{k-1} - \phi\hat{x}_{k-2}) \tag{3.22}$$

and

$$\begin{aligned}
C_{\nu,k}^* &= H P_k^- H^T + \hat{R}_k && (3.23)\\
&= H\phi(I - (\phi P_{k-2} + \hat{Q})H^T C_{k-1}^{-1} H)(\phi P_{k-2}\phi^T + \hat{Q})\phi^T H^T + H Q_0 H^T + \hat{R} && (3.24)
\end{aligned}$$

Figure 3.3 show the J^S (first), innovation (second), innovation covariance (third), and esimate position error (forth) for a filter using the TANIS estimator and varying values of β. It is clear that J^S has a minimum for the theoretical value of $\beta = 0$

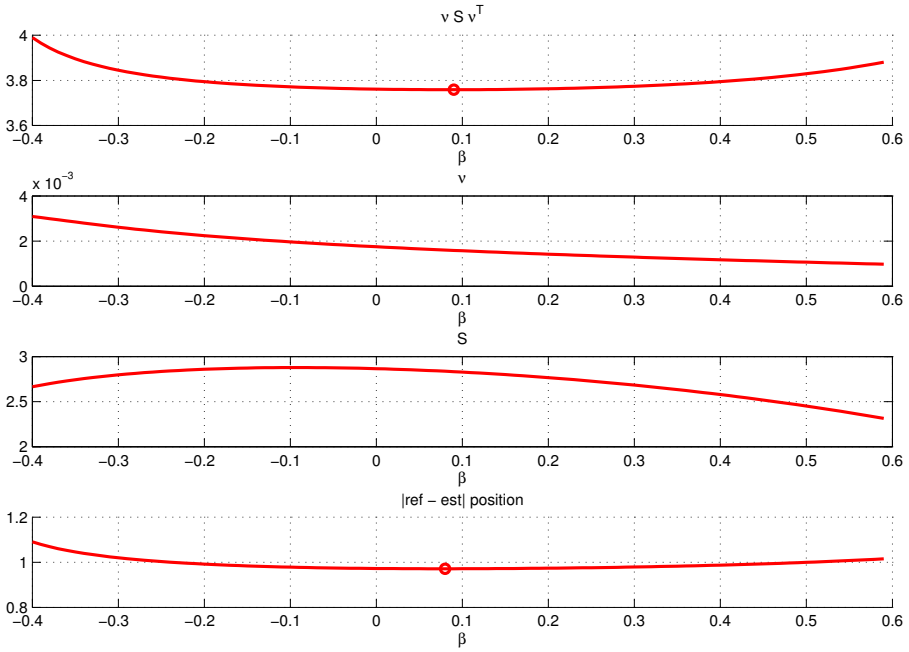

FIGURE 3.3: Cost functions ofr different values of β

Therefore it is viable to use J^S to estimate β. To get an idea on how well β can be estimated, the Cramer-Rao bound is shown in fig. 3.4. Cramer-Rao lower bound is the lowest value the state estimate variance a filter can achieve.

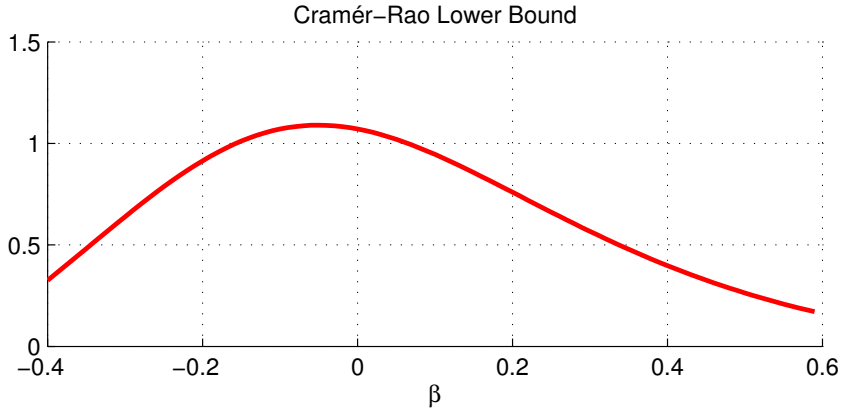

FIGURE 3.4: Cramer-Rao lower bound for different values of β

It has been noted that MLE can be done sequentially by viewing Δ (2.47) as a noisy input[6] . Where Δ is calculated from the data in each step rather than from all available measurements which MLE normally does.

As finding β now is a value that minimize the cost J^S defined as

$$J^S = \nu_k^*(C_{\nu,k}^*)^{-1}\nu_k^{*T} \tag{3.25}$$

As Δ will itself be corrupted by noise, and therefore some filtering method need to be applied. In this implementation SH was used. While some more refined method could be investigated, it was not as it soon becase appearant that the computational time and resources needed were too much for this method to be viable.

Algorithm 3.4: ALMF algorithm for estimating R

Parameters: i (Step counter)
b (Forgetting factor)

Input: ν_k, ν_{k-1} (Innovations)
$H, P_{k-2}, \phi, C_{k-1}^{-1}$ (Filter parameter)
$\hat{x}_{k-2}$ (Previous Estimate)

Output: β_k (Estimate)

1 $d = (1-b)/(1-b^i)$
2 Calculate the gradient $\mathcal{G}$ of J^S (3.25)
3 Calculate the Hessian $\mathcal{H}$ of J^S (3.25)
4 $\beta_k = (1-d)\beta_{k-1} + d(\mathcal{H}^{-1}\mathcal{G})$

Design Parameters

The design choise to be made here is how to smooth the incomming input to β. And the design parameters will therefore differ. Here SH has been chosen.

Difficulties

Eventhough this is a suboptimal sequential MLE method evaluating only the last two steps, meaning that this is not garantee to give the most likely estimate given previous data, it still rely on calculating the gradient and hessian. This is computationally heavy, to such an extent it might prove unsuited. The speed will greatly depend on the implementation, when implementing this method the Matlab functions *gradient()* and *hessian()* were used.

3.5 TANIS / Sage-Husa Hybrid Estimator (hybrid)

The TANIS/Sage-Husa hybrid estimator seeks to combine the rapid convergance of the TANIS estimator with the accuracy of Sage-Husa.

Algorithm

To combine the estimate $\hat{R}$ is defined as

$$\hat{R} = \alpha_k R_0 + \Delta_k \tag{3.26}$$

α_k is the same as described in subsection 3.3. The step length α_0 has again been set to 2.

SH has been modified to produce the difference between $\alpha_k R_0$ and the observed R.

$$\Delta_k = (1-d)\Delta_{k-1} + d(\nu_k \nu_k^T - HP_k^- H - \alpha_k R_0) \tag{3.27}$$

d is the same as in subsection 3.2, with $b = .99$.

Unlike the TANIS estimator the hybrid does start before the stack of innovations ν is filled. Before the stack is full, the estimate is given by SH and the TANIS hypothesis testing is halted.

Each iteration it adds the latest innovation ν_k to the sample stack, and removing the oldest innovation ν_{k-N}. A consistency test then determine if the current α and filter parameter fit. The same hypothesis as the TANIS estimator is used. (See sec. 3.3.)

If neither hypothesis pass, SH is used to update Δ.

$\mathcal{H}\prime$ or $\mathcal{H}_\infty$ passing indcate that the filter is inconsistent and the estimate must be changed. α is then changed as described in section 3.3. And Δ_R and k is reset to 0 and 15 respectivly.

Algorithm 3.5: TANIS/Sage-Husa Hybrid algorithm for estimating R

Data: ν_k, P_k, H
Result: $\hat{R}$
$\nu_k \rightarrow \boldsymbol{\nu}$ Add latest innovation to stack
if $\mathcal{H}_0$ **then**
    $\alpha_k = \frac{1}{2}\alpha_{k-1}$;
    $\Delta_k = 0$;
    $i = 15$;
else if $\mathcal{H}_1$ **then**
    $\alpha_k = 2\alpha_{k-1}$;
    $\Delta_k = 0$;
    $i = 15$;
else
    $\alpha_k = \alpha_{k-1}$;
    $d = \frac{1-b}{1-b^i}$;
    $\Delta_k = (1-d)\Delta_{k-1} + d(\nu_k\nu_k^T - HP_k^-H^T - \alpha_k R_0)$;
    Increment i;
end
$\hat{R}_k = \alpha_k R_0 + \Delta_{R_k}$

Design Parameters

There is a number of design parameters. Like the TANIS estimator is has a significance level, an innovation stack of size N, and a step size α_0. The dicription in sec. 3.3 is valid for the hybrid as well.

From SH a forgetting factor b is used.

Since Δ and i is reset whenever a hypothesis pass, it needs to be reset to a value i_0. i is not reset to 1 for two reasons. αR_0 is an estimate of R in itself, indicating that $R \in [2^{n-1}R_0, 2^{n+1}R_0]$. This mean that when $\hat{R}$ is changed by the consistency test, it is closer to the target than for any other α. Therefore if $k = 0$ that information is discarded. SH also has a higher variance during the first steps, as shown in fig. 3.5. This might push the estimate past the next α step. Resetting k to 15 were found to be a good value in this case.

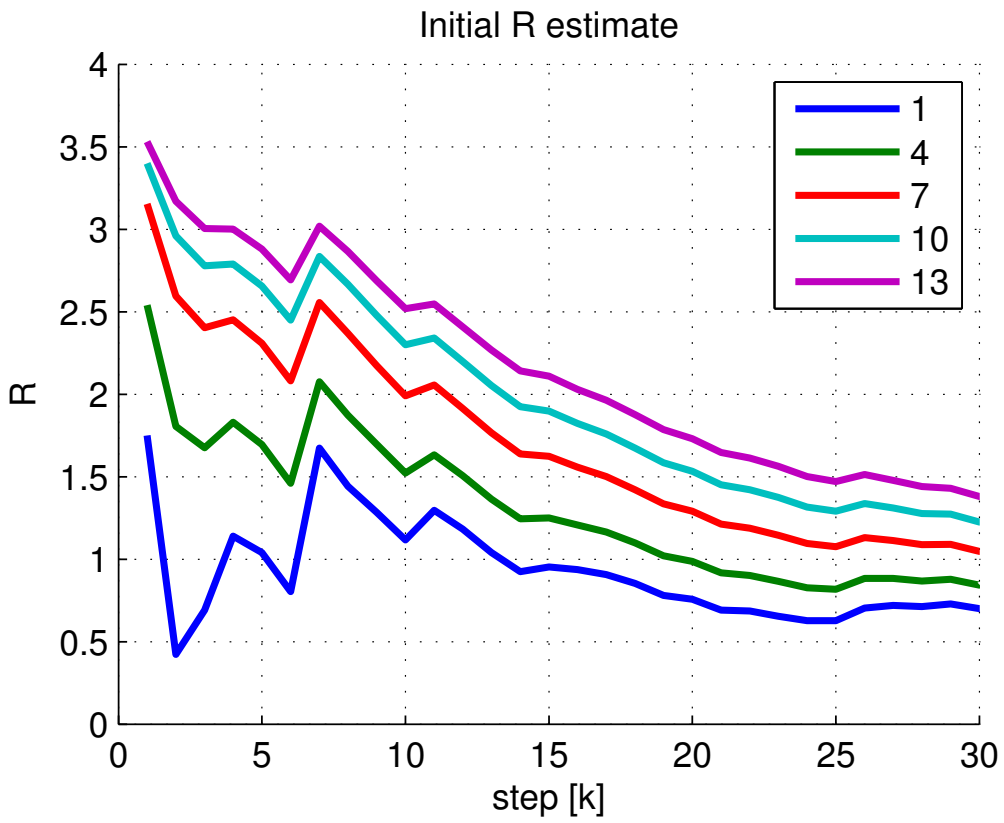

FIGURE 3.5: First R SH estimates for different starting values of k.

Difficulties

The overestimation problem TANIS has will be circumvented by the addition of SH. While the slow convergance and possible error in Δ is no longer a problem, as the SH part of the hybrid is reset each time the consistency test fail.

As Δ is added to αR_0, the estimate cannot be garanteed positive definite. To make sure the estimate is set to a lower limit of 0.001

CHAPTER 4

PRELIMINARY SIMULATIONS

Following are plots of different cost functiones, as described in section 2.3. These show the average cost for differently set tuning parameters. The costs were collected from a single measurement signal of 5000 steps generated from the HiPAP approximation described in section 2.1. Only one dimention was considered, meaning that $R \in \boldsymbol{R}^1$ and $Q \in \boldsymbol{R}^2$. The Q matrix has the form

$$Q_0 = \begin{bmatrix} Q(1,1) & 0 \\ 0 & Q(2,2) \end{bmatrix} \tag{4.1}$$

$$Q(1,1) = Q(2,2) \tag{4.2}$$

Since the average costs have been calculated from a finite number of samples, it will not neccecarily have to correct minimum, but will nonetheless give a good indication. The lower plots show an average of a 10 sample section of the signal. This shows the amount of information gained in a few steps, which indicate possibilities for estimating quickly. It also shows how the impact of noise, as the minimum does not fall at the theoretical optimum.

A log10-scale has been used for both axis, and give the value of each element in the matrices.

4.1 EKF performance

As getting good estimates are of the first importance, the ability of the filter needs to be shown.

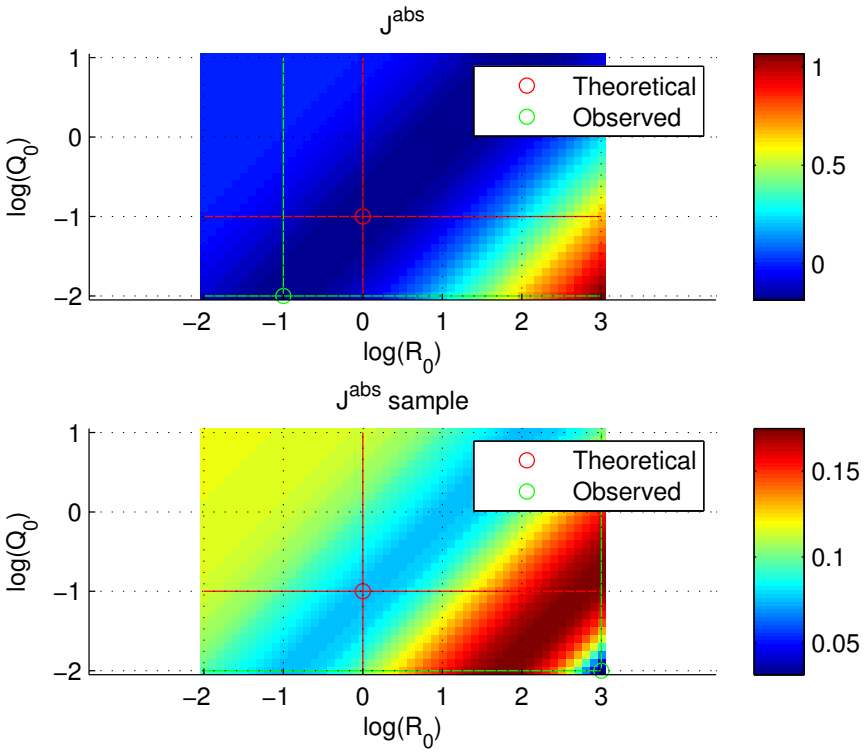

FIGURE 4.1: Colourmap showing average absolute state estimation error (J^{abs}(2.18)) for different tuning parameters. Lower plot show average of 10 steps.

Looking only at the estimate error J^{abs}, it is clear from fig. 4.1 that the R/Q ratio determine the accuracy of the estimates.

J^{abs} does not take the state estimate variance into consideration. Since a good filter produces estimates that are 'trustworthy', the covariance P also needs to be considered. J^P is therefore a better measure on how well a filter performes.

There are several things that can be seen in fig. 4.2.

It is clear that a well tuned filter performs better. But there is a region where the difference in performance is negligible. In fig. 4.2, as well as the other colourmaps, this is the darkest blue area. This minimum area is where the shows where covariance estimates aim for, and in the case of J^P, which filter parameters give the best result.

In fig. 4.2 the minimum area is slightly higher than the theoretical optimum for both R and Q. As mentioned previously, this is because the cost has not been calculated from an infinite number of samples.

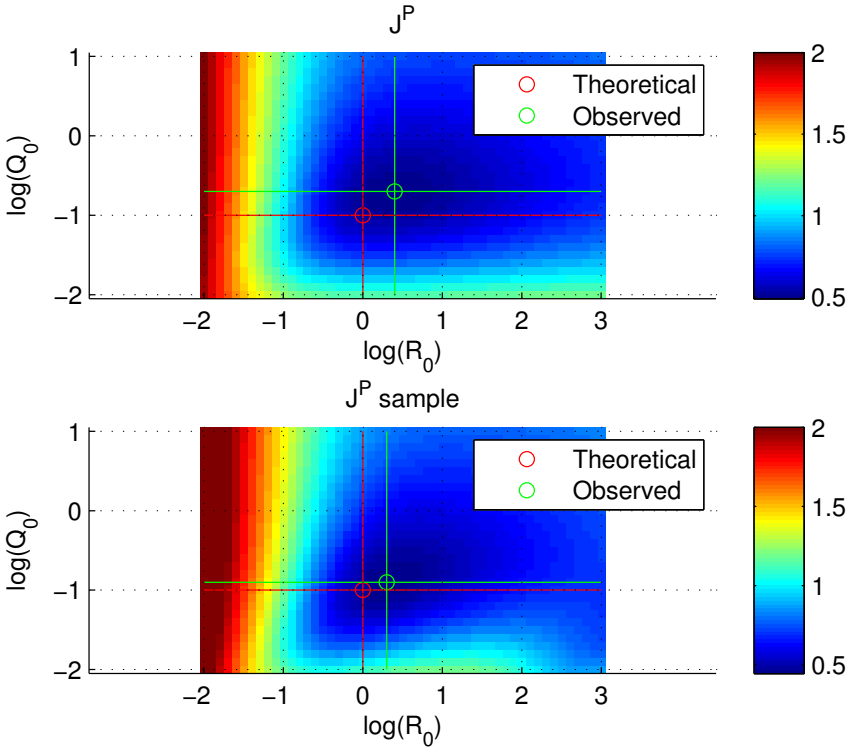

FIGURE 4.2: Colourmap showing average state estimation error (J^P(2.20)) for different tuning parameters. Lower plot show average of 10 steps.

The shape and size of J^P minimum region shows which parameter is more important to improve performance. It can be seen that the filter is slightly more sensitive to Q, in that the minimum region is smaller along the Y-axis. This means that R can be set less accuratly than Q, while still resulting in a close to optimal filter. It should be noted that this difference is slight.

The cost sharply increases for lower values of R and Q. Meaning that it is better that they be set too high, than too low. If set to half below optimum has the same cost as a 100 times too high.

The lower plot of fig. 4.2 shows the same characteristics as the upper. The minimum region is more elongated long the diagonal, again showing the importance of a correct R/Q ratio. The contours are less defined, but that just shows that less information give less clear indicators.

4.2 R estimation possibility

Having established that the filter works better when using close-to-correct tuning parameters, the possibility of estimating those must be investigated. As the only source of information is the measurements, J^P and J^{abs} cannot be used. J^S and J^{χ^2} are still viable though, and will be evaluated below.

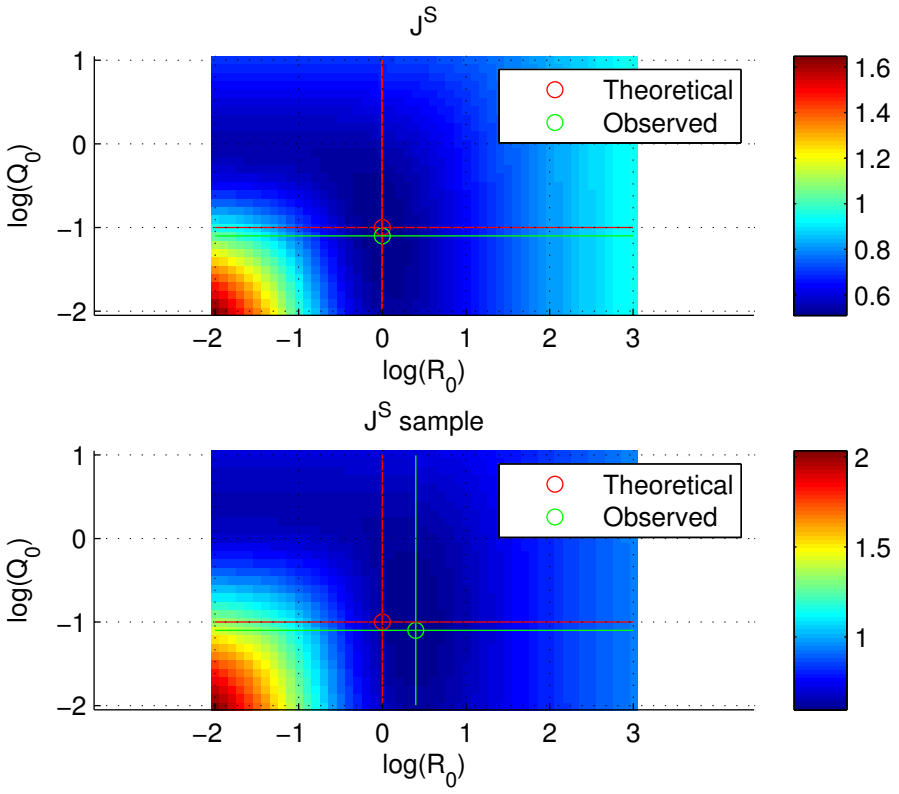

FIGURE 4.3: Colourmap showing average measurement cost (J^S(2.22)) for different tuning parameters. Lower plot show average of 10 steps.

Several things can be learned from fig. 4.3. The minimum area has a center around the target, which shows that the correct covariances can be estimated. It is however larger along the y-axis, which means Q is harder to estimate than R using J^S. Further, if approached from above the cost decrease slower than from below the target. This indicate that starting estimation at a value higher than the target will converge slower than if started from below.

There is an area close to the minimum area which is shaped like a quarter circle. This speaks to the fact that J^S will be close to minimum for a given sum of Q and R. This has been outlined in section 2.5.4. The effect of this is that the R estimate will not converge towards the correct value if Q is set incorrectly, and vice versa. For $Q(1,1) = 1$

there is no distinct minimum for R for instance. This problem worsens as Q if fixed at higher values, as the upper border for Rs minimum is pushed further away from the optimum. Meaning that if approached from above, and stopped when the cost function gradient becomes suffeciently small, the R estimate would still be too low.

The other way around is in some ways worse. If R_0 is fixed at 10^{-2} we see that a $\hat{Q}$ wil settle at 1, 10 times larger than it's true value. This is because it is the measurement innovation covariance that the target, not the noise covariances.

Comparing fig. 4.2 and fig. 4.3 shows that just because the estimates settle on the quarter circle, does not neccesarily result in a good filter.

The sample shows the same characteristics, but the minimum area is larger with a center slightly off target.

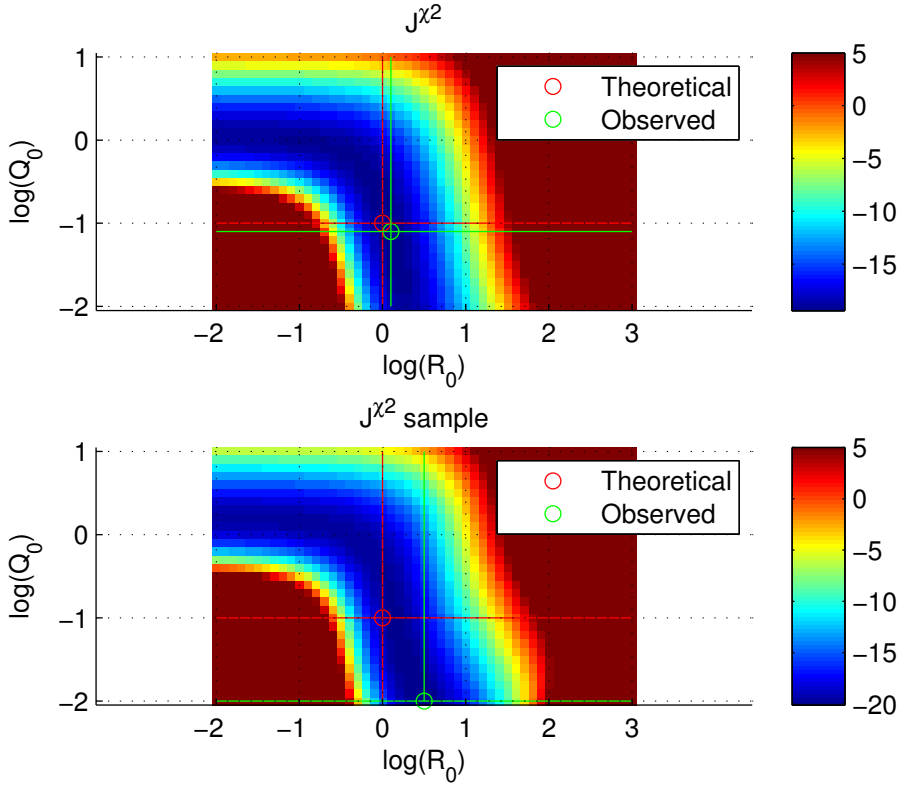

FIGURE 4.4: Colourmap showing average consistency cost (J^{χ^2}(2.24)) for different tuning parameters. Lower plot show average of 10 steps.

As in fig. 4.3 there is a region close to minimum in a quarter circle shape. It differs by being more sharply defined from both directions, as well as having a minimum region that is much larger than J^S.

The same observations made from fig. 4.6 appear in fig. 4.4 as well. Exept that estimating from above the target should converge quicker and that the minimum streches ffarther along the curve.

4.3 Impact of R estimation

Estimating R will obiously change the cost function. If the estimate is done correctly the minimum region will increase, as a wrong inital value is compensated for and produce lower cost than without estimation. The plots will also show what information remains after estimation, and the possibility of estimating Q.

It can be seen in fig. 4.5 that with R estimation, the starting value R_0 no longer have much impact. The minimum area is a line close to the correct Q_0. This is an indication that the estimating R is beneficial, atleast when Q have been set close to correct.

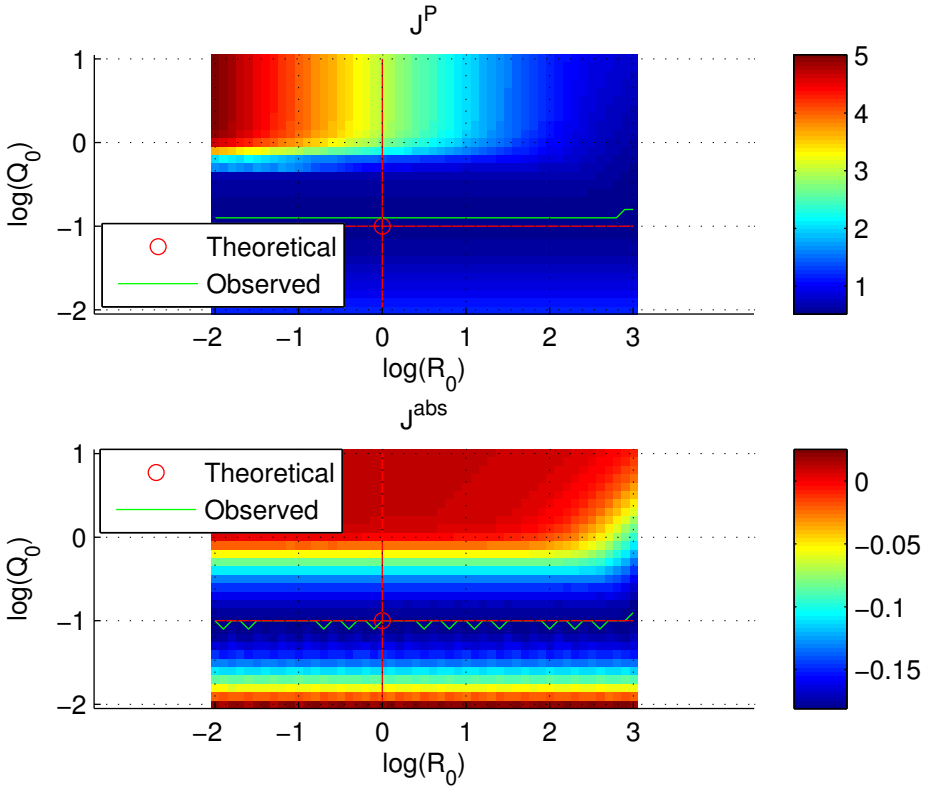

FIGURE 4.5: Colourmap showing average state estimation error (J^P(2.20), upper) and absolute state estimation error (J^{abs}(2.18), lower). Red show the theoretical minimum, green the observed minimum for each R_0.

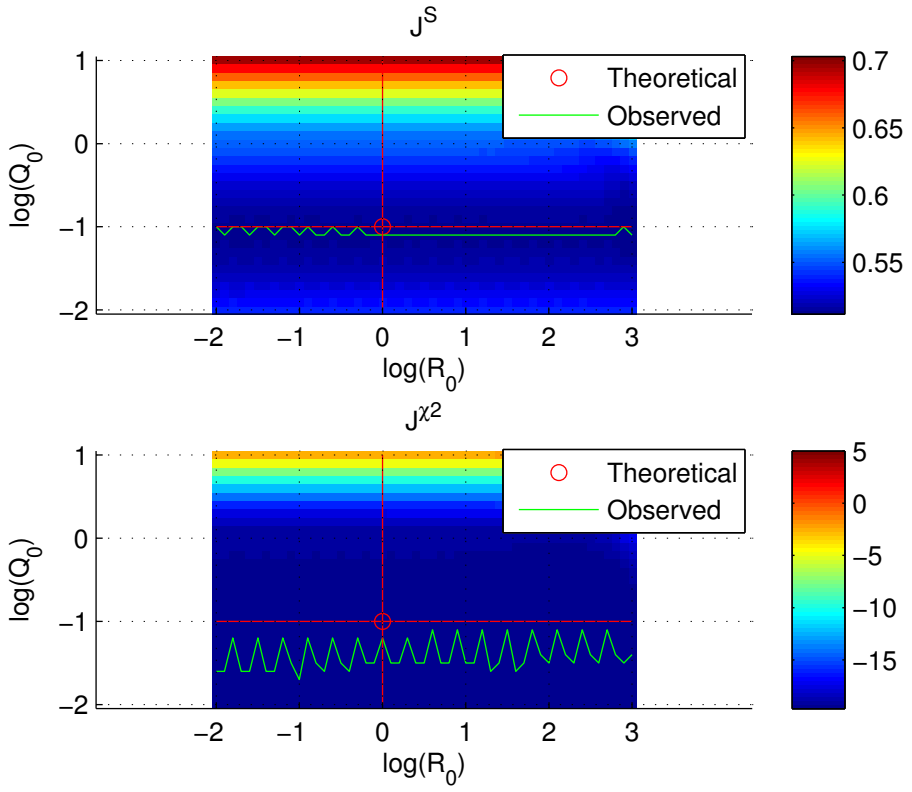

FIGURE 4.6: Colourmap showing average measurement cost(J^S(2.22), upper) and consistency cost (J^{χ^2}(2.24), lower). Red show the theoretical minimum, green the observed minimum for each R_0.

The cost is worse if the filter is started with both R_0 and Q_0 higher than the target. Comapring fig. 4.5 with fig. 4.2 it can be seen that if the filter uses the initial values of $R_0 = 10$ and $Q_0 = 1$ performance is much worse with R estimation.

Unsuprisingly J^S, shown in the upper plot of fig. 4.6 ,has a minimum area which is a line along the x-axis, centered close to the target Q_0. Although this shows that estimating Q using J^S certainly is possible, the cost difference is low. Wchich will result in slow convergance. It can be seen that the starting value R_0 playes makes no difference, in that the average cost is equal along each tested value of Q_0.

Lastly, J^{χ^2} shown in the lower plot of fig. 4.6clearly has a large minimum area. As J^{χ^2} is derived from the TANIS estimators test value, if it not suprising that the estimator minimizes J^{χ^2} to a much larger extent than the other cost functions.

4.4 Summary

The state estimation error J^{abs} has been shown to be lowest for a correct balance between R and Q. With estimation J^{abs} has a minimum around the correct Q. Meaning that filter estimate percision is dependent on a well tuned Q is only R is estimated.

J^P has been shown to achieve a minimum in the area a little over the noise covariances used to generate the measurements. Meaning that when used to compare performance the cost can be lower for filterparameters which are too high. J^P is still a valuable measure as it will spike when $P \to 0$.

On estimation possibility, it was shown that both J^S and J^{χ^2} has a minimum around the target covariances. But that any estimation using J^S is more difficult when the filter parameters are set too high.

Furthermore, if one parameter is wrongly set, the minimum for the other might be wrong, or might not exist. This is true for both J^S and J^{χ^2}. This indicate that estimation of one covariance might not be possible if the other is off-target.

Lastly, when comparing J^S and J^{χ^2} in figure 4.6, it is clear that an estimate of Q from J^{χ^2} cannot be accuratly gotten while estimating R. And while it can be estimated from J^S, it will be difficult and the estimates $\hat{Q}$ will have a high varianse.

CHAPTER 5

ESTIMATOR PERFORMANCE

This chapter compares the different adaptive filters. Using Monte-Carlo simulations of 5'000 runs, the average and variance of different statistics are used to get an idea of the strengths and weaknesses of the different estimation methods. And give a clear idea of their usefullness.

The adaptive filters concidered are

- The Adaptive Limited Memory Filter (ALMF) as described in subsection 3.1.
  It was implemented with a stack size N of 15.
- Sage-Husa (SH) as described in subsection 3.2.
  Implemented with a forgetting factor $b = .95$
- Time-Averaged Normalized Innovation Squared Consistency Estimator (TANIS) as described in subsection 3.3.
  It is implemented with sample size $N = 5$ and a signifigance region $? = 0.05$. The signifigance region means that the estimate change when it is less than 10% chance the filter is consistent with the measurements.
- TANIS / Sage-Husa Hybrid (Hybrid) as described in subsection 3.5.
  It is implemented with the same parameters as the TANIS and SH implementations. SH is reset to $k = 15$.

5.1 R Estimation

Here ALMF, SH, TANIS, and the TANIS/SH hybrid adaptive filters are used to estimate the measurement noise covariance R. Three cases are simulated.

R is kept constant, and the filter are using the correct process noise covariance Q. This shows consistency of the filters.

R is a single square wave from one to four. This shows how the filters respond to sudden change.

R is kept constant, but the filters are using Q wich is set five times too high ($Q = 0.5$). This will show the effect of bad tuning on the estimates.

5.1.1 Static conditions. Correct Q

State Estimation

Looking at the estimate error it is not clear from fig. 5.1 which adaptive filter performes best. However all produce estimates closer to the real value than the badly tuned filter .

Looking at the estimate variance, filters with TANIS and ALMF estimators has the higest variance. The TANIS method will have a high variance, as false positive has a fixed impact. ALMF have a high variance because of its small sample size $N = 15$, but it can also be seen that it rises steadily. This indicate that adaptive filters using ALMF algorithm is not consistent. This is probably due to the resetting method, and has been touched upon in subsection 3.1.

The SH filters variance consistently lies close to the optimal filter.

The TANIS/SH hybrid filter consistently has a slightly lower variance then the optimum. Worryingly this indicate that the filter is inconsistent. There is not further indications of harmful effects.

In turn J^P shows that only SH has a value that matches the optimal. As shown in fig. 4.2 J^P has a minimum for filters with Q and R slightly higher than optimum. The result is that the filter with a too high R, has a lower J^P than the optimum.

J^P also shines a light on ALMF instability. Frequent spikes in the plott indicate that P often get much too small. The EKF equations (2.9)-(2.15) show that if R is small, C_{nu} (2.13) will become too small, which in turn makes the kalman filter gain K too

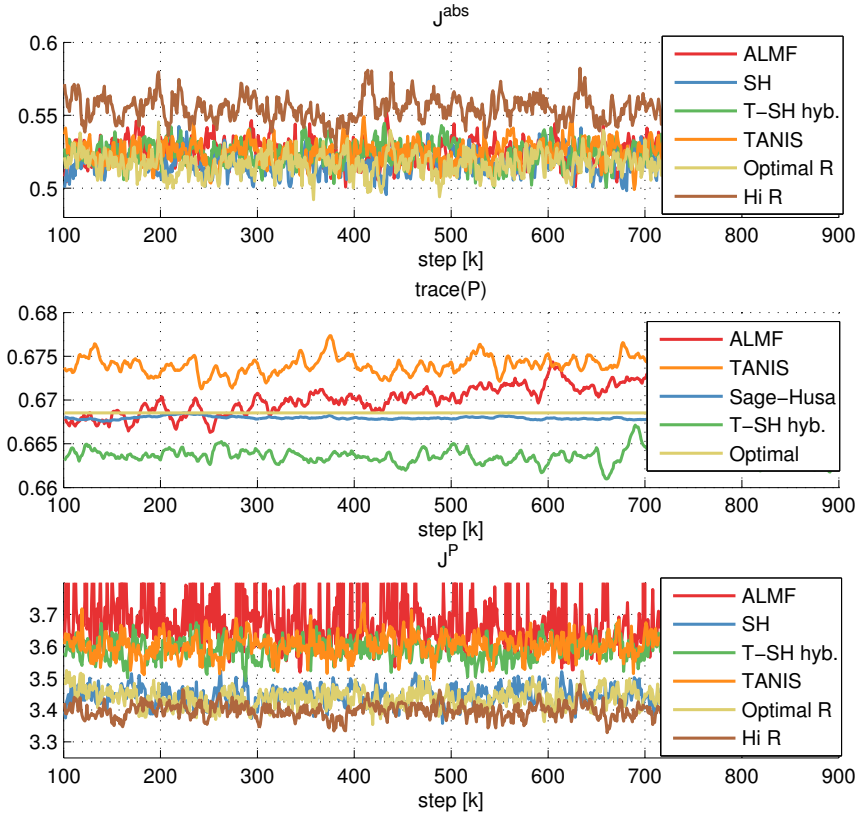

FIGURE 5.1: Abosulte estimation error (upper), trace of filter estimate covariance (middle), and J^P (lower) for filters under static noise conditions.

large. During the update step a posteriori Pand the next a priori P^- will then become too small. If the next estimate $\hat{R}_{k+1}$ also is too low, the problem worsen. The spikes are therefor an indication that the estimator has reset the estimate, and explain the increasing P seen in figure 5.1.

State Estimate Variance

Getting better position estimates are all well and good, but if they vary wildly they are not as useful. Further more well tuned filter will have the estimate covariance P match the actual state estimates. Fig. 5.2 shows the trace of P compared to the calculated estimate variance. The calculated P is the average of $(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T$ at each step.

All the adaptive filter have P sufficiently close to the observed P, meaning that they are performing well. The hybrid hit the observed covariance worst, being consistenly lower, but not so much as to deem the method unsuitable. The TANIS and ALMF filters are also slightly lower than the observed, but less so than the hybrid.

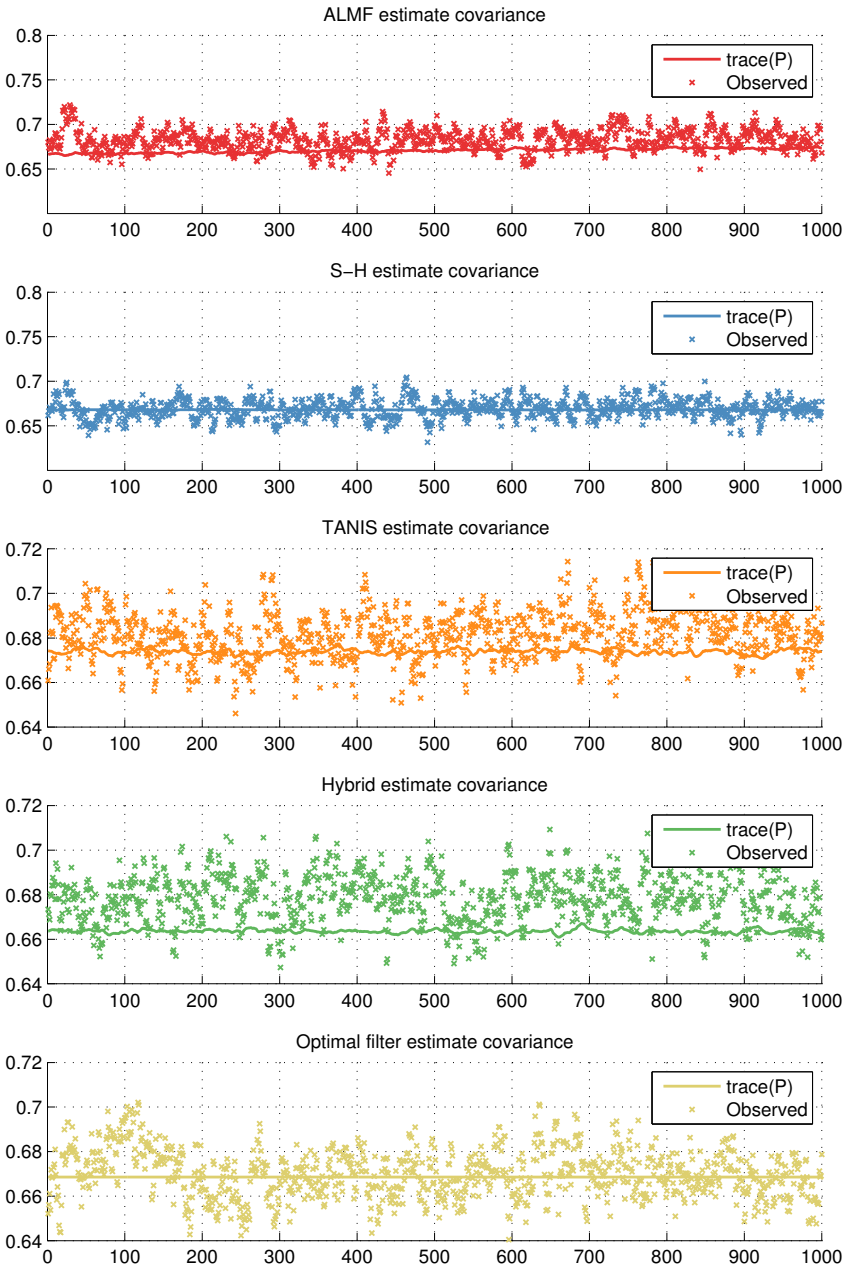

FIGURE 5.2: The trace of average estimate covariance P used by the filters compared with the average calcuated estimate covariance $(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T$.

Comparing the methods is easier in fig.5.1, but considering what can be seen in fig.5.2 some futher observations can be made.

TANIS has the highest variance, and while the hybrid has the lowest. SH is the closest match to the optimal, both in the observed and filter P, and matches best with its own observed state estimate covariance. The other methods has a slightly lower P than ideal.

R Estimates

Fig. 5.3 shows the mean $\hat{R}$ divided by the optimal, $R = 1$.

The ALMF estimate can be seen steadily rising. This is due to the increasing systematic error resulting from the handeling of non positive definite estimates. This has been explained in subsection 3.1, and is what caused its P to steadily rise (see fig. 5.1).

TANIS has consistenly a higher mean estimate which is as expected because when the TANIS estimate increase, it is increased four times more (multiplied by 2) compared to when the estimate is decreased (estimate is halved). This means that if there are the same amount of false positives increasing and decreasing the estimate, the mean estimated R will be higher rather than equal the true value.

This is also true for the hybrid estimator, but to a much lesser extent, indicating the addition of SH has a positive effect on the estimates.

SH is close to the optimum. Concidering figures 5.1 and 5.2 this is not supprising given how well the SH filter match the optimal.

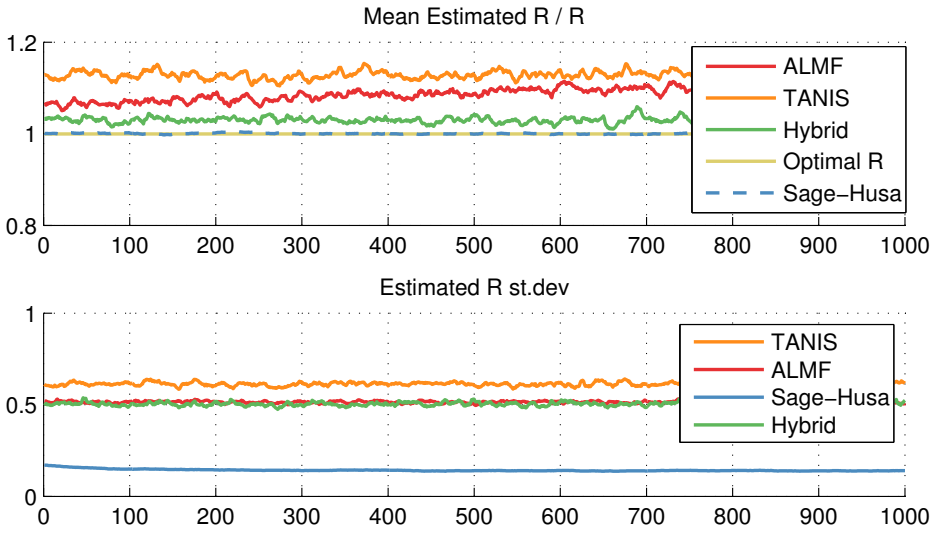

FIGURE 5.3: The average $\hat{R}$ of MC simulation with 5000 runs (upper), and their standart deviation (lower).

Looking at the different estimate variances (lower plot in fig. 5.3) it is clear that: the TANIS estimator has the highest variance, the hybrid matches the ALMF, which is lower. and the lowest is SH.

Initial steps

To allow the filters to settle, a preliminary 150 steps are run before considering performance. The initial steps are of some interest, particularly the initial convergance. As the hybrid filter resets its SH frequently, the SH methods ability to match the statistic initially is important.

It is seen in fig. 5.4 that the Hybrid and TANIS estimators settle at the same time. The addition of SH does not make the Hybrid settle quicker than TANIS, eventhough it is started initially, while TANIS waits for a full sample.

ALMF too settle at the same time, despite having a sample stack three times the size (N = 15). It and SH initially underestimates, but settle just as quick as the other methods. This might be suprising for SH, as it has been mentioned that it is the slowest of the metods considered. However during the initial phase, the weight d is is much higher, allowing the incomming inputs to have a greater impact.

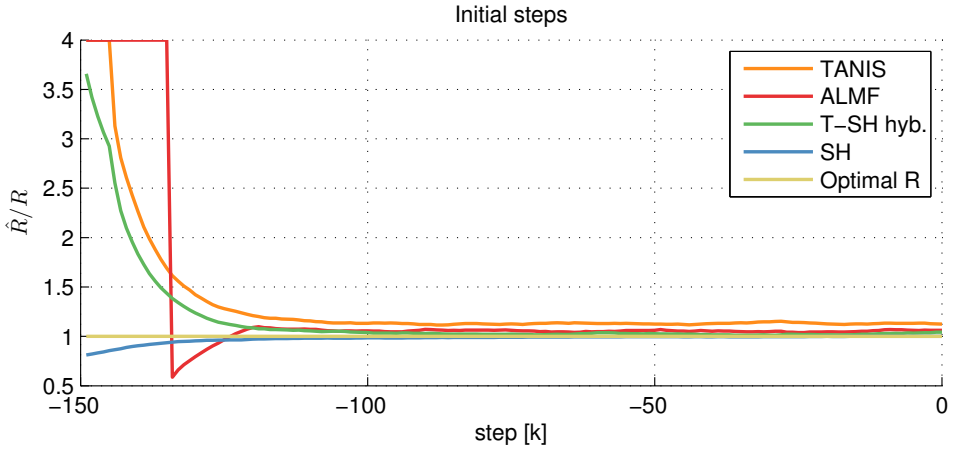

FIGURE 5.4: Noise covariance $\hat{R}$ during the initial steps. Filter Q matches the generated process noise covariance.

5.1.2 Square wave

In the following simulation R increase from 1 to 4 at step 250, and decrease to 1 at step 750. The badly tuned filter uses $R = 4$ throughout, meaning it will match the optimal from 250 to 500. All filter use the correct Q.

State Estimation

Although it is difficult to distinguish the performance of the estimators in fig. 5.5 it is clear from the top plot that adaptive filters perform better than the badly tuned filter. This is true both before and after the noise step.

Focusing on the filters estimate covariance P during the step, it can be seen that ALMF and the hybrid are both below the optimal filter, indicating they are inconsistent with the actual estimate covariances. This was also observed under static conditions in fig. 5.2.

The covariance estimates convergance speed will be dealt with shortly, but it is clear that SH is the slowest to adapt to the changing R. So slow in fact that it can be ruled out as a viable method for estimating a quickly changing R. Eventhough it is the closest to the optimal filter when it settles.

ALMF has fewer spikes when R is higher, which is clear from the lower plot in fig. 5.5. This is an indication that ALMF if performing better for higher values of R. Still, the inconsistencies is still a problem.

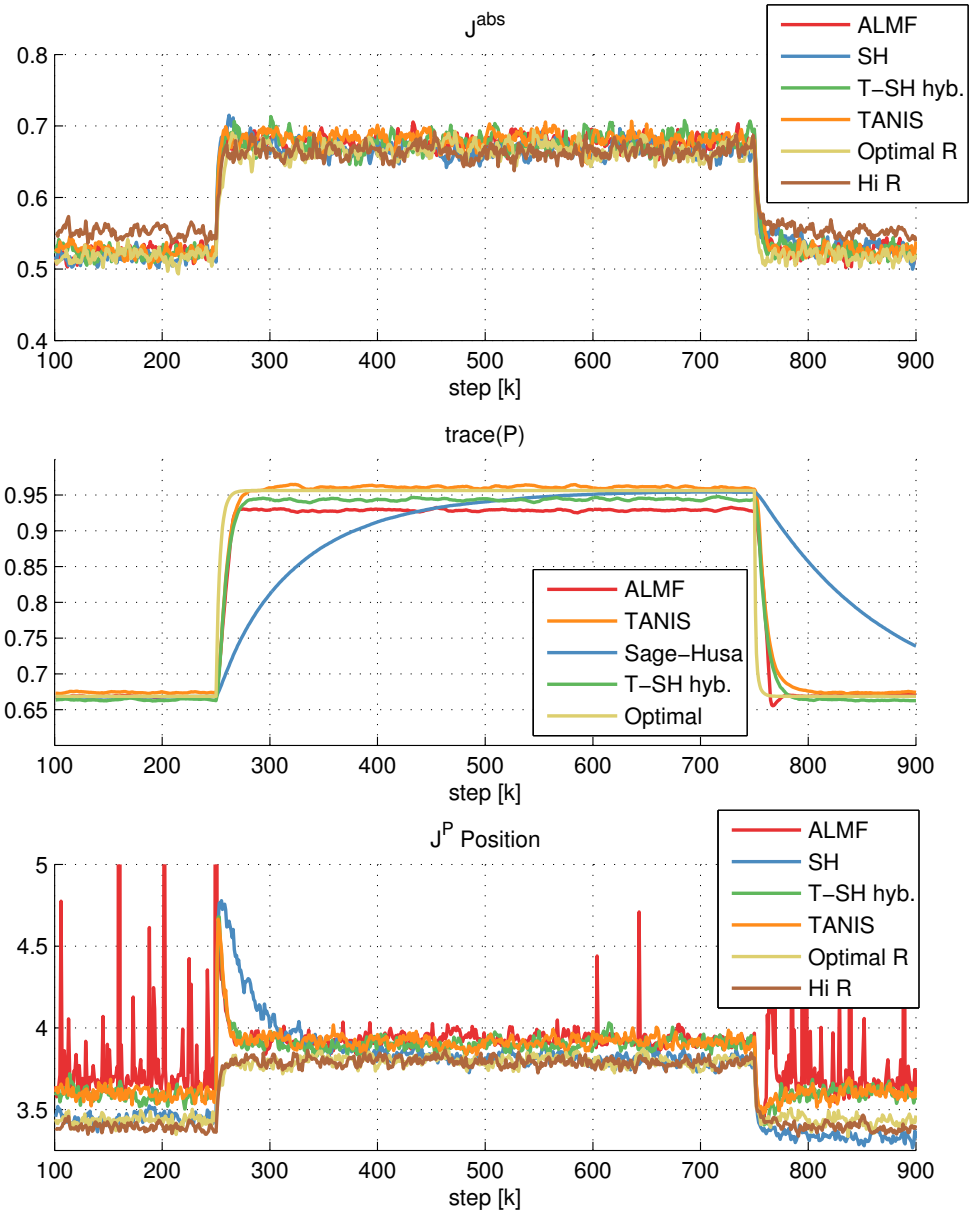

FIGURE 5.5: The trace of average estimate covariance P used by the filters compared with the average calcuated estimate covariance $(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T$. The noise caovariance R of the signal is modeled as a square wave.

Rise and Fall times

As it is difficult to see the estimators responses from fig. 5.5, a closer look at the estimate $\hat{R}$ is shown in fig. 5.6. It shows the mean of the MC simulation divided by the optimal R. The target is therefore 1 throughout. Because SH is slow, the figure focus on the remaining methods.

All the estimators settle around step 25 after the step increase, and around step 30 - 35 after the step down.

ALMF has an overshoot for both the step up and down, which can cause inconsistencies if R suddenly becomes low. Overshoot is not seen in the other methods.

The Hybrid performs best of them all, quickly setteling without overestimating as much as the TANIS estimator.

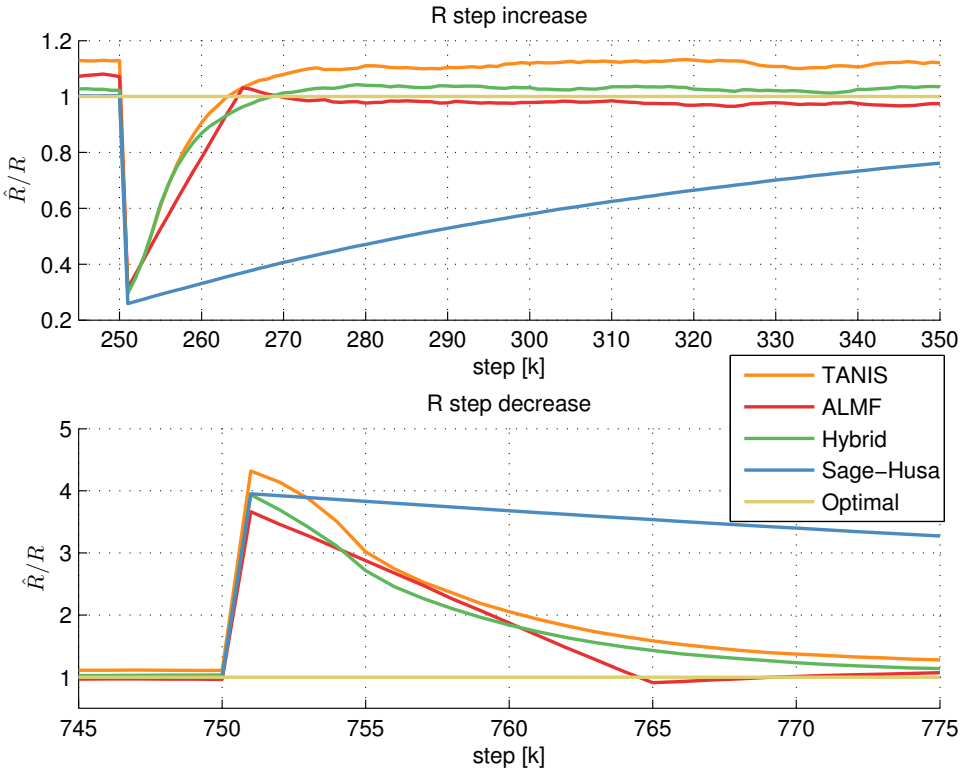

FIGURE 5.6: The average $\hat{R}$ of Monte-Carlo simulation. Sections right after step up (upper) and step down (lower) in signal covariance. The estimates $\hat{R}$ are divided by the optimal value of R for clearity.

Lasting effects

To investigate any lasting effects of the change on the estimators other than SH, the first and last 100 steps of the R estimate is shown in fig. 5.7.

From the figure it can be observed that the TANIS and Hybrid estimates are unchanged, while the ALMF has increaced slightly. This is more likely due to the aforementioned inconsistency of the ALMF method.

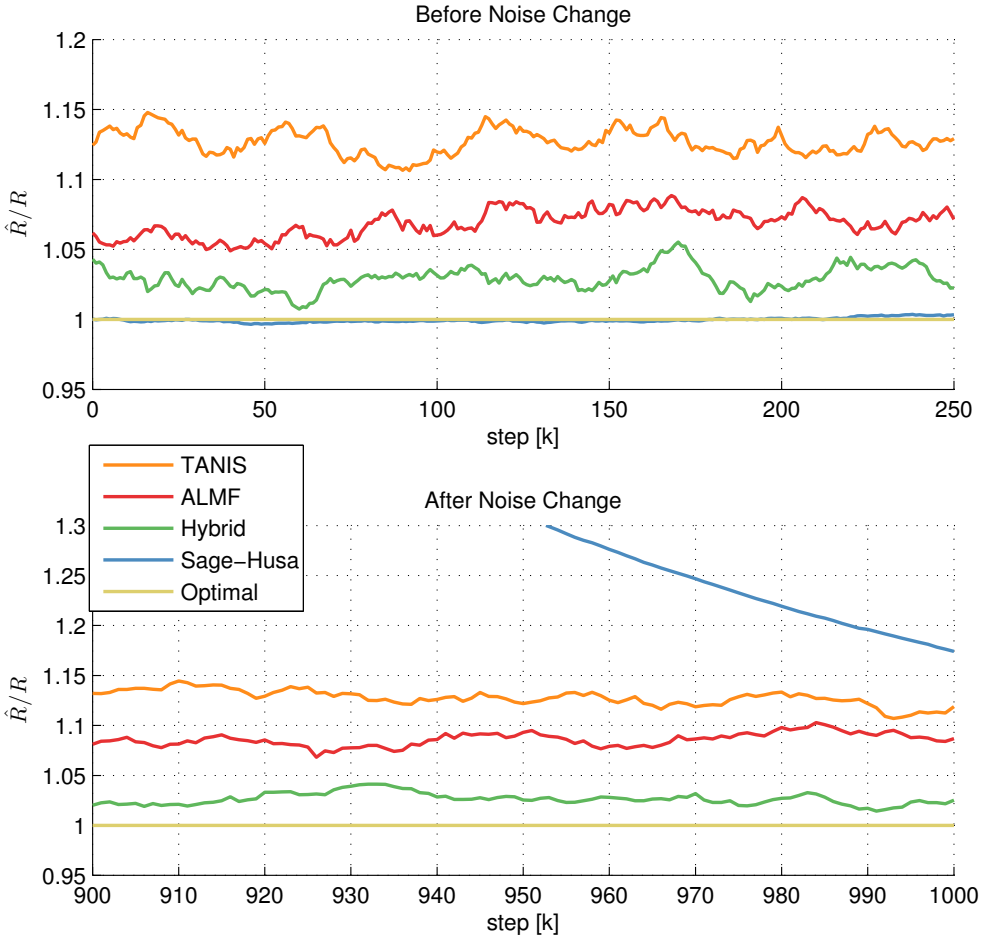

FIGURE 5.7: Mean noise covariance estimate $\hat{R}$ divided by the true value, before (upper) and after (lower) change in measurement noise covariance.

5.1.3 Badly tuned filter

Q usually cannot be assumed known. In most filter is it approximated and set too high. As previously mentioned, this affect R estimation, which can result in less accurate state estimates (see subsection 2.5.5).

To show this, simulations have been run on the same signals, but with the diagonal Q elements set to .25, five times larger than their theoretical value.

State Estimation

Fig. 5.8 clearly show that the adaptive filters lead to worse positional estimates than the wrongly tuned filter. It is expected as the badly tuned filters R/Q ratio is closer to the optimal. And it points to a systematic error in $\hat{R}$ for all methods. Curiously, ALMF produces better estimates than the others.

The estimates covariance P is much higher than both the optimal and the badly tuned filters, which shows that their performance has suffered from the bad tuning. Interestingly the ALMF has a higher P than the others, even though it has less state estimate error.

Lastly the cost function J^P is as expected from the other plots in fig. 5.8, and shows frequent singularities for the ALMF. Again, this is an indication that the ALMF filter is reset often, and the ensuing permanent error mentioned in subsection 3.1 is apparent.

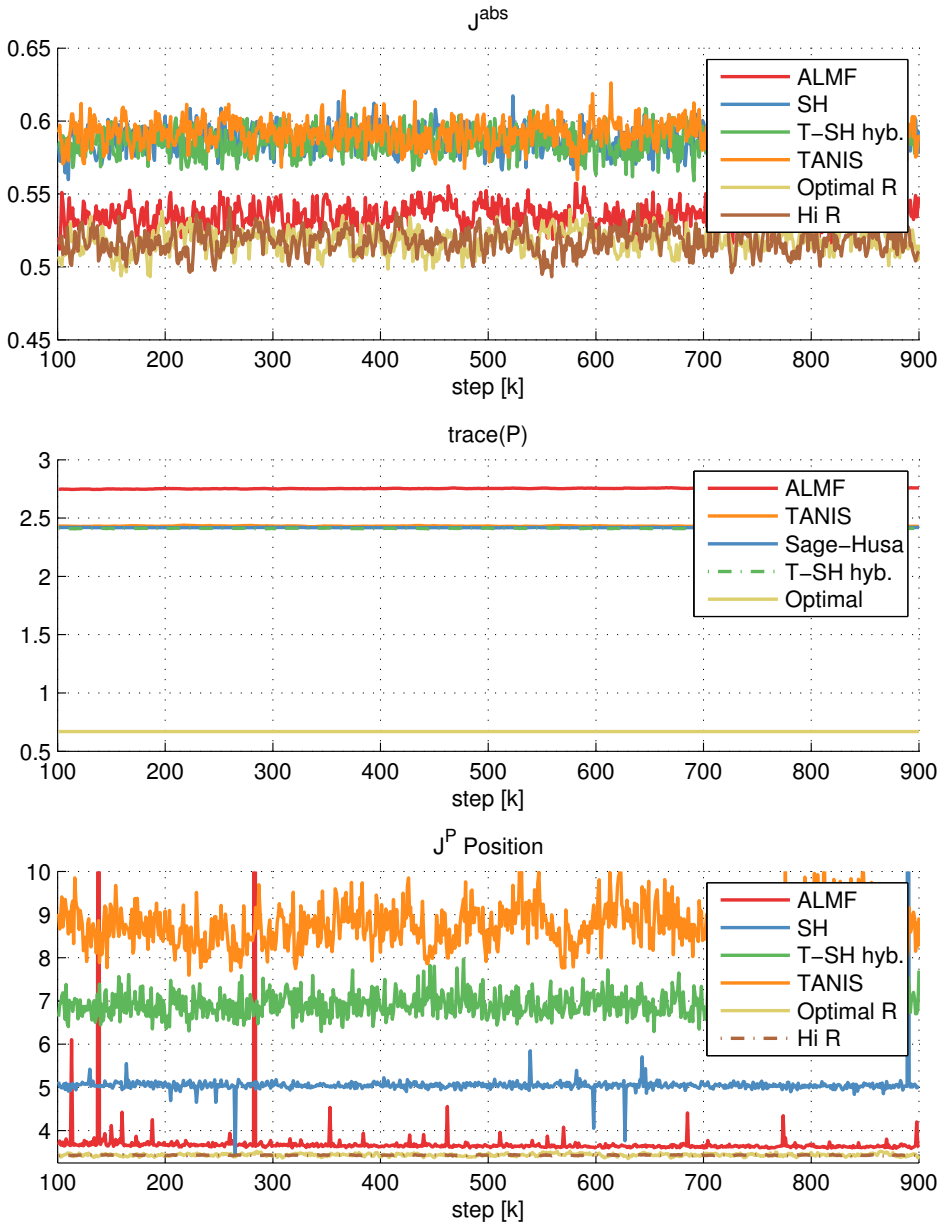

FIGURE 5.8: Abosulte estimation error (upper), trace of filter estimate covariance (middle), and J^P (lower) for filters under static noise conditions. Adaptive filters use $Q = 0.5$.

State Estimate Variance

Unsuprisingly, the filters P matches badly with their respective observed state estimate covariance. Showing again that Q must be set close to the correct value for R estimation to be effective on its own. The filters P and calculated state estimate covariance is shown in fig. 5.9.

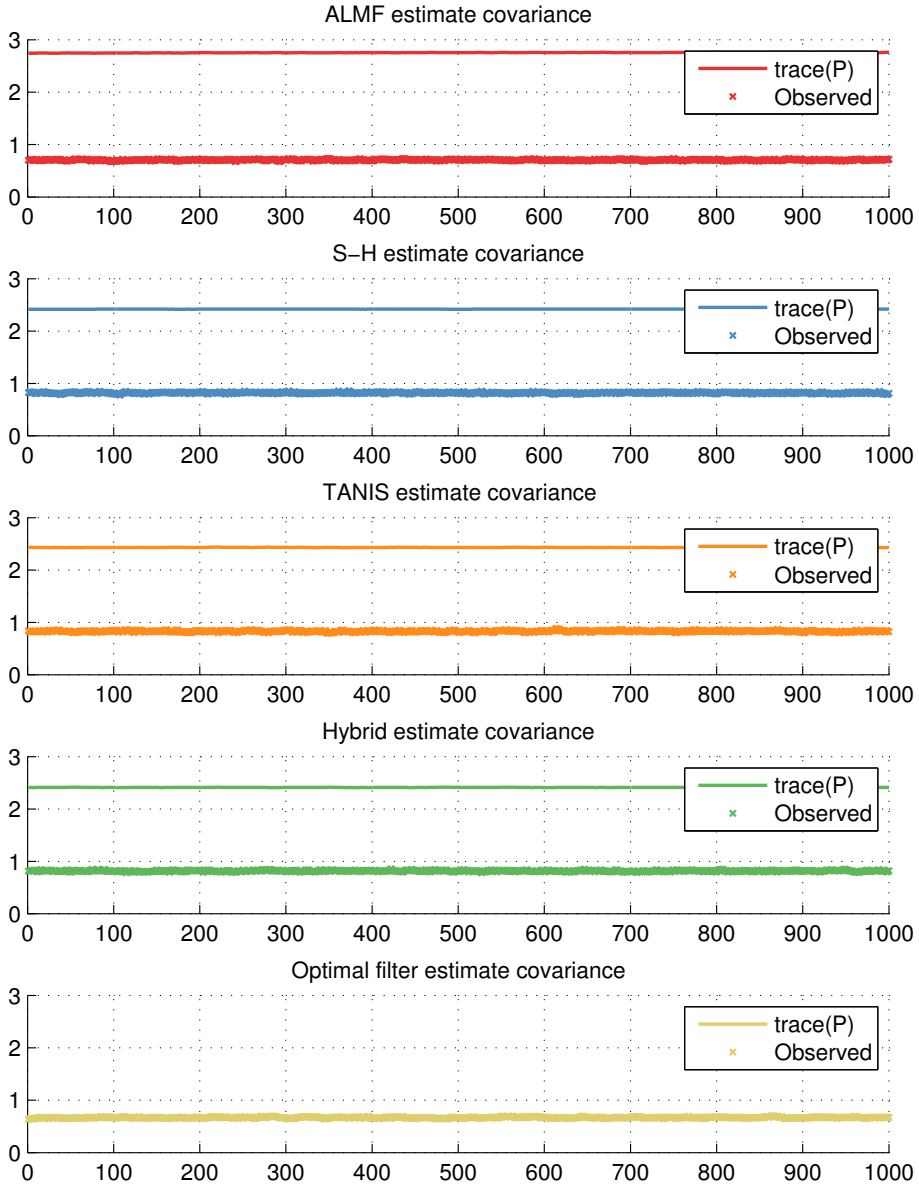

FIGURE 5.9: Trace of filter estimate covariance for various adaptive filters, with correctly tuned KF (bottom).

R Estimates

Fig. 5.10 shows clearly that estimators produce $\hat{R}$ which is too low. TANIS, SH, and TANIS/SH hybrid estimators all settle at around half the R target. As indicated by eq. 2.80, this is to be expected.

The ALMF estimates are of particular interest, as they increase slowly, almost having settled at a value close to two times the target. This will be dealt with in more detail in the next part of this section.

Again it is seen that SH estimates has the lowest R estimate std., hybrid slightly below TANIS, with ALMF having the highest. Compared to fig. 5.3, TANIS and Hybrid has a lower std., which is expected as the false positives will have less of an impact when the estimates are low. ALMF has a higher std., expected because of the clear systematic error seen. SHs std. closely match.

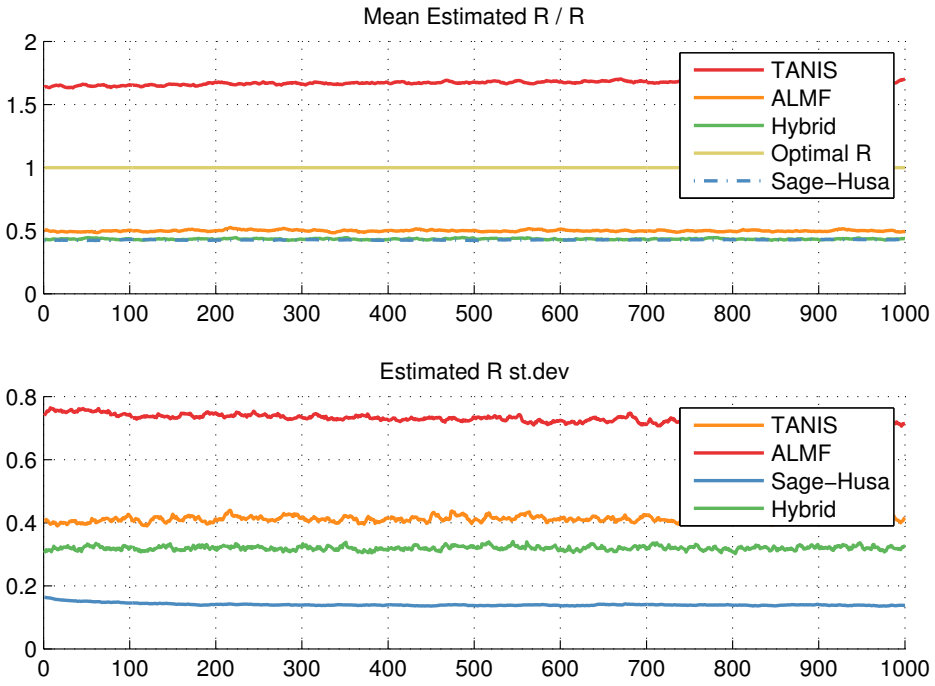

FIGURE 5.10: Estimated noise covariance $\hat{R}$ divided by the correct R. Adaptive filter initiated with $Q = 0.5$. Correct $Q = 0.1$.

ALMF instability

To better see why ALMF gives such different estimates than the other estimators, the initial steps are the most informative. Fig. 5.11 show that the other estimators perform

much as in fig. 5.4, apart from converging to a lower value. They will therefore not be discussed further.

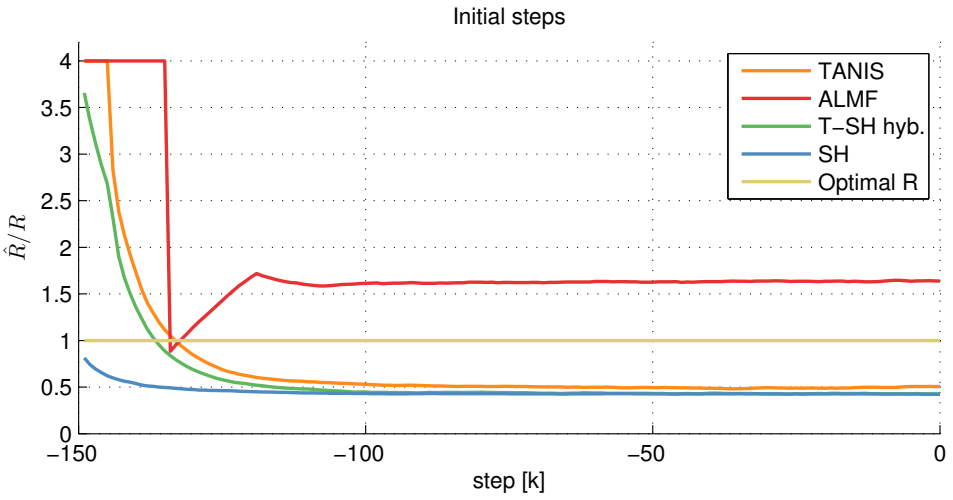

FIGURE 5.11: Initial noise covariance estiamtes $\hat{R}$ from different estimators, with Q set incorrectly to 0.5.

In fig. 5.11 ALMF estimates can be seen starting at a value close to the TANIS and Hybrid estimates but then steadily increase. As ALMF estimates using the same mathematical basis as SH, it is expected to decrease their $\hat{R}$ towards the same value. However the ALMF estimates increase, which is explained by the fact that the estimate is reset to its absolute value whenever the estimate becomes negative. Futhermore, it is clear that this method is incapable to overcome this error.

5.2 Q Estimation

Estimating the process noise covariance Q is of some interest, but not directly relevant on its own. Because it is assumed that Q is constant or slow moving, the adaptive filters will only estimate unchanging Q. The estimation methods concidered are ALMF, SH, and TANIS. The hybrid is not included because it is designed for estimating quickly changing statistics, and since it is a combination of TANIS and SH, adding it here would be surperfluous. β-MLE is not included in this section as it is designed to work in conjunction with a R estimator.

The optimal filter had uses the value for Q which has been used to generate the simulated measurement signal. $Q = 0.1 * I$. The other filters have been initialized with, and the badly tuned filter use $Q = 0.25 * I$.

The figures data is the mean and variance calculated from a Monte-Carlo simulation with 5000 runs.

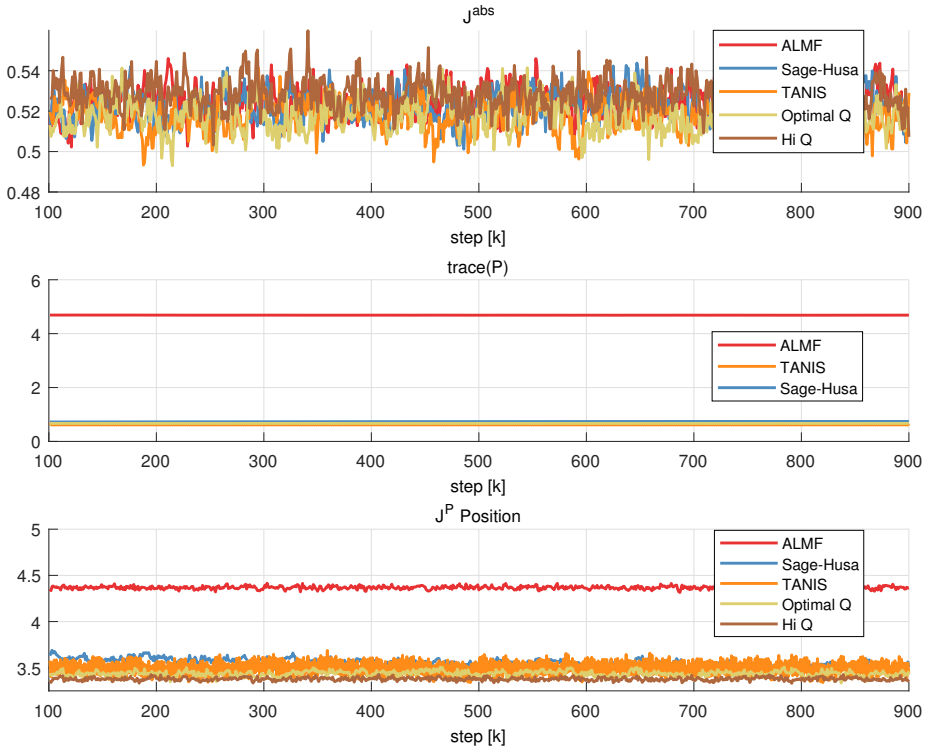

FIGURE 5.12: Abosulte estimation error (upper), trace of filter estimate covariance (middle), and J^P (lower) for filters under static noise conditions, while estimating Q.

State estimation

From figure 5.12 it is hard to judge which filter has the least absolute estimate error (top), but it is clear that the addition of estimators does not give conciderably worse estimates.

In the middle plot of the same figure, it is clear that the ALMF filter use a estimate covariance P which is much too high. TANIS and SH lies close to the optimal, albeit slightly lower and higher respectivly.

Q estimation

Figure 5.13 shows that the ALMF estimates are consistently wrong, and for $\hat{Q}$ is worse than the itital value. There is nothing to show that the method is capable to overcome this. Because ALMF reset the diagonal elements to their absolute value whenever they become negative, the off-diagonal element compensate by becomming increasingly lower. This is an indiaction that ALMF is only usefull when the estimated statistic is higher

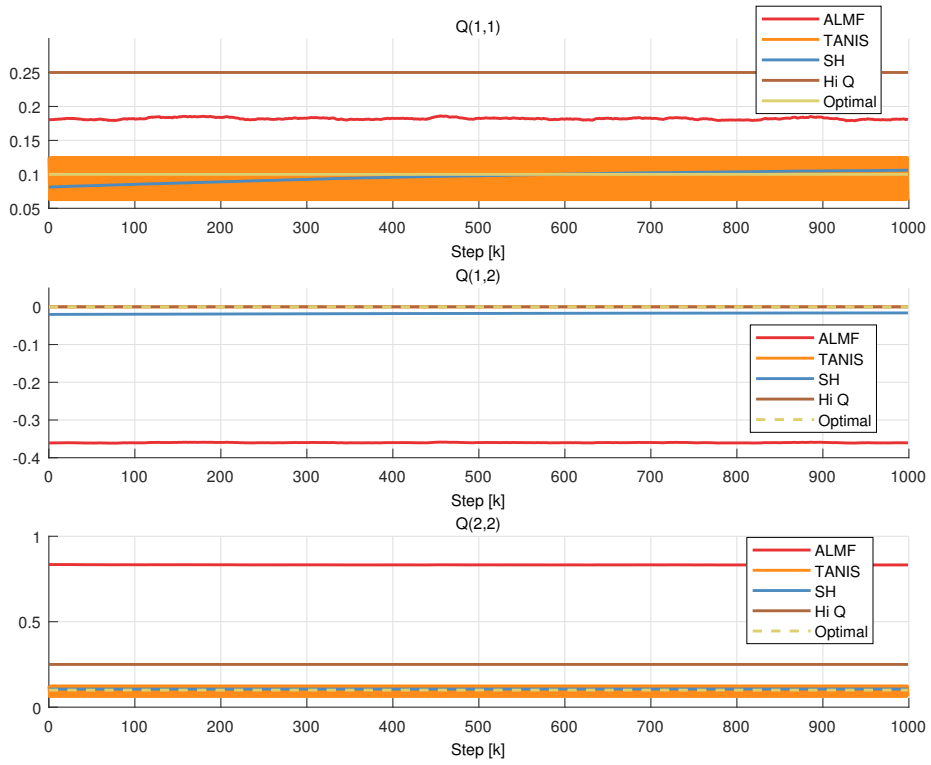

FIGURE 5.13: Mean process noise estimates $\hat{Q}$ of Monte-Carlo simulation of 5000 runs.

and its variance is lower than a certain level. This level is undoubtedly related to the sample size N, so no general rule can be found.

SH can be seen to match closely, although $\hat{Q}_{(1,1)}$ rises steadily (but slower towards the end of the simulation). Closer examination of the data revealed that the off-diagonal elements went towards zero at the same speed, so $\hat{Q}_{(1,1)}$ could converge at some point, but from figure 5.12 this cannot be definitly concluded. $\hat{Q}_{(2,2)}$ closely match the target 0.1 thoughout the simulation.

TANIS can be seen occilating around the target. Because of the structure of the TANIS estimates, which has been mentioned in section 3.3, the estimates are unable to exactly match the target. Since TANIS scales a known structure, its $\hat{Q}$ has no off-diagnal elements.

Looking at figure 5.14 it is seen that ALMF produce estimates with the highest standard deviation, SH estimates are lower, while TANIS estimates std. are close to zero. This indicate that TANIS is superior to the other methods used.

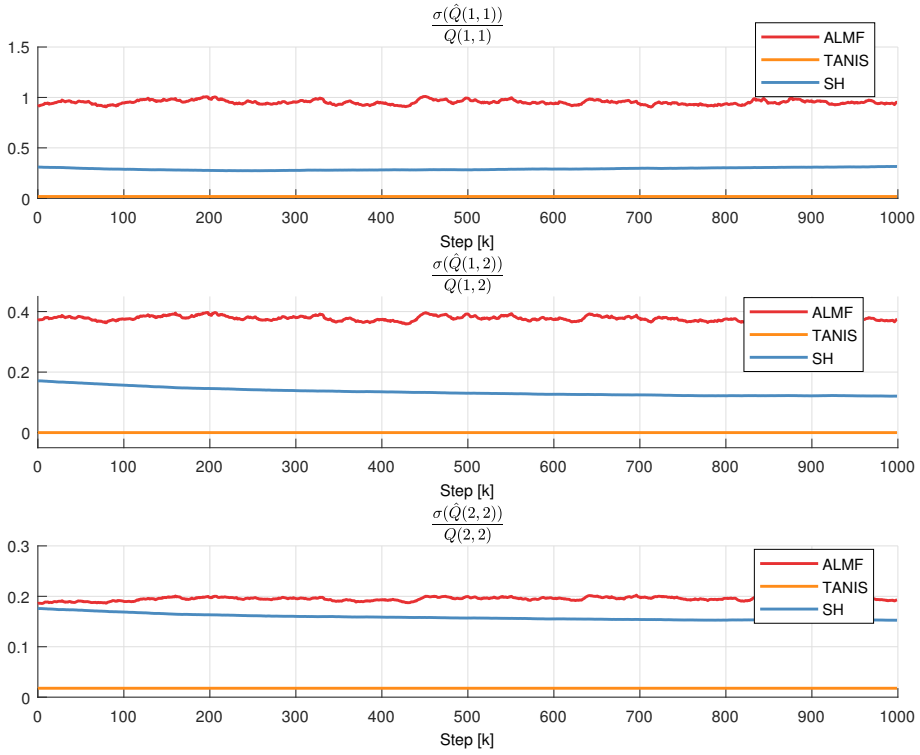

FIGURE 5.14: Standard deviation of $\hat{Q}$ divided by the true value $Q = 0.1$

5.3 Concurrent R and Q Estimation

Since the process noise covariance Q has an impact on the performance of the filters, concurrent R and Q estimation needs to be concidered. Since the effects of $\hat{R}_k$ on P^- is not seen before the next step, Q and R cannot be estimated during the same iteration.

In these simulations two estimators have been used to create an adaptive filter which estimate R and Q on alternate steps.

- ALMF uses the ALMF method for both R and Q estimation.
- SH uses the SH method for both R and Q estimation.
- TANIS uses the TANIS method for both R and Q estimation.
- Tanis/Sage-Husa hybrid (T-SH) uses the hybrid method for estimating R and SH for estimating Q.
- β-MLE uses the hybrid method to estimate R and the ratio balancer to increase Q and decrease R as described in section 3.4.

- The optimal filter again uses the correct filter parameters throughout the simulations
- The badly tuned filter (Hi R) uses the correct Q, while having R set four times too high (R = 4).

Because of the long time it took running the β-MLE estimator, its values shows have been calculated from a Monte-Carlo simulation with 1000 runs, rather than 5000 runs as simulated for the other methods.

5.3.1 Static conditions

First the performance during static conditions are investigated, meaning that the measurement signal was generated with constant noise covariance.

The adaptive filters have been initialised with the diagonal elements of Q = .4, and R = 4. The signal was generated with $Q(i,i) = .1$ and $R = 1$.

State estimation

Figure 5.15 shows clearly that both the TANIS and ALMF methods are unsuited for estimating R and Q together. ALMF has a consistently high error and has the highest aposteriori estimate P. This does not mean that it is no longer estimable, but rather that the estimates are less accurate than the other methods (apart from TANIS).

β-MLE can also be seen to have a high P.

TANIS on the other hand clearly performs the worst, with estimate errors that are much larger than the rest. Interestingly trace(P)$\rightarrow$ 0 which indicate a loss of estimability (Definition 2, section 2.5.2)

To accuratly compare the remaining methods figure 5.16 shows a narrowed vertical section of fig. 5.15.

We see that the AMLF filter has more estimate error and a higher cost J^P than the other methods.

β-MLE does not have less error in its positional estimates then the badly tuned filter, or the other methods. As the method is a simplification of a full MLE method, it should be noted the effect of noise is prononced. And it does seem give better results over time, indicating that it either has settled at wrong estimates, or will converge very slowly.

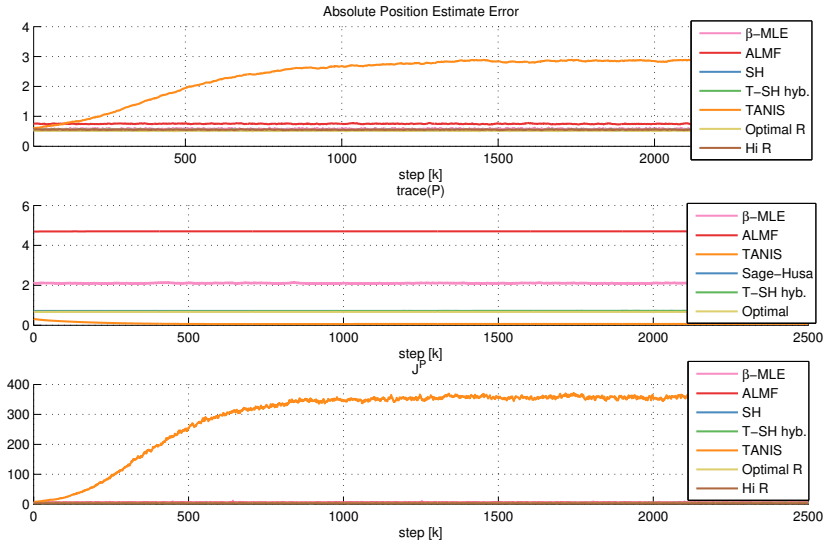

FIGURE 5.15: Absolute state estimate error, state estimate covariance, and J^P during concurrent estimation under static conditions.

The spikes the method has in J^P are not inconsistencies, like those frequently seen for the ALMF method, but rather the result of noise entering the estimates.

SH and the hybrid are those that perform best. Having less estimate errors than the badly tuned filter, the lowest cost J^P, and P closest to the optimal. However, the middle plot in fig. 5.15 it is clear that the hybrid is unstable, increasing similar to TANIS. Not unexpected because they both rely on a consistency hypothesis test which change the estimate in steps up and down, but it shows that even with a method giving continuous Q estimates creates instabilities in for $\hat{R}$.

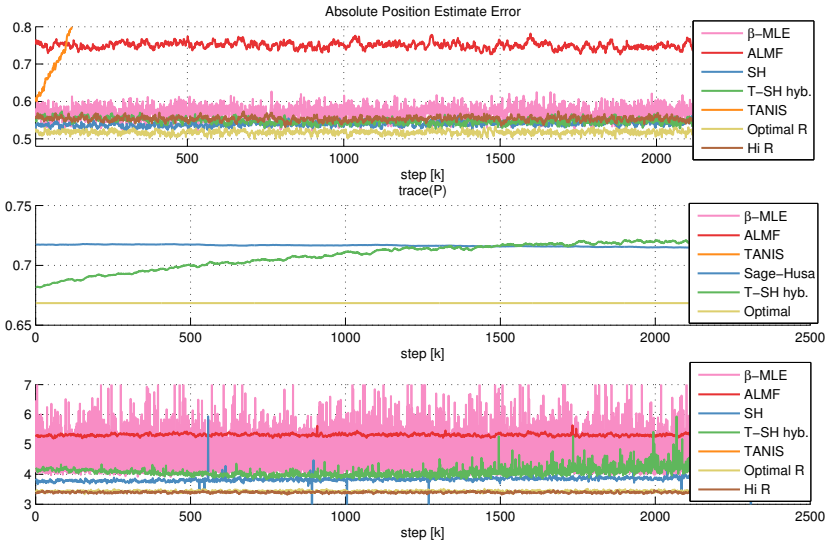

FIGURE 5.16: Absolute state estimate error, state estimate covariance, and J^P during concurrent estimation under static conditions.

Estimate Covariance

Looking closer at the estimate covariance P in fig. 5.17 gives an idea of how well the estimability property holds as well as the quality of the state estimates.

Stating what already was observed from fig. 5.15 and 5.16, the β-MLE and ALMF filters P settle much higher than the observed estimate covariance. Showing that they perform worse than SH, which matches its oberved value and is much closer to the optimal.

The effect of having P close to zero is shown for TANIS, the ture estimate covariance is steadily increasing, while the filter does not follow.

The hybrid on the other hand, increase P to match the observed values. Although it is not garanteed that it will settle, the change is slower towards the end.

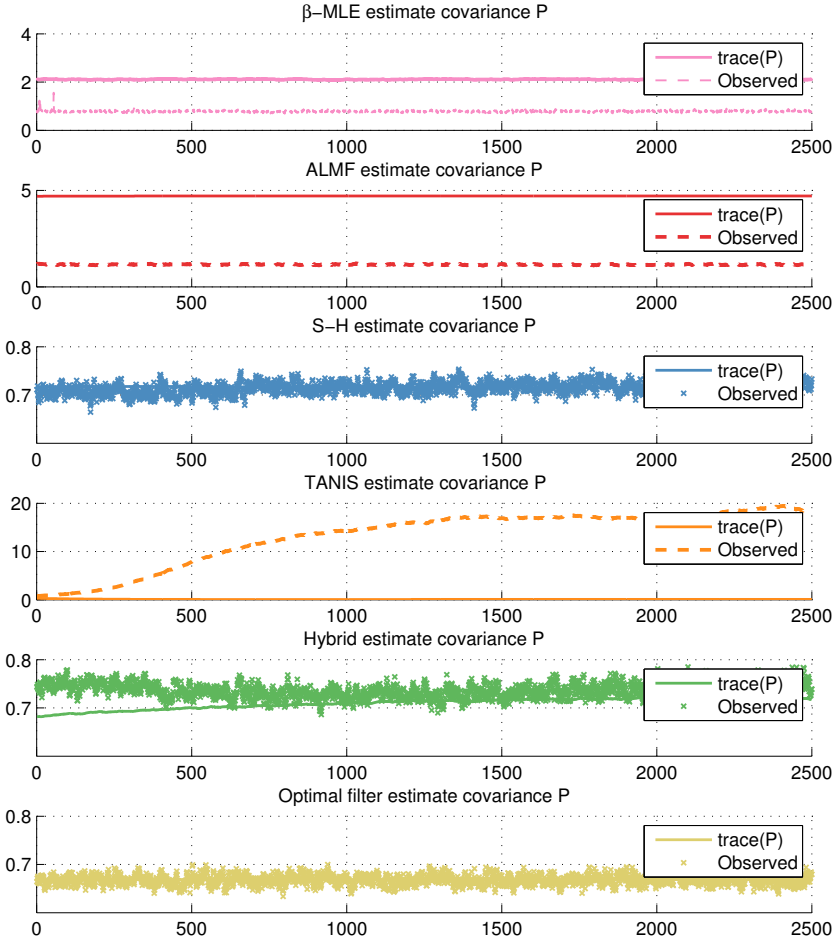

FIGURE 5.17: Aposterioi state estimate variance and the observed state variance

Covariance estimation

Figure 5.18 shows what already was indicated in the previous figures. The TANIS estimates increase throughout the simulation, while the ALMF has settled at a value higher than the initial value. β-MLE produce estimates that are slightly lower than the optimal, which points towards an inability to get $\hat{Q}$ to converge to a correct value. The other methods are close to the optimal but the hybrid deverge downward towards the end.

It can also be seen that the standard deviation (shown in the lower fig. 5.18 plot) is what one would expect. TANIS with its increasing step sice is the highest, and ALMF is higher than the remaining methods.

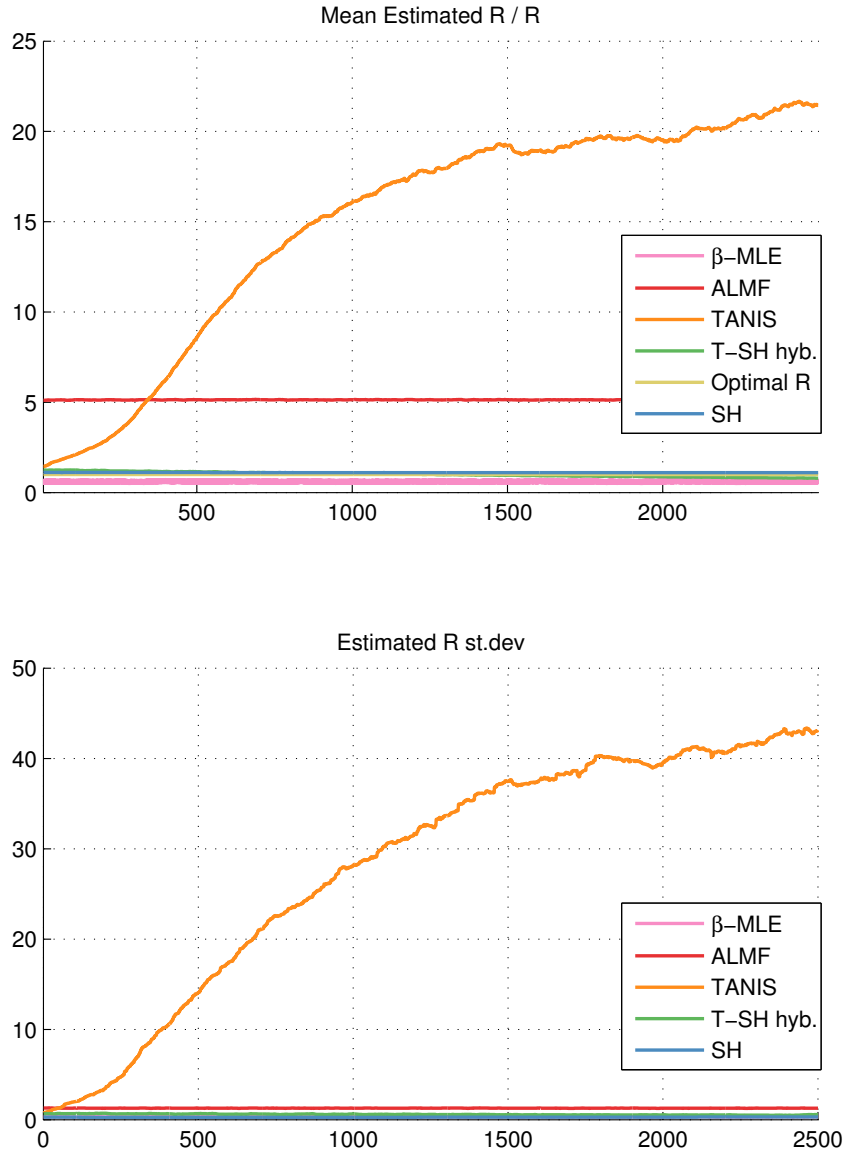

FIGURE 5.18: Mean $\hat{R}$ during estimation of R and Q

In figure 5.19 we see that β-MLE give $\hat{Q}$ which do not match the optimal, abeit better than the initial, it is still much larger than the target. This is what leads to its R estimates to become too small, which has been shown for other methods in subsection 2.5.4. Remembering that TANIS and β-MLE uses a known structure for its estimates, they do not estimate off-diagonal elements with the current implementation.

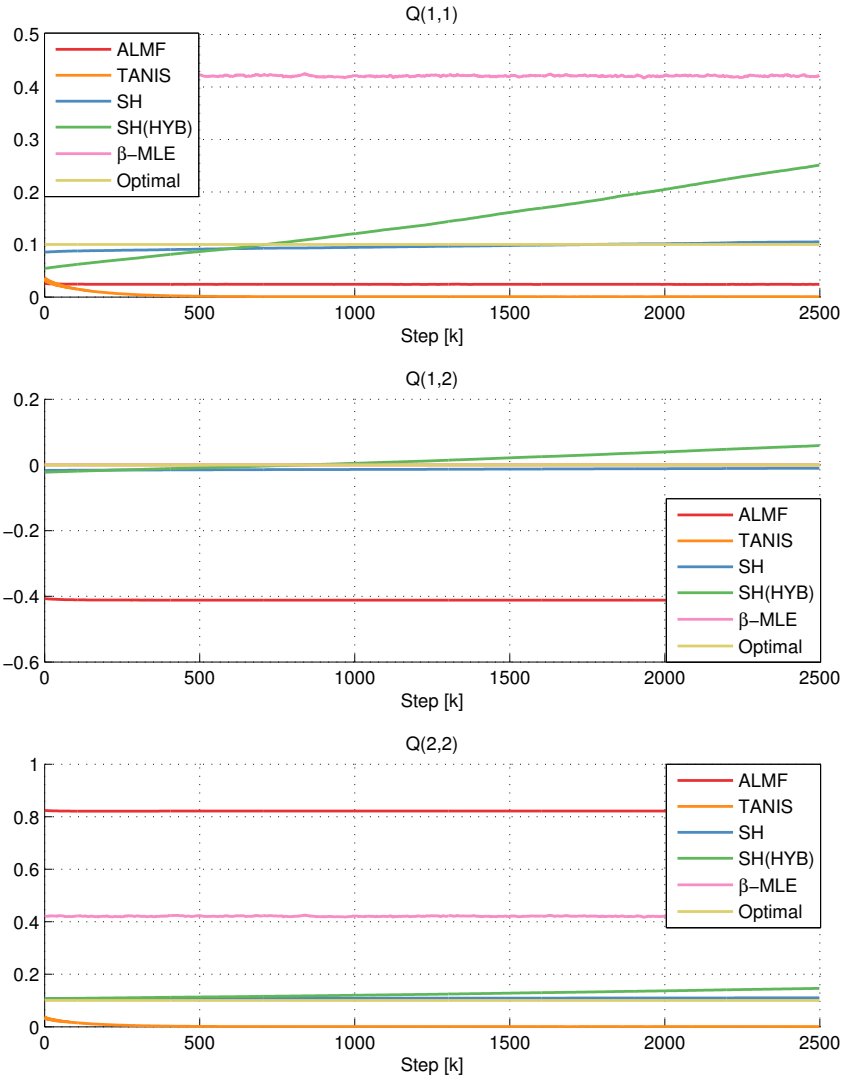

FIGURE 5.19: Mean $\hat{Q}$ during estimation of R and Q

Again what has been noted in the preceding figure are shown here as well. the TANIS estimates are unstable, as $\hat{R}$ is increasing $\hat{Q}$ is decreasing, converging on zero. Even-though it is theoreticly impossible for TANIS estimates to actualy become zero, they clearly become close enough to cause the estimates to become unreliable. And had not a lower limit been set at one thousandth, the estimates would have been rounded down

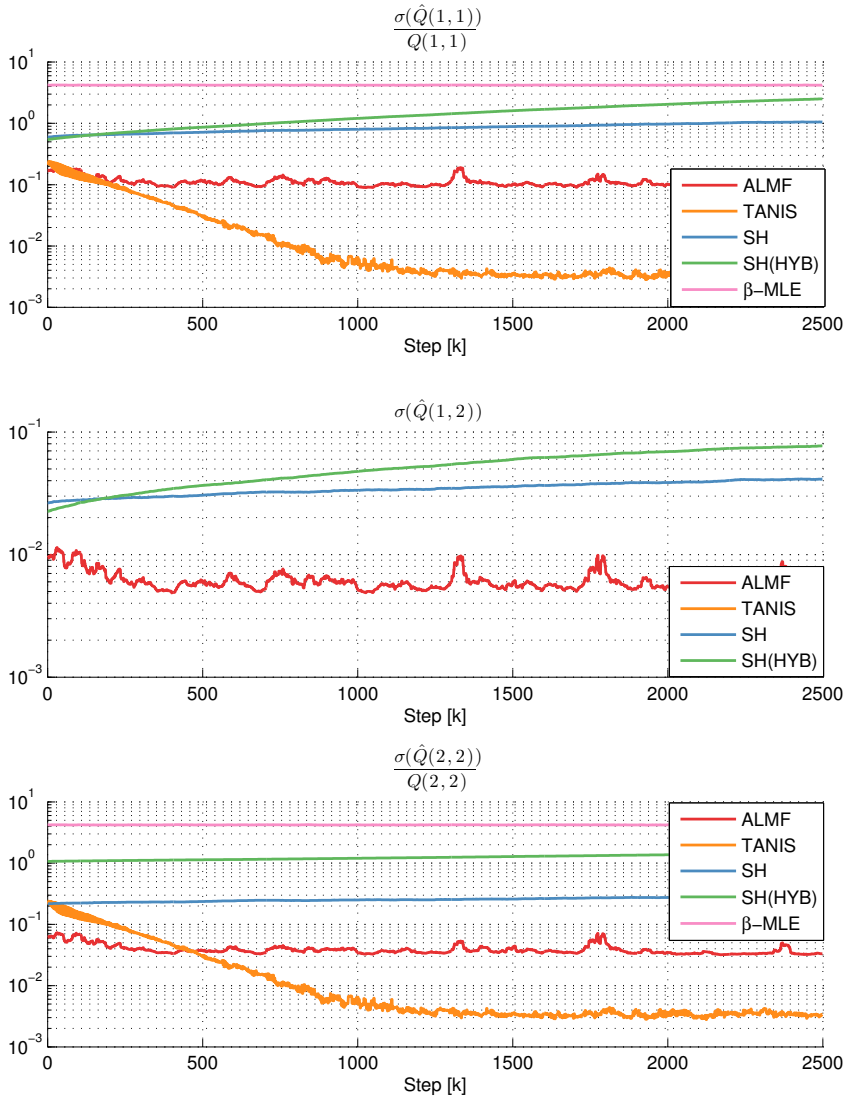

FIGURE 5.20: Std of $\hat{Q}$ during estimation of R and Q

to zero in the simulation. Therefore estimability cannot be given to TANIS estimation of both R and Q.

ALMF underestimates $\hat{Q}$ consistently, which is consistent with its overestimation of $\hat{R}$. It also has significant off-diagonal values. SH has estimates which are close to the target optimal through the whole simulation.

Of particular interest are the hybrid estimates, which increase past the target, and that has increasing off-digaonal elements. Again, this might mean that it is slow to stabilize, or the presense of an instability as seen for the TANIS estimator, although supressed by SH estimating Q.

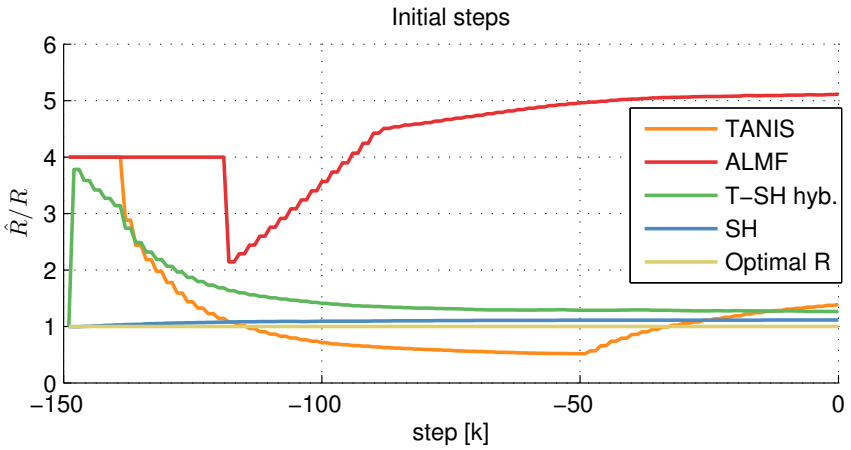

FIGURE 5.21: $\hat{R}$ during initial steps while estimating R and Q.

It is seen in fig. 5.20 the standard deviation conform with what has been seen in the previous plots.

Figures 5.21 and 5.22 are included as a verification on the comments made hithertoo in this section. As restating the observations are felt redundant, no further comment is made, other than the initial steps confirms with the previous statements.

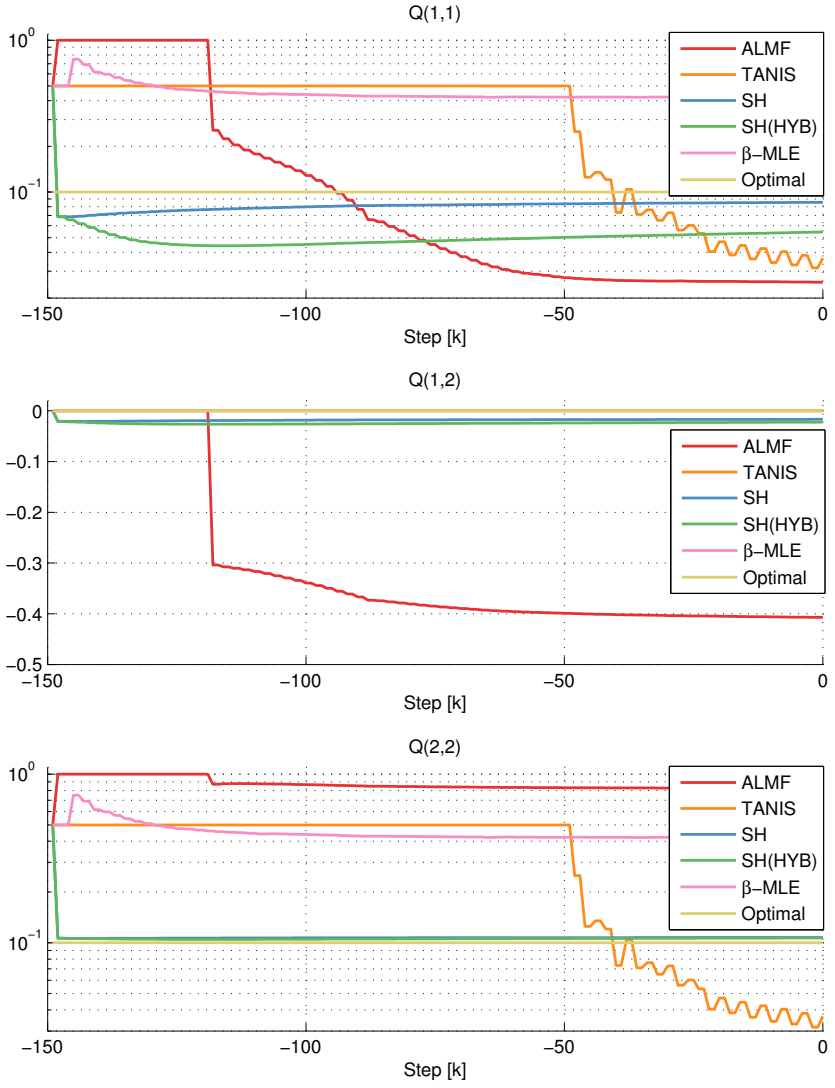

FIGURE 5.22: $\hat{Q}$ during initial steps while estimating R and Q.

5.3.2 Jump in measurement noise

The main task of the estimation schemes are to adapt to changing circumstances, in this case a change in measurement noise covariance. Since an increased performance is dependent on a well tuned Q, and Q can be estimated, the following simulation try to show how the estimation schemes perform when estimating both R and Q, but when R no longer is kept constant.

From the observations made in the previous subsection (5.3.1) the following scheme is proposed. Use the first initial steps to estimate R and Q with the Sage-Husa method.

Then keep the latest Q estimate, and only estimate R using the TANIS/Sage-Husa hybrid.

The proposed method have some underlying assumptions; Q can be approximated to constant and is unknown but has a known structure; measurements with static noise conditions are available for the initial estimation; and there is sufficient initial data for the initial estimates to converge.

Because the structure of Q is assumed known, the average of the estimated elements are distributed with the given weights. In this case we assume $Q = \kappa \boldsymbol{I}_{2x2}$, for some unknown positive constant κ. Which results in the estimated value $\hat{Q} = \frac{1}{2} trace(\hat{Q}_*)\boldsymbol{I}_{2x2}$. $\hat{Q}_*$ is the estimate produced by SH in the initial phase. This method is marked "Proposed" in the figures.

Because the filters using TANIS and ALMF were unstable and did not converge towards the target under stable noise conditions, they have been left out of the figures.

All adaptive filters have initiated with $R = 4$ and $Q(i, i) = .5$.

State estimation

Considering the mean position estimate error in fig. 5.23 it is clear that the aforementioned possible instability for the hybrid / SH combination estimator seen under static conditions, is present here as well. It has a clear result on the positional estimates, which have more errors than the other methods towards the end.

β-MLE is has the second highest mean error and cost J^P. During the step it also has the highest state estimate covariance P.

The most important observation is that the proposed method outperforms the other filters. It has consistently the lowest absolute positional Error estimate, the lowest variance P, and the lowest cost J^P. It also reacts much quicker, although it is not a fair comparison as R is estimated on every step after the initial phase. It shows that even a Q that does not match the target exactly perform better just estimating R, than filters which estimate both.

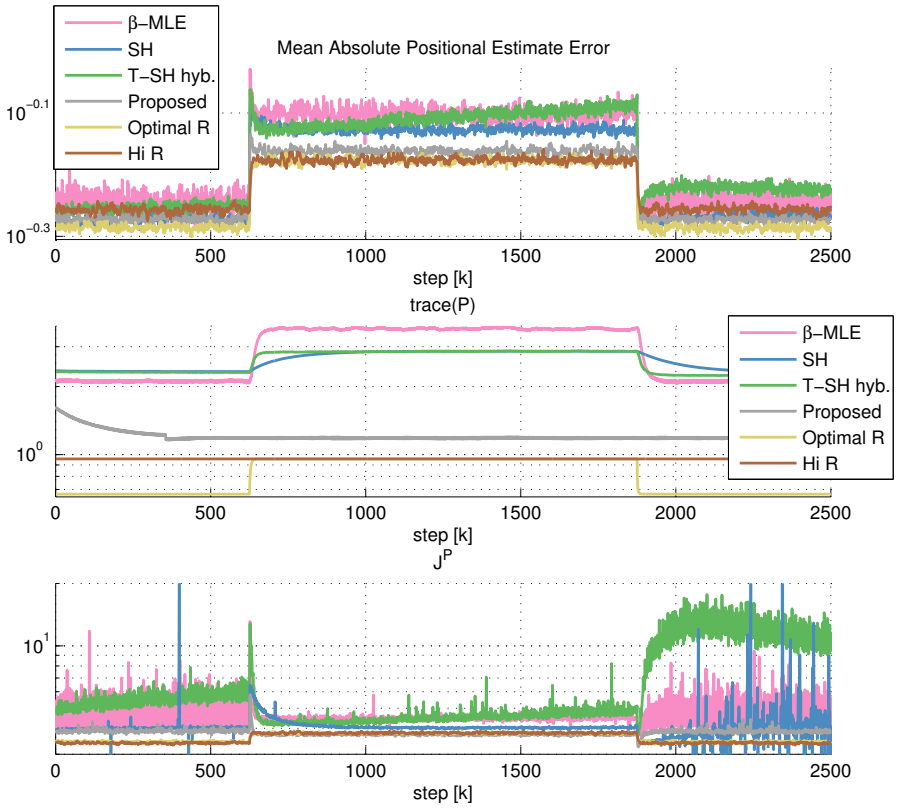

FIGURE 5.23: Absolute state estimate error, state estimate covariance, and J^P during concurrent estimation. Changing R

State Estimate Variance

Figure 5.24 show that none the methods match P with the observed covariance, but all are larger. This indicate, atleast during the simulation, that the adaptive filters are consistent. Moreover the observed estimate covariance is approximatly the same, all closely matching that of the optimal filter.

The proposed method clearly has best matching P, showing that it performs closer to its best than the others. β-MLE, SH, and the hybrid all perform much the same considering only P and the observed values.

The hybrid P values does not increase, which is reassuring. But the change seen in fig. 5.23 is reflected in the observed state estimate covariance, which is steadily increasing during the step.

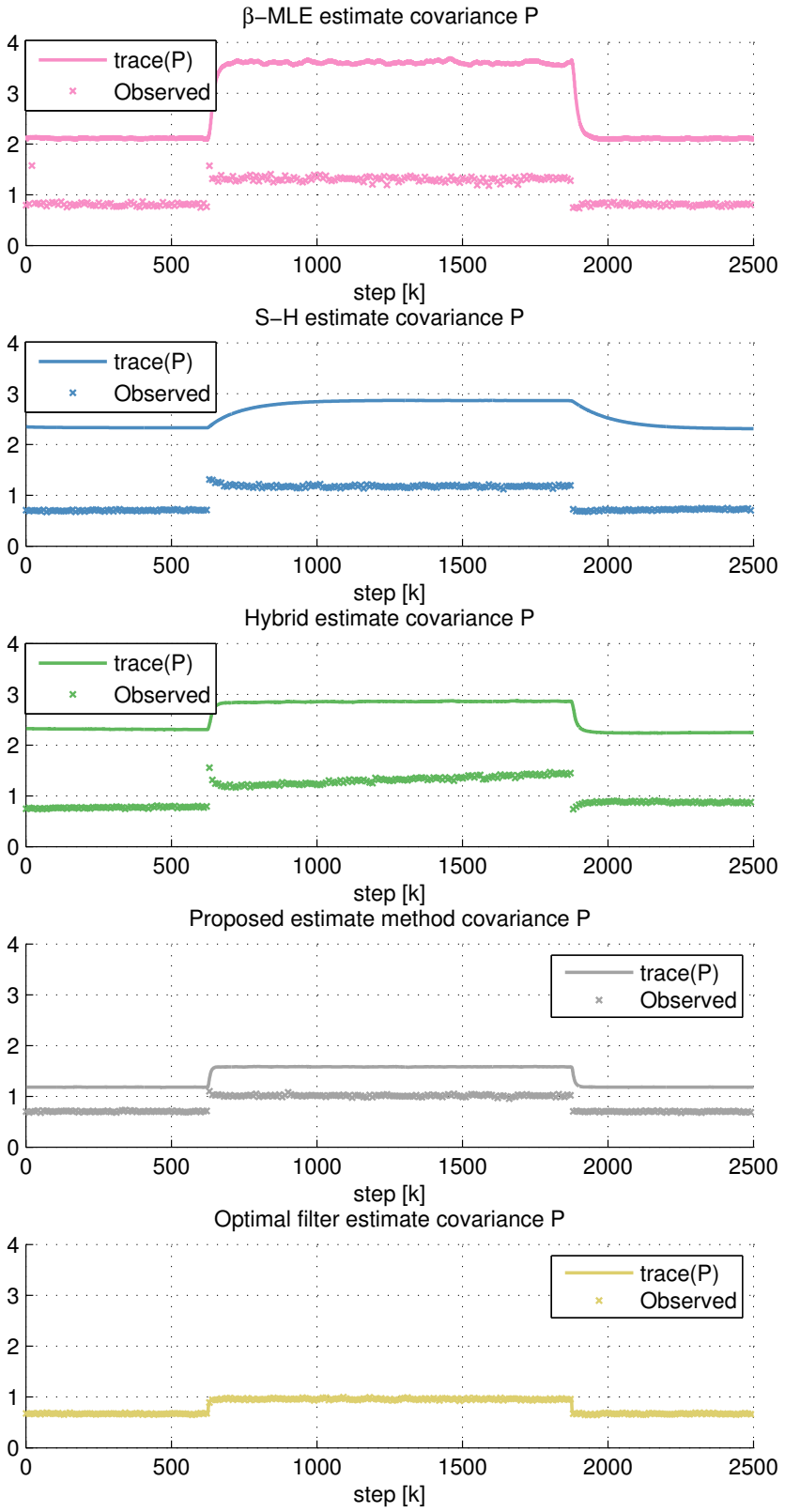

FIGURE 5.24: Aposterioi state estimate variance and the observed state variance

Covariance estimation

In figure 5.25 the mean R estimate has again been divided by the target value. The step in R is clearly seen by the spikes.

The proposed method is the closest match, and converges faster than the others. No doubt because it produces an R estimate each step, rather than every second, is the reason for this. But also looking at figure 5.26 we see that some of the change in R has been estimated as a change in Q. This decrease the percision of $\hat{R}$, same as when Q is fixed to a wrong value in subsection 2.5.4. This error is removed after the step up.

β-MLE and SH Q estimates are the most affected by the sudden increase.

The estimate $\hat{Q}$ used by the hybrid is increasing until the step down. This would account for the increasing state estimate error seen in 5.23; as Q increase more weight is given the measurements. As $\hat{Q}$ already is higher than the target, any further increase cause an unwarrented 'trust' in the incomming measurement.

It should be noted that the Q used by the propose method has been fixed at 0.2. With $\hat{R}$ and $\hat{Q}$ closer than the other estimates, it shows using a preliminary training before fixing Q at its estimated value, increase performance in terms of reducing state and covariance R estimation error. This has been seen in all plots in this section.

For a closer inspection, the more important data is presented in table 5.1.

Method	β-MLE	ALMF	TANIS	Hybrid	SH	Prop.	Optimal
Pos. Est. Err.	0.68	0.773	2.53	0.68	0.642	0.612	0.592
P Error	0.87	1.56	13.8	0.7	0.813	0.263	0.009
$\hat{R}$ Error	1.0	4.03	15.6	1.07	0.743	0.853	0
$\hat{R}$ Std.	1.06	1.7	35.1	1.17	0.86	1.25	0
$\hat{Q}$ Err(diag.)	0.324	0.298	0	0.331	0.197	0.013	0
$\hat{Q}$ Std(diag.)	0.103	0.007	0	0.096	0.063	0	0
$\hat{Q}$ Err(off-diag.)	0	0.404	0	0.112	0.0931	0	0
$\hat{Q}$ Std(off-diag.)	0	0.008	0	0.073	0.0419	0	0

TABLE 5.1: Average performance statistics from concurrent estimation of R and Q. R modeled as wave step. Values has been rounded three decimals.

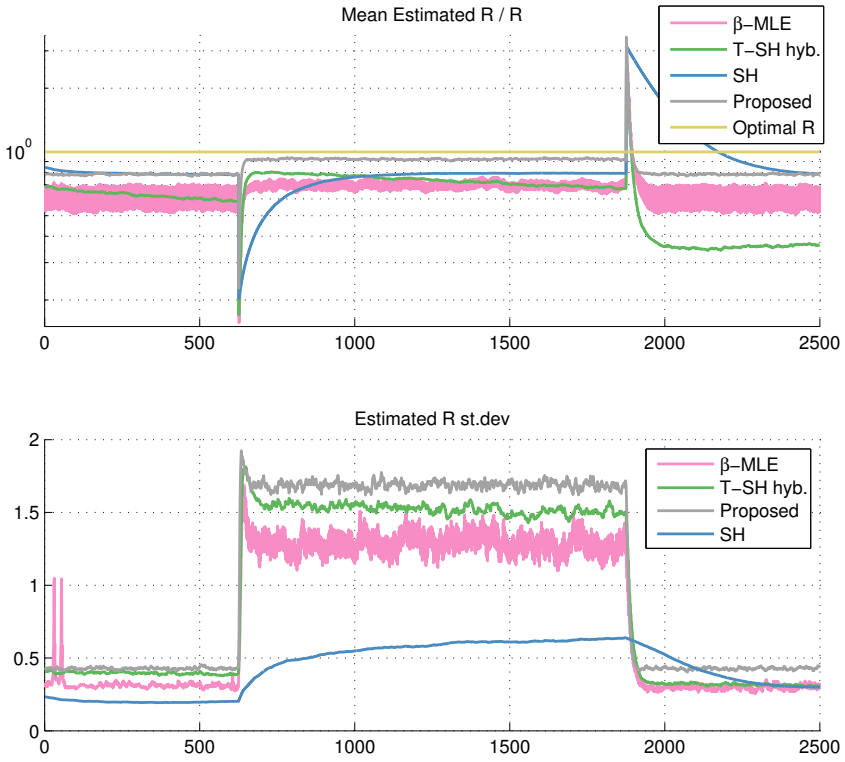
Mean Estimated R / R
β–MLE
T–SH hyb.
SH
Proposed
Optimal R
10^0
0
500
1000
1500
2000
2500
Estimated R st.dev
β–MLE
T–SH hyb.
Proposed
SH
2
1.5
1
0.5
0
0
500
1000
1500
2000
2500

FIGURE 5.25

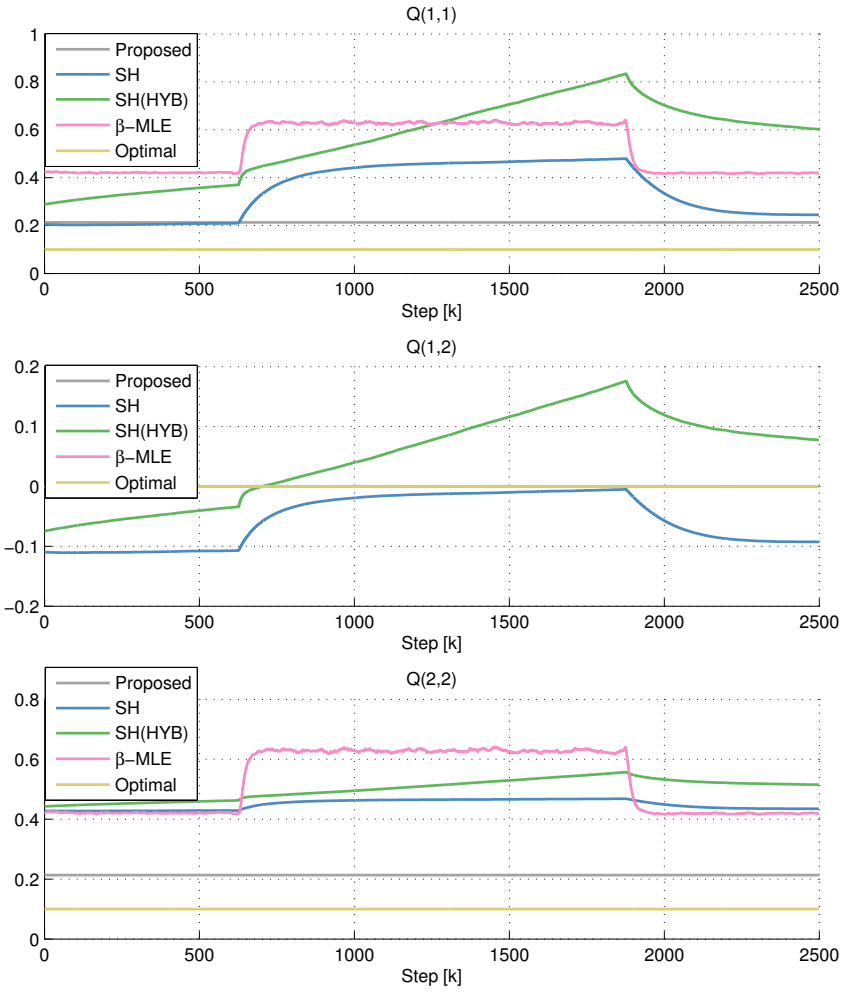
Q(1,1)
Proposed
SH
SH(HYB)
β–MLE
Optimal
1
0.8
0.6
0.4
0.2
0
0
500
1000
1500
2000
2500
Step [k]
Q(1,2)
Proposed
SH
SH(HYB)
β–MLE
Optimal
0.2
0.1
0
−0.1
−0.2
0
500
1000
1500
2000
2500
Step [k]
Q(2,2)
Proposed
SH
SH(HYB)
β–MLE
Optimal
0.8
0.6
0.4
0.2
0
0
500
1000
1500
2000
2500
Step [k]

FIGURE 5.26

5.4 Summary

The filter using the ALMF estimator is prone to accumulating a systematic error in its estimates, even under ideal circumstances. These result from a high variance that frequently produce negative estimates and the way it handles those events. It will eventually settle, but as was seen during concurrent estimation, there is no garantee that it will be better than what starting value is has for the estimates.

The SH method is reliable, abeit slow. It reacts slow to a changing target, but its estimates are accurate and precise. These qualities has made it a favourable addition to the TANIS estimator. SH performed best during the concurrent estimation with static conditions.

The TANIS estimator produce discontinuous estimates, which during concurrent estimation of R and Q, makes the states no longer estimable (if limits for the estimates were not set). When just estimating one statistic, its estimates converge quickly and have a lower variance than the ALMF estimates would have for the same sample size. But because a step up is larger than a step down, it overestimates on average. If it cannot scale exactly to the target it has been shown to ocillate around the target.

As a combination of TANIS and SH, the hybrid has shown improvments on the TANIS method. Compared to the TANIS its estimates has a lower mean error and variance, it tracks a target quicker than SH. There was indications that it might be unstable when estimating R in conjunction with SH estimating Q.

All the aforementioned method are incapable to hit the target covariance if the other filterparameters are incorrect. An error in Q create an error in $\hat{R}$.

β-MLE seemed incapable to balance that error when tested, having a persistant large error in its estimate. Its estimates also had a high variance. When simulated it also had a considerable longer running-time (more than ten fold), showing that this method is unsuided for on-line estimation.

Because the change in R affected the estimators, and because the hybrid showed unstable tendensies, a method was proposed. Namely, first use measurements under unchanging noise conditions to get a better estimate of R and Q. Then use the latest estimate for Q, initialize the hybrid with the latest $\hat{R}$, and let it only estimate R. This method was favourable compared to the other methods when estimating R and Q, with a change in R (see table 5.1).

CHAPTER 6

CONCLUSION

Much work has gone into covariance estimation. As it has been a problem for a long time, many different schemes have been created. Most of the most reliable are to be used with linear systems, after the data is collected. When it comes to sequentially estimating covariances on-line for non-linear systems, the number of viable methods dwindle. In recent years covariance mathcing schemes have been combined with neural-networks, but as the aim of this thesis was do find something akin to "plug-and-play", only general methods were considered. Only estimation of noise covariance R and process noise covariance Q were attepted.

When considering whether the covariances are possible to estimate, it was found that even given that the system was observable, was no garantee for estimability. For filters with incorrectly set filter parameters, the estimates can at best converge at the wrong target. At worst it introduce inconsisencies, making the states unestimable with the adaptive filter in question. Given that the filter stays consistent, measurement noise covariance estimation with a wrongly tuned process noise covariance resulted in filter estimates that were worse than the initially badly tuned filter. This is because the innovation covariance is used as a basis for estimation of both Q and R, meaning that is Q is set too high, the estimate $\hat{R}$ will be too low. This in turn result in an increased error in state estimates, as it given by the ratio between R and Q, not their combined size. With that in mind, it was clear that Q also needed to be estimated. This was shown possible by using the cost J^S, but with more uncertainty than for R.

In a simulation in which both R and Q were to be estimated, the best results were had by first estimating R and Q during unchanging noise characteristics using the Sage-Husa method. Then haing a hybrid of TANIS and SH estimate R, while Q is kept at the last estimate from the initial phase. From this it is concluded that concurrent online estimation of both R and Q gives worse results, than if Q is known or estimated beforehand.

There is alot more work that needs to be done. There is no formal proof of estimator stability, or covariance estimability offered. Further more, the HiPAP is part of a larger system containing different sensors, and the effect of estimation on them has not been investigated. And when done should allow testing on real data. Also refinements in estimation should be investigating, such as a varying step length for the TANIS and hybrid methods.

BIBLIOGRAPHY

[1] R. K. Mehra. Approaches to adaptive filtering. In *1970 IEEE Symposium on Adaptive Processes (9th) Decision and Control*, pages 141–141, Dec 1970. doi: 10.1109/SAP.1970.269992.

[2] Hipap information and documentation. https://www.km.kongsberg.com/hipap. Accessed 10.06.2017.

[3] Hain information and documentation. https://www.km.kongsberg.com/hain. Accessed 19.12.2016.

[4] Patrick Y.C. Hwang Robert Grover Brown. *Introduction to Random Signals and Applied Kalman Filtering.* John Wiley & Sons, Inc., 2012.

[5] Thiagalingam Kirubarajan Yaakov Bar-Shalom, X. Rong Li. *Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software.* Wiley, 2001.

[6] Alex Tremain Nelson. *Nonlinear estimation and modeling of noisy time-series by dual Kalman filtering methods.* PhD thesis, Oregon Graduate Institute of Science and Technology, 2000.

[7] Fred C. Schweppe. *Uncertain Dyamic Systems.* Prentice-Hall, 1973.

[8] Jorge Nocedal and Stephen Wright. *Numerical optimization.* Springer Science & Business Media, 2006.

[9] Y. Baram and T. Kailath. Estimability and regulability of linear systems. *IEEE Transactions on Automatic Control*, 33(12):1116–1121, Dec 1988. ISSN 0018-9286. doi: 10.1109/9.14433.

[10] Mostafa Elzoghby, Usman Arif, Fu Li, and XI Zhi Yu. Investigation of adaptive robust kalman filtering algorithms for gps/dr navigation system filters. *IOP Conference Series: Materials Science and Engineering*, 187(1):012019, 2017. URL http://stacks.iop.org/1757-899X/187/i=1/a=012019.

[11] Kenneth A. Myers and Byron D. Tapley. Adaptive sequential estimation with unknown noise statistics. *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, pages 520 – 523, August 1976.

[12] A. P. Sage and G. W. Husa. Algorithms for sequential adaptive estimation of prior statistics. In *1969 IEEE Symposium on Adaptive Processes (8th) Decision and Control*, pages 61–61, Nov 1969. doi: 10.1109/SAP.1969.269927.