A Semi-Definite Programming Approach for Robust Tracking

Shimrit Shtern · Aharon Ben-Tal

Abstract Tracking problems are prevalent in the present day GPS and video systems. The problem of target tracking is a specific case of dynamic linear system estimation with additive noise. The most widely used filter for these systems is the Kalman Filter (KF). The optimality of the KF and similar Bayesian filters is guaranteed under particular probabilistic assumptions. However, in practice, and specifically in applications such as tracking, these probabilistic assumptions are not realistic; indeed, the system noise is typically bounded and in fact might be adversarial. For such cases, *robust estimation* approaches, such as $\mathcal{H}_\infty$ filtering and set-value estimation, were introduced with the aim of providing filters with guaranteed worst case performance. In this paper we present an innovative approximated set-value estimator (SVE) which is obtained through a Semi-Definite Programming (SDP) problem. We demonstrate that our problem is practically tractable even for long time horizons. The framework is extended to include the case of partially statistical noise, thus combining the KF and SVE frameworks. A variation of this filter which applies a rolling window approach is developed, achieving fixed computational cost per-iteration and coinciding with the classical SVE when window size is one. Finally, we present numerical results that show the advantages of this filter when dealing with adversarial noise and compare the performance of the various robust filters with the KF.

Keywords Robust Optimization · Semidefinite Programming · Estimation of Linear Systems

Mathematics Subject Classification (2000) 90C22 · 90C47 · 93C05

S. Shtern · A. Ben-Tal
The William Davidson Faculty of Industrial Engineering and Management,
Technion - Israel Institute of Technology, Haifa 32000, Israel.
E-mail: shimrits@tx.technion.ac.il.

A. Ben-Tal
E-mail: abental@ie.technion.ac.il.

A. Ben-Tal
CentER Extramural Fellow, CentER, Tilburg University, Tilburg, The Netherlands.

A. Ben-Tal
Visiting Senior Research Scientist, IEOR Department, Columbia University, New York, NY, USA.

1 Introduction

A tracking system aims at revealing properties of moving objects, such as location, velocity, size etc., through sensor measurements. Tracking takes place in both civilian GPS and video security systems, as well as military missile warning and interception, and border surveillance systems.

The linear tracking problem is a specific case of *state estimation* for the following *dynamic linear system*

$$\begin{aligned} \mathbf{x}_t &= \mathbf{F}_t \mathbf{x}_{t-1} + \mathbf{w}_t \\ \mathbf{y}_t &= \mathbf{H}_t \mathbf{x}_t + \mathbf{v}_t. \end{aligned} \tag{1}$$

The first equation in (1) describes the linear state transition between the time steps. Here $\mathbf{x}_t \in \mathbb{R}^r$ is the state vector of the system at time t, $\mathbf{F}_t$ is a $r \times r$ transition matrix between the states at time t, and $\mathbf{w}_t \in \mathbb{R}^r$ is the dynamic system noise at time t. The second equation in (1) describes the observed output at time t, $\mathbf{y}_t \in \mathbb{R}^q$ which is a result of a linear transformation of the state by matrix $\mathbf{H}_t$ of size $q \times r$ plus an additive observation noise vector $\mathbf{v}_t$. It is assumed that the model parameters $\mathbf{F}_t$ and $\mathbf{H}_t$ are known, but both system noise and observation noise, and therefore the system state, are unknown. The goal is to estimate $\mathbf{x}_t$ by an estimator $\widehat{\mathbf{x}}_t$ depending only on the observed output $\mathbf{y}_t$. The transformation, which determines the value of the estimator $\hat{\mathbf{x}}_t$ based on the observed output, is called a filter. The measure used to asses a given filter's performance is usually a function of the estimation error $\|\mathbf{x}_t - \widehat{\mathbf{x}}_t\|$.

This problem has been widely investigated in the literature, and the method in which it is solved depends heavily on the underlying assumptions on the noise vectors $\mathbf{w}$ and $\mathbf{v}$. There are commonly two types of assumptions on the noise: *probabilistic* and *bounded.* They lead to two types of approaches to address the tracking problem.

If probabilistic noise is assumed then a Bayesian filtering approach is used. This approach tries to better estimate the system's state distribution, or expected estimation error, given the measurements. The most widely used filter for this problem was introduced in 1960 by Rudolph Kalman and is known as the *Kalman Filter* (KF) [14]. Given that the noise is white (expected value of zero, uncorrelated) with known covariance, KF is the optimal least mean square error (LMSE) estimator for the Gaussian case, and the optimal Linear LMSE estimator for the general case. Other Bayesian filters, such as the *particle filter*, need less restrictive assumptions but full knowledge of the distribution. However, in practice, and specifically in application such as *tracking*, these probabilistic assumptions are not reasonable, since the system noise is in fact bounded, and might be adversarial. In this case KF does not necessarily ensure good worse case or even good average performance.

Robust Filtering is the approach aimed to cope with the case of bounded adversarial noise, through minimizing the worst case performance. Perhaps the most well known robust method is $\mathcal{H}_\infty$ *filtering*, which tries to find a recursive filter such that a certain performance measure J will be at most γ, a given "robustness parameter", for any realization of the noise over a given finite horizon. J is usually the ratio between the estimation square error and total noise normalized error. The choice of γ is crucial, since a large γ is less robust but for small γ an appropriate filter may fail to exist [29,25]. Moreover, the existence of such an estimator is conditioned on the positive definiteness of some recursion matrices, and a violation of this condition can lead to filter divergence. A more elaborate description of the filter and its application can be found in [18,27,3,29].

A different robust approach is *set-valued estimation* (SVE) proposed by Schweppe in 1968 [26]. The estimate obtained through SVE is not a point but a set, usually an ellipsoid, in which the true state lies. The center of the estimation ellipsoid does not necessarily serve as a good estimator, since no underlying statistical model is assumed. In this context there are two types of uncertainty constraints: *one-step update* which consists of three separate ellipsoid constraints, one for each noise type and one for prior estimation, and an *energy type* which constrains the sum of the entire history of noise norms. Examples of filters constructed for one-step-update uncertainty can be found in [26,15,8], and examples for energy uncertainty can be found in [6,10]. All these filters have a recursive closed form solution, usually obtained through ellipsoidal calculus. Moreover, the filters suggested by Schweppe [26] and El Ghaoui and Calafiore [8](the latter dealing also with model parameter uncertainty), depend on the measurements specific realization while the other previous mentioned filters do not. Furthermore, the closed form solution, obtained for the classic SVE filters, is only well defined if positive definiteness and nullspace inclusion constraints hold for some of the computed recursion matrices. However, as shown in [25], these constraints do not hold in the general case and so the filter recursion can break down.

In this paper we suggest an alternative SVE method, which uses individual ellipsoid bounds on each of the noise vectors at each point in time, instead of the classical one-step-update and energy bounds. These constraints, which we term as *full recollection* noise bound, lead to a more accurate worst case estimate. Thus, our model can be viewed, or interpreted, as an approximated $\mathcal{H}_1$ *filter*, where the filter formulation is independent of the measurements' realization and can be found recursively. This recursive filter is developed according to robust optimization methodology, presented in [5], and obtained numerically by using *Semi-Definite Programming* (SDP). The resulting SDP optimization problems, for finding a worst case solution and the optimal filter matrices, are solved by an interior point method, which we demonstrate to be practically tractable for both problem. Our method is well defined for any time horizon length and no positive definite assumptions on the recursion matrices are needed. We also present a version with fixed computational cost, using a rolling window updating scheme. In this scheme we construct a recursive matrix, which defines the estimation ellipsoid, and use its inverse in the following iterations. However, when implementing the algorithm we do not need to find the matrix inverse explicitly, except to retrieve the worst case noise realization, which can be achieved by using the Penrose-Moore pseudoinverse instead. In this version the solution obtained by using a rolling window of size one is equivalent to using a one-step-update uncertainty bound, similar to the one presented in [6].

The KF also has a deterministic interpretation which is derived from the solution to a regularized least square (RLS) problem [31,25]. Robust KF can be derived by solving a minimax problem pertaining to this RLS. In that respect, SVE filters can be viewed as the solution of the so called *robust counterpart* of the RLS formulation with noise uncertainty. In this paper we suggest a different way to unify the SVE and KF frameworks. We present a generalized SVE which includes both probabilistic and bounded noise. This model uses SDP to approximate a filter which minimizes the worst case *expected* estimation error, and is reduced to either the KF or SVE when the noise is either entirely probabilistic or bounded, respectively.

The rest of the paper is organized as follows: Sect. 2 presents the KF model and its disadvantage in dealing with bounded adversarial noise. Sect. 3 describes the worst case optimization problem and its SDP approximation. In section 4 a few filtering models and their solutions are described, including partly affine and fully affine filters, mixed probabilistic and bounded noise optimal filter, and rolling horizon affine filters. Finally, Sect. 5 contains numerical examples of a tracking problem, comparing the performances of the various robust filters and the KF.

We will use the following notation throughout the paper. For any $m \in \mathbb{N}$ the set $\{1, \dots, m\}$ is denoted by $[m]$. For any vector set $\{\mathbf{a}_1, \dots, \mathbf{a}_m\}$ of column vectors $\mathbf{a}_i \in \mathbb{R}^{n_i}$ we use the notation $\mathbf{a} = [\mathbf{a}_1; \dots; \mathbf{a}_m]$ to denote a vertical concatenation of these vectors into a single vector $\mathbf{a}$ of dimension $n = \sum_i n_i$. The matrix $\mathrm{Diag}(\mathbf{a})$ is diagonal matrix whose diagonal elements correspond with vector $\mathbf{a}$. $\mathbf{a}_i \in \mathbb{R}^{n_i}$ denotes the ith partition element of vector $\mathbf{a}$ while $a_i \in \mathbb{R}$ denotes its ith coordinate. Notation $\|\mathbf{a}\|$ indicates the Euclidean norm of vector $\mathbf{a}$, unless stated otherwise. Let $\mathbf{A}$ be a matrix, then $\mathbf{A}'$, $\mathrm{Tr}(\mathbf{A})$, $\mathrm{rank}(\mathbf{A})$ denote its transpose, trace and rank respectively. Let

$\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times m}$, then $\mathbf{A}'\mathbf{B} \in \mathbb{R}^{n\times n}$ denotes the standard matrix product, $\mathbf{A} \circ \mathbf{B} \triangleq \mathrm{Tr}(\mathbf{A}'\mathbf{B}) \in \mathbb{R}$ denotes the matrices Frobenius inner product, and $\mathbf{A} \bullet \mathbf{B} \in \mathbb{R}^{n\times m}$ denotes the component wise, or Hadamard, product, resulting in a matrix of the same dimensions. We denote $\mathbf{A} \succeq 0$ if $\mathbf{A}$ is a symmetric positive-semi definite (PSD) matrix. A_{ij} denoted the element in row i and column j of the matrix, $\mathbf{A}_{i\cdot}$ denotes the ith row of the matrix as a row vector and $\mathbf{A}_{\cdot j}$ denotes the column j of the matrix as a column vector. Finally, $\mathbf{e}$ denotes a vector of all ones, $\mathbf{0}$ denotes matrix of all zeros, $\mathbf{I}$ denotes the identity matrix, and $\boldsymbol{\mathcal{I}}_i$ denotes a square matrix whose i diagonal block corresponds with the identity matrix and all other elements are zero.

2 Problem Formulation and Bayesian Filtering

In this section we will discuss the assumptions and disadvantages of Bayesian estimation, and more specifically the KF, with respect to the estimation problem of system (1).

In classic Bayesian estimation it is assumed that both $\mathbf{w}_t$ and $\mathbf{v}_t$ are independent random variables, which are referred to as noise. The noise is assumed to be white and independent of the state. Typically, both $\mathbf{w}_t$ and $\mathbf{v}_t$ are assumed to be Gaussian with distributions $\mathbf{w}_t \sim N(0, \mathbf{Q}_k)$ and $\mathbf{v}_t \sim N(0, \mathbf{R}_k)$. An unbiased estimator is considered one for which $\mathbb{E}(\widehat{\mathbf{x}}_t) = \mathbf{x}_t$. We will denote the covariance matrix of such an estimator by $\mathbf{P}_t \equiv \mathrm{Cov}(\widehat{\mathbf{x}}_t - \mathbf{x}_t)$.

Given these assumptions the KF produces an unbiased *least mean square error* (LMSE) estimator, i.e., it is an optimal filter with regard to the objective function of minimizing $\mathbb{E}(\|\widehat{\mathbf{x}}_t - \mathbf{x}\|^2)$, or equivalently $\mathrm{Tr}(\mathbf{P}_t)$. Given $\hat{\mathbf{x}}_{t-1}$, an unbiased estimator for time $t-1$ with covariance matrix of $\mathbf{P}_{t-1}$, the KF updates the estimator in two stages: the prediction stage and the updating stage. In the prediction stage, $\widehat{\mathbf{x}}_{t|t-1}$, an a priori (predicted) estimator of $\mathbf{x}_t$, and an estimated observed output $\widehat{\mathbf{y}}_t$ are constructed as follows:

$$\begin{aligned} \widehat{\mathbf{x}}_{t|t-1} &= \mathbf{F}_t \widehat{\mathbf{x}}_{t-1} \\ \widehat{\mathbf{y}}_t &= \mathbf{H}_t \widehat{\mathbf{x}}_{t|t-1}. \end{aligned}$$

In the updating stage, the a posteriori estimator is obtained by correcting the a priori estimator $\widehat{\mathbf{x}}_{t|t-1}$ using the purified output $\mathbf{z}_t$ and gain matrix $\mathbf{K}_t$ as follows:

$$\begin{aligned} \mathbf{z}_t &= \mathbf{y}_t - \widehat{\mathbf{y}}_t \\ \widehat{\mathbf{x}}_t &= \widehat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t \mathbf{z}_t. \end{aligned}$$

The gain matrix $\mathbf{K}_t$ balances the weight of the prior estimation with the current observed output, and the recursive formulation for the optimal $\mathbf{K}_t$ is then given by the following equation:

$$\mathbf{K}_t^* = \mathbf{P}_{t|t-1} \mathbf{H}_t' \mathbf{S}_t^{-1}$$

where $\mathbf{S}_t = \mathrm{Cov}(\mathbf{z}_t)$ and $\mathbf{P}_{t|t-1}$ is the a priori estimation error covariance matrix.

The KF belongs to the family of linear filters. Linear filters determine the estimator's value as an affine function of the observed output history, or equivalently, of the purified output history $\mathbf{Z}_t = (\mathbf{z}_1, \ldots, \mathbf{z}_t)$. Linear filters are the most widely used since they maintain the linearity of the problem and are very easily implemented. The KF is the optimal linear filter for this setting, meaning that the KF has a lower expected square estimation error (MSE) value than all other linear filters, even if the noise is not Gaussian but still white. An extensive description of the various types of KF variations and its limitations are given in [29], from which we will discuss a few.

There are extensions of KF which deal with correlation between process and measurement noise and correlation between noise at different time steps. The Basic KF is sensitive to modeling errors and implementation errors, as described early on by [12] and [23]. The filter can be made more robust by using such heuristics as adding process noise and fading memory of “old” measurements.

Though KF is the optimal linear filter, when dealing with non-Gaussian type noise large variances may occur by applying a linear filter, as shown in [16,17,30]. The authors address this problem by using a score function which is dependent both on the probability density function and the realization.

The main problem with creating a robust KF is that we assume knowledge of not only the model but indeed the noise distribution. The filter might be far from optimal when the noise is not Gaussian, correlated, uni-modal, non-zero mean, or has a different variance than what we assumed in the model. The robustness of KF was tackled by many researchers, mostly dealing with the steady state filter. In [24] the authors showed KF is mildly sensitive to variation in the noise. Some examples of robust KF, which deal with model and noise uncertainty, and give some guaranty on the estimation error variance can be found in [32,9,33,7]. The KF and its variations are still widely used filters in tracking, navigation, GPS software, image processing and robotics [1,2].

Let us now examine the assumptions of classic estimation. The first problem we encounter is the assumption white uncorrelated noise with known moments which is independent of the system state. Let us consider the following implementation of estimation to what is known as the $\alpha - \beta$ model for tracking, which assumes $\mathbf{x}_t$ is a state vector consisting of location and velocity and $\mathbf{w}_t$ is represented by a linear transformation of the acceleration. The representation in the one dimensional case is:

$$\mathbf{x}_t = \begin{bmatrix} l_t \\ s_t \end{bmatrix}, \ \mathbf{F}_t \quad = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \ \mathbf{w}_t = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} a_t$$

where $l_t \in \mathbb{R}$ is the location,$s_t \in \mathbb{R}$ is the velocity and $a_t \in \mathbb{R}$ is the unknown acceleration of the object. For most cases a_t is bounded and its distribution is not necessarily known. Furthermore, given a purposeful object with bounded velocity, the a_t are dependent on the state as well as on each other.

Hence, in this case, the KF underlying assumption of uncorrelated white noise with zero mean is not valid. In fact, in many cases the true distributions are unknown and can not be used to obtain a score function or construct an appropriate particle filter. Furthermore, if we assume we only have knowledge of the bounds on state and noise variables then the measure the KF optimizes (expected square error) is not appropriate. In this setting, not only is the worst case performance of the KF unknown but it also may occur with high probability and be significantly worse than the average performance; moreover, the true average performance might be worse than the theoretical measure indicates.

In order to demonstrate this point consider the following tracking example: An object is tracked moving only in one dimension, it has an initial uncertain state vector, with location of at most $20m$ around the zero point and velocity of at most $10\frac{m}{sec}$. The object’s maximum acceleration is $2\frac{m}{sec^2}$ and the error of the sensor which measured the sensor location is at most $20m$. In applying the KF we therefore assume that the probability of being below the maximal errors and noise specified is 99% (three sigma), and derive the appropriate Gaussian distributions. Now let us consider a scenario shown in Figure 1 where both acceleration and measurement error oscillate between their extreme values. The KF produces an estimation which is as far as $47m$ from the actual object and a velocity estimation error of up to $17\frac{m}{sec}$. Notice that a

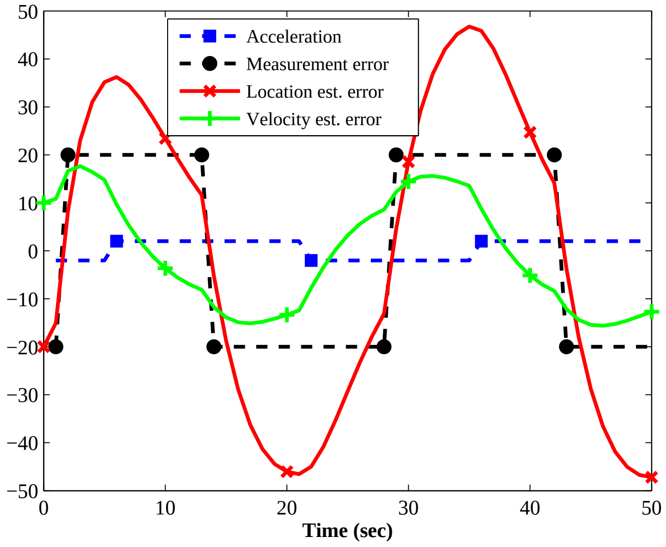


Fig. 1 KF Performance for Tracking Problem Example

naive location estimation, which takes the observation as the estimator, would have reduced the maximal location error to only $20m$.

3 Worst Case Approximation

Consider a setting in which the noise is unknown but bounded. The assumptions presented for this setting are alternative to the ones given above for the classic Bayesian estimation, and motivate the SVE approach.

Consider $\mathbf{w}_t$ and $\mathbf{v}_t$ which are constrained in uncertainty sets $\mathcal{W}_t$ and $\mathcal{V}_t$ respectively such that:

$$\mathcal{W}_t = \left\{ \mathbf{w}_t \in \mathbb{R}^r \;\middle|\; \|\mathbf{w}_t\|_{\mathbf{Q}_t^{-1}} = \sqrt{\mathbf{w}_t'\mathbf{Q}_t^{-1}\mathbf{w}_t} \leq \alpha_t \right\}$$
$$\mathcal{V}_t = \left\{ \mathbf{v}_t \in \mathbb{R}^q \;\middle|\; \|\mathbf{v}_t\|_{\mathbf{R}_t^{-1}} = \sqrt{\mathbf{v}_t'\mathbf{R}_t^{-1}\mathbf{v}_t} \leq \beta_t \right\}$$

where $\mathbf{R}_t$ and $\mathbf{Q}_t$ are positive definite (PD) symmetric matrices. This form is referred to as "Mahalanobis Distance" and it is a normalized probability distance. There are situations where $\mathbf{Q}_t$ itself is PSD and not PD, due to the fact that $\mathbf{w}_t$ is actually a result of a linear transformation of the actual noise $\mathbf{a}_t \in \mathbb{R}^{\tilde{r}}$ for $\tilde{r} < r$, e.g., when $\mathbf{a}_t$ is the acceleration which affects both location and velocity, as we have seen in the previous section. In this case we can write $\mathbf{w}_t = \mathbf{G}_t\mathbf{a}_t$ and so $\mathbf{Q}_t = \mathbf{G}_t'\mathbf{Q}_t^{\mathbf{a}}\mathbf{G}_t$ where $\mathbf{Q}_t^{\mathbf{a}}$ is PD. So we will, from now on, refer to this case and define

$$\mathcal{W}_t^{\mathbf{a}} = \left\{ \mathbf{a}_t \in \mathbb{R}^{\tilde{r}} \;\middle|\; \|\mathbf{a}_t\|_{(\mathbf{Q}_t^{\mathbf{a}})^{-1}} = \sqrt{\mathbf{a}_t'(\mathbf{Q}_t^{\mathbf{a}})^{-1}\mathbf{a}_t} \leq \alpha_t \right\}$$

In this context, since there is no probabilistic information, we are interested in finding a filter which minimizes the worst case, rather than the mean, square estimation error. In order to find such a filter we must first solve the problem of calculating the worst case estimation error given a filter. This is equivalent to finding the radius of a ball around the filter's estimation in which the true system state is guaranteed to exist.

A filter is a general function of the entire purified output history $\mathbf{Z}_t = (\mathbf{z}_1, \ldots, \mathbf{z}_t)$ which produces an estimate $\widehat{\mathbf{x}}_{t+1}$. For simplicity, we restrict our discussion to affine mappings resulting in a general linear filter, which takes the form

$$\widehat{\mathbf{x}}_{t+1} = \mathbf{F}\widehat{\mathbf{x}}_t + \mathbf{k}_t + \sum_{\tau=1}^{t} \mathbf{K}_t^{\tau}\mathbf{z}_{\tau}, \tag{2}$$

where $\mathbf{k}_t \in \mathbb{R}^r$ is an update term which is independent of the output and $\{\mathbf{K}_t^{\tau}\}_{\tau=1}^{t}$, such that $\mathbf{K}_t^{\tau} \in \mathbb{R}^{r\times q}$, are coefficient matrices for the output dependent update. Linear filters are widely used in literature, since they enable constructing a simpler model and solution, and under some assumptions, as in the case of KF, are proven to be optimal.

We will start by discussing the formulation and disadvantages of the one-step update noise bound and proceed to discuss the full recollection noise bound. When implementing a one-step update noise bound, such as presented in [26,6], one must choose a filter which maintains the independence between time steps. Therefore, only filters which depend solely on the last purified output, similarly to the KF, may be considered, i.e.,

$$\hat{\mathbf{x}}_t = \mathbf{F}\hat{\mathbf{x}}_{t-1} + \mathbf{K}_t^t\mathbf{z}_t. \tag{3}$$

We also assume that the estimator $\widehat{\mathbf{x}}_{t-1}$, from the previous time period, is known to guaranty a maximum square deviation of ℓ_{t-1} from the previous (unknown) state, so that:

$$\mathcal{X}_{t-1} = \{\mathbf{x}_{t-1} \mid \|\widehat{\mathbf{x}}_{t-1} - \mathbf{x}_{t-1}\| \leq \ell_{t-1}\}.$$

Under these assumptions calculating the worst case estimation error is equivalent to solving the following semi-infinite optimization problem.

$$\begin{aligned}
&\min_{\tau} \tau \\
&s.t. \\
&\left.\begin{aligned}
\|\widehat{\mathbf{x}}_t - \mathbf{x}_t\| &\leq \tau \\
\widehat{\mathbf{x}}_t &= \mathbf{F}_t\widehat{\mathbf{x}}_{t-1} + \mathbf{K}_t^t\mathbf{z}_t \\
\mathbf{x}_t &= \mathbf{F}_t\mathbf{x}_{t-1} + \mathbf{w}_t \\
\mathbf{w}_t &= \mathbf{G}_t\mathbf{a}_t \\
\mathbf{y}_t &= \mathbf{H}_t\mathbf{x}_t + \mathbf{v}_t \\
\mathbf{z}_t &= \mathbf{y}_t - \mathbf{H}_t\mathbf{F}_t\widehat{\mathbf{x}}_{t-1}
\end{aligned}\right\}
\begin{matrix} \forall \mathbf{a}_t \in \mathcal{W}_t^{\mathbf{a}} \\ \forall \mathbf{v}_t \in \mathcal{V}_t \\ \forall \mathbf{x}_{t-1} \in \mathcal{X}_{t-1} \end{matrix}
\end{aligned} \tag{4}$$

We can now eliminate all the equations by replacing each LHS variable with its RHS equivalent to obtain:

$$\begin{aligned}
\mathbf{z}_t &= \mathbf{H}_t\mathbf{F}_t(\mathbf{x}_{t-1} - \widehat{\mathbf{x}}_{t-1}) + \mathbf{H}_t\mathbf{G}_t\mathbf{a}_t + \mathbf{v}_t \\
\widehat{\mathbf{x}}_t &= \mathbf{F}_t\widehat{\mathbf{x}}_{t-1} + \mathbf{K}_t^t\mathbf{H}_t\mathbf{F}_t(\mathbf{x}_{t-1} - \widehat{\mathbf{x}}_{t-1}) + \mathbf{K}_t^t\mathbf{H}_t\mathbf{G}_t\mathbf{a}_t + \mathbf{K}_t^t\mathbf{v}_t \\
\widehat{\mathbf{x}}_t - \mathbf{x}_t &= (\mathbf{F}_t - \mathbf{K}_t^t\mathbf{H}_t\mathbf{F}_t)(\widehat{\mathbf{x}}_{t-1} - \mathbf{x}_{t-1}) + (\mathbf{K}_t\mathbf{H}_t\mathbf{G}_t - \mathbf{G}_t)\mathbf{a}_t + \mathbf{K}_t^t\mathbf{v}_t \\
\|\widehat{\mathbf{x}}_t - \mathbf{x}_t\|_2 &= \left\|(\mathbf{I} - \mathbf{K}_t^t\mathbf{H}_t)\mathbf{F}_t(\widehat{\mathbf{x}}_{t-1} - \mathbf{x}_{t-1}) + (\mathbf{K}_t^t\mathbf{H}_t - \mathbf{I})\mathbf{G}_t\mathbf{a}_t + \mathbf{K}_t^t\mathbf{v}_t\right\|
\end{aligned}$$

Moreover, defining $\boldsymbol{\delta}_t = \widehat{\mathbf{x}}_t - \mathbf{x}_t$, we can treat the problem as finding the maximal $\|\boldsymbol{\delta}_t\|$ where,

$$\|\boldsymbol{\delta}_t\| = \left\|(\mathbf{I} - \mathbf{K}_t^t\mathbf{H}_t)\mathbf{F}_t\boldsymbol{\delta}_{t-1} + (\mathbf{K}_t^t\mathbf{H}_t - \mathbf{I})\mathbf{G}_t\mathbf{a}_t + \mathbf{K}_t^t\mathbf{v}_t\right\|.$$

Therefore, the condition $\boldsymbol{\delta}_{t-1} \in \mathit{\Delta}_{t-1} = \{\boldsymbol{\delta}_{t-1} : \|\boldsymbol{\delta}_{t-1}\| \leq \ell_{t-1}\}$ can be used instead of $\mathbf{x}_{t-1} \in \mathcal{X}_{t-1}$ to obtain the same worst case solution. Furthermore, since we are dealing with one time step, we can omit the t or $t-1$ indexes, for ease of notation, and denote $\mathcal{W}^{\mathbf{a}} \equiv \mathcal{W}^{\mathbf{a}}_t$, $\mathcal{V} \equiv \mathcal{V}_t$, $\mathit{\Delta} \equiv \mathit{\Delta}_t$, $\mathbf{a} \equiv \mathbf{a}_t$, $\mathbf{v} \equiv \mathbf{v}_t$, $\boldsymbol{\delta} \equiv \boldsymbol{\delta}_{t-1}$, $\alpha \equiv \alpha_t$, $\beta \equiv \beta_t$, $\ell \equiv \ell_t$, $\mathbf{G} \equiv \mathbf{G}_t$, $\mathbf{Q_a} \equiv \mathbf{Q}^{\mathbf{a}}_t$, $\mathbf{R} \equiv \mathbf{R}_t$, $\mathbf{K} \equiv \mathbf{K}^t_t$, $\mathbf{H} \equiv \mathbf{H}_t$, and $\mathbf{F} \equiv \mathbf{F}_t$.

$$\begin{aligned} \max_{\mathbf{a},\mathbf{v},\boldsymbol{\delta}} \quad & \|(\mathbf{I}-\mathbf{KH})\mathbf{F}\boldsymbol{\delta} + (\mathbf{KH}-\mathbf{I})\mathbf{Ga} + \mathbf{Kv}\| \\ s.t. \quad & \mathbf{a}'\mathbf{Q}_{\mathbf{a}}^{-1}\mathbf{a} \leq \alpha^2 \\ & \mathbf{v}'\mathbf{R}^{-1}\mathbf{v} \leq \beta^2 \\ & \boldsymbol{\delta}'\boldsymbol{\delta} \leq \ell^2 \end{aligned}$$

It is clear that the maximum is obtained since this is a continuous function in the uncertain variables $\mathbf{w}, \mathbf{v}$ and $\boldsymbol{\delta}$, which all lie within compact sets. But this problem is generally difficult to solve, since we are trying to maximize a convex quadratic function of these variables.

The problem of finding the best linear one-step filter for this measure can be formulated as a minimax problem of the following form:

$$\begin{aligned} \min_{\mathbf{K}} \max_{\mathbf{a},\mathbf{v},\boldsymbol{\delta}} \quad & \|(\mathbf{I}-\mathbf{HF})\mathbf{F}\boldsymbol{\delta} + (\mathbf{KH}-\mathbf{I})\mathbf{Ga} + \mathbf{Kv}\| \\ s.t. \quad & \mathbf{a}'\mathbf{Q}_{\mathbf{a}}^{-1}\mathbf{a} \leq \alpha^2 \\ & \mathbf{v}'\mathbf{R}^{-1}\mathbf{v} \leq \beta^2 \\ & \boldsymbol{\delta}'\boldsymbol{\delta} \leq \ell^2 \end{aligned} \tag{5}$$

The assumptions that $\boldsymbol{\delta}$ is constrained only by the norm obtained by the optimization for the previous time step, allows us to create independence between time step decisions and maintain both structure and dimension of the problem. The problem with this one-step update approach is its conservativeness, since it does not necessarily reflect the real dynamics of the system. The worst case $\|\boldsymbol{\delta}_t\|$ is not necessarily generated by the same noise vector as the worst case $\|\boldsymbol{\delta}_{t-1}\|$. However, problem (5) implicitly assumes the existence of such a noise vector, and therefore might protect us against, not only unlikely, but actually infeasible events. So even for this simple filter structure there is an advantage in finding the true worst case solution, based on the entire noise history.

Therefore, we will now consider an alternative problem of computing the worst case estimation error based on the entire noise history. Since we are no longer restricted to a dependency on only one time-step, we revert to discussing general linear filters defined in equation (2). Denoting the uncertainty vector at time T as $\boldsymbol{\omega}_T = [\boldsymbol{\delta}_0^1; \ldots, \boldsymbol{\delta}_0^p; \mathbf{a}_1; \ldots; \mathbf{a}_T; \mathbf{v}_1; \ldots, \mathbf{v}_T]$ where $\{\boldsymbol{\delta}_0^i\}_{i=1}^p$ is some partition of the initial estimation error $\boldsymbol{\delta}_0$, we can define the corresponding uncertainty set as:

$$\mathit{\Omega}_T = \left\{ \boldsymbol{\omega}_T = [\boldsymbol{\delta}_0^1; \ldots, \boldsymbol{\delta}_0^p; \mathbf{a}_1; \ldots; \mathbf{a}_T; \mathbf{v}_1; \ldots, \mathbf{v}_T] \;\middle|\; \begin{aligned} \mathbf{a}_t'\mathbf{Q}_{\mathbf{a}}^{-1}\mathbf{a}_t &\leq \alpha^2 & t = 1, \ldots, T \\ \mathbf{v}_t' R^{-1}\mathbf{v}_t &\leq \beta^2 & t = 1, \ldots, T \\ {\boldsymbol{\delta}_0^i}'\boldsymbol{\delta}_0^i &\leq \ell_i^2 & i = 1, \ldots, p \end{aligned} \right\}. \tag{6}$$

We refer to this uncertainty set as a *full recollection* noise bound, since it maintains the entire noise history information, and therefore allows only feasible realizations to be considered.

The resulting estimation error vector, induces by applying linear filter (2), is then given by:

$$\begin{aligned}\boldsymbol{\delta}_T = \widehat{\mathbf{x}}_t - \mathbf{x}_t &= \mathbf{F}(\widehat{\mathbf{x}}_{t-1} - \mathbf{x}_{t-1}) + \mathbf{k}_t + \sum_{\tau=1}^{t} \mathbf{K}_\tau^t(\mathbf{HGa}_\tau + \mathbf{v}_\tau - \mathbf{HF}(\widehat{\mathbf{x}}_{\tau-1} - \mathbf{x}_{\tau-1})) - \mathbf{Ga}_t \\ &= \mathbf{k}_t + (\mathbf{K}_t^t\mathbf{HG} - \mathbf{G})\mathbf{a}_t + \mathbf{K}_t^t\mathbf{v}_t + (\mathbf{F} - \mathbf{K}_t^t\mathbf{HF})(\boldsymbol{\delta}_{t-1}) \\ &+ \sum_{\tau=1}^{t-1} \mathbf{K}_t^\tau(\mathbf{HGa}_\tau + \mathbf{v}_\tau - \mathbf{HF}(\boldsymbol{\delta}_{\tau-1})).\end{aligned}$$

We denote $\mathbf{K}_T = [\mathbf{k}_T, \mathbf{K}_T^1, \ldots, \mathbf{K}_T^T]$ as the gain matrix at time T. The worst case estimation error, assuming known gains $\{\mathbf{K}_t\}_{t=1}^T$ and constant problem parameters $\mathbf{F}, \mathbf{H}, \mathbf{G}$, is given as a solution of the following optimization problem,

$$\max_{\boldsymbol{\omega}_T \in \Omega_T} \left\| \mathbf{c}_T + \mathbf{A}_T^{\boldsymbol{\delta}_0}\boldsymbol{\delta}_0 + \sum_{t=1}^{T}(\mathbf{A}_T^{\mathbf{a}_t}\mathbf{a}_t + \mathbf{A}_T^{\mathbf{v}_t}\mathbf{v}_t) \right\|, \tag{7}$$

where matrices $\mathbf{A}_T^{\boldsymbol{\delta}_0}$, $\{\mathbf{A}_T^{\mathbf{a}_t}\}_{t=1}^T$, $\{\mathbf{A}_T^{\mathbf{v}_t}\}_{t=1}^T$ and vector $\mathbf{c}_T$, given in (8), are computed recursively using formula (3).

$$\begin{aligned} \mathbf{A}_T^{\mathbf{a}_t} &= \begin{cases} (\mathbf{K}_T^T\mathbf{HG} - \mathbf{G}) & t = T,\ T \geq 1 \\ \mathbf{FA}_{T-1}^{\mathbf{a}_t} + \mathbf{K}_T^T(-\mathbf{HFA}_{T-1}^{\mathbf{a}_t}) + \mathbf{K}_T^t\mathbf{HG} - \sum\limits_{\tau=t+1}^{T-1} \mathbf{K}_T^\tau\mathbf{HFA}_{\tau-1}^{\mathbf{a}_t} & 1 \leq t < T \\ \mathbf{0} & \text{otherwise} \end{cases} \\ \mathbf{A}_T^{\mathbf{v}_t} &= \begin{cases} \mathbf{K}_T^T & t = T,\ T \geq 1 \\ \mathbf{FA}_{T-1}^{v_t} + \mathbf{K}_T^T(-\mathbf{HFA}_{T-1}^{\mathbf{v}_t}) + \mathbf{K}_T^t - \sum\limits_{\tau=t+1}^{T-1} \mathbf{K}_T^\tau\mathbf{HFA}_{\tau-1}^{\mathbf{v}_t} & 1 \leq t < T \\ \mathbf{0} & \text{otherwise} \end{cases} \\ \mathbf{A}_T^{\delta_0} &= \begin{cases} (\mathbf{FA}_{T-1}^{\boldsymbol{\delta}_0} - \sum\limits_{t=1}^{T} \mathbf{K}_T^t\mathbf{HFA}_{t-1}^{\boldsymbol{\delta}_0}) & T \geq 1 \\ \mathbf{I} & \text{otherwise} \end{cases} \\ \mathbf{c}_T &= \begin{cases} \mathbf{k}_T + \mathbf{Fc}_{T-1} - \sum\limits_{t=1}^{T} \mathbf{K}_T^t\mathbf{HFc}_{t-1} & T \geq 1 \\ \mathbf{0} & \text{otherwise.} \end{cases} \end{aligned} \tag{8}$$

For a one-step update compatible filter, in the form of (3), an even simpler formulation is obtained:

$$\max_{\boldsymbol{\omega}_T \in \Omega_T} \left\| \sum_{t=1}^{T} \mathbf{L}_{t+1}^T((\mathbf{K}_t^t\mathbf{H} - \mathbf{I})\mathbf{Ga}_t + \mathbf{K}_t^t\mathbf{v}_t)\ + \mathbf{L}_1^T\boldsymbol{\delta}_0 \right\| \tag{9}$$

where

$$\mathbf{L}_{\underline{t},\bar{t}} = \begin{cases} \prod_{t=\underline{t}}^{\bar{t}}(\mathbf{F} - \mathbf{K}_t^t\mathbf{HF}) & \underline{t} \leq \bar{t} \\ \mathbf{I} & \underline{t} > \bar{t}. \end{cases}$$

Obviously problem (3) is a specific case of (9) with $T = 1$ and $p = 1$, and therefore problem (9) is harder. However, given the set of gain matrices $\{\mathbf{K}_t^t\}_{t=1}^T$ the problem has identical structure (though not dimension) to (3) and gives a more accurate bound on the performance of the filter.

Similarly to (5) in the one-step update solution, we are actually interested in finding $\{\mathbf{K}_t\}_{t=1}^T$ which minimize (7). However, in the full recollection case, presented in (7), as well as the simpler version in (9), the objective function coefficients are not separable in $\{\mathbf{K}_t\}_{t=1}^T$. Therefore, the question arises: how can we find a sequence $\{\mathbf{K}_t\}_{t=1}^T$ which minimizes the error over all $t = 1,\ldots,T$ simultaneously. Due to the high non convexity of the problem in variables $\{\mathbf{K}_t\}_{t=1}^T$, the intuitive solution would be to produce a greedy algorithm, which at each point t finds the best $\mathbf{K}_t$ and uses it as a parameter for the next time step. This greedy solution, however, does not necessarily ensure an optimal worst case performance, since the solution for each time step does not take into account the adverse effect it might have on the following time steps. The only guarantee is for the first time step, which is not effected by prior decisions. Thus, given $\{\mathbf{K}_t\}_{t=1}^{T-1}$, equation (8) ensures that the matrix $\mathbf{E} = [\mathbf{c}_T, \mathbf{A}_T^{\boldsymbol{\delta}_0}, \mathbf{A}_T^{\mathbf{a}_1}, \ldots, \mathbf{A}_T^{\mathbf{a}_T}, \mathbf{A}_T^{\mathbf{v}_1}, \ldots, \mathbf{A}_T^{\mathbf{v}_T}]$ is actually an affine transformation of $\mathbf{K}_T$.

Given $\{\mathbf{K}_t\}_{t=1}^T$, for some fixed T, problem (7) is a maximization of a convex non-homogeneous quadratic function over convex quadratic constraints which define a compact set. This problem is a *Block Quadratic Constrained Quadratic Problem*(Block-QCQP), and has a normalized reformulation, as shown in [28]. In this reformulation, we first add a scalar variable s such that $s^2 \leq 1$, and change the objective function to:

$$\left\| \mathbf{c}_T s + \mathbf{A}_T^{\boldsymbol{\delta}_0}\boldsymbol{\delta}_0 + \sum_{t=1}^{T}(\mathbf{A}_T^{\mathbf{a}_t}\mathbf{a}_t + \mathbf{A}_T^{\mathbf{v}_t}\mathbf{v}_t) \right\|,$$

which allows us to achieve a homogeneous objective function, without changing the optimal solution (since there will always exist an optimal solution with $s^2 = 1$ as proven in [28]). We then construct the normalized noise vector $\boldsymbol{\chi} = [\boldsymbol{\chi}_1; \ldots; \boldsymbol{\chi}_m] \in \mathbb{R}^n$, where $m = 2T + p + 1$ and $n = 1 + p + \tilde{r}T + qT$, and the objective function coefficient matrices $\{\tilde{\mathbf{A}}_i\}_{i=1}^m$ as follows,

$$\boldsymbol{\chi}_i = \begin{cases} s, & i = 1 \\ \boldsymbol{\delta}_0^j, & i = j+1,\ j \in [p] \\ \mathbf{Q}_a^{-1/2}\mathbf{a}_t, & i = j+p+1,\ t \in [T] \\ \mathbf{R}^{-1/2}\mathbf{v}_t, & i = t+T+p+1,\ t \in [T], \end{cases} \qquad \tilde{\mathbf{A}}_i = \begin{cases} \mathbf{c}_T, & i = 1 \\ \mathbf{A}_T^{\boldsymbol{\delta}_0^p}, & i = j+1,\ j \in [p] \\ \mathbf{A}_T^{\mathbf{a}_t}\mathbf{Q}_a^{1/2}, & i = j+p+1,\ t \in [T] \\ \mathbf{A}_T^{\mathbf{v}_t}\mathbf{R}^{1/2} & i = t+T+p+1,\ t \in [T], \end{cases} \tag{10}$$

where $\mathbf{A}_T^{\boldsymbol{\delta}_0^p}$ is the matrix whose columns are the columns of $\mathbf{A}_T^{\boldsymbol{\delta}_0}$ which correspond with partition element $\boldsymbol{\delta}_0^p$. Therefore, the domain of $\boldsymbol{\chi}$ is given by $\tilde{\Omega}$ where

$$\tilde{\Omega} = \left\{ \boldsymbol{\chi} = [\boldsymbol{\chi}_1; \ldots; \boldsymbol{\chi}_m] \in \mathbb{R}^n \ :\ \boldsymbol{\chi}_i \in \mathbb{R}^{n_i},\ \|\boldsymbol{\chi}_i\|^2 \leq 1,\ \ i = 1,\ldots,m \right\}. \tag{11}$$

Moreover, for each $i \in [m]$, $\tilde{\mathbf{A}}_i$ can be presented as $\tilde{\mathbf{A}}_i = \mathbf{K}\mathbf{C}_i + \mathbf{D}_i$, for some given matrices $\mathbf{C}_i$ and $\mathbf{D}_i$. Moreover, $\boldsymbol{\delta}_T = \tilde{\mathbf{E}}\boldsymbol{\chi}$ such that $\tilde{\mathbf{E}} = \mathbf{K}_T\mathbf{C} + \mathbf{D}$, $\tilde{\mathbf{E}} = [\tilde{\mathbf{A}}_1, \ldots, \tilde{\mathbf{A}}_m]$, $\mathbf{C} = [\mathbf{C}_1, \ldots, \mathbf{C}_m]$ and $\mathbf{D}_T = [\mathbf{D}_1, \ldots, \mathbf{D}_m]$. The normalized representation of problem (7) is given by the following formulation

$$\psi_T(\mathbf{K}_T) = \max_{\boldsymbol{\chi} \in \tilde{\Omega}} \|(\mathbf{K}_T\mathbf{C} + \mathbf{D})\boldsymbol{\chi}\| \tag{WCE}$$

The solution of problem (WCE), where the gains $\{\mathbf{K}_t\}_{t=1}^T$ are known and therefore $\tilde{\mathbf{E}}$ is a fixed matrix, is discussed extensively in [28], where the authors show that the problem is NP-hard. They later show how to bound this problem using an semidefinite relaxation (SDR), which provides a $\sqrt{2/\pi}$ approximation. they also introduce various methods of obtaining a lower bound, and discuss several cases in which there are exact solutions. Finally it is proved in [28] that this approximation can be computed efficiently even for noise vectors with dimension reaching a few thousands.

An optimal greedy solution can then be obtained by solving the *optimal robust linear filter* problem:

$$\min_{\mathbf{K}_T} \psi_T(\mathbf{K}_T) \tag{P}$$

Problem (P) is an optimal SVE problem, where the estimation set is an ℓ_2 ball, and the noise has a full-recollection bound. At each point in time we find the filter which minimizes the size (radius) of this ball. We will later show how this ball can be substituted by a more general ellipsoid.

In Sect. 4 we demonstrate how the same SDR, which was used to approximate problem (WCE), can also be utilized to efficiently obtain a sequence of gain matrices $\{\mathbf{K}_t\}_{t=1}^T$ which will provide an approximation of (P). We discuss different types of filters, which utilize either the entire noise and observation history, or part of it.

4 Robust Filtering

In this section we focus on solving problem (P). We assume that the recursive scheme discussed in the previous section is utilized to obtain the sequence $\{\mathbf{K}_t\}_{t=1}^T$. Therefore, given a fixed T and known gain matrices $\{\mathbf{K}_t\}_{t=1}^{T-1}$, we denote the worst case solution of (WCE) for a certain filter gain $\mathbf{K} \equiv \mathbf{K}_T$ as $\psi(\mathbf{K}) \equiv \psi_T(\mathbf{K})$. Using equation (WCE), problem (P) can be represented as the following non convex problem in $\mathbf{K}$.

$$\min_{\mathbf{K}\in\mathbb{R}^{r\times(qT+1)}} \psi(\mathbf{K}) = \min_{\mathbf{K}\in\mathbb{R}^{r\times(qT+1)}} \max_{\boldsymbol{\chi}\in\tilde{\Omega}} \left\| \sum_{i=1}^{m} (\mathbf{K}\mathbf{C}_i + \mathbf{D}_i)\boldsymbol{\chi}_i \right\| = \min_{\mathbf{K}\in\mathbb{R}^{r\times(qT+1)}} \max_{\boldsymbol{\chi}\in\tilde{\Omega}} \|(\mathbf{K}\mathbf{C}+\mathbf{D})\boldsymbol{\chi}\|.$$

Since finding the worst case solution, when the filter gain matrices are given (the inner maximization), is an NP-hard problem, it follows that finding the optimal filter gain matrices for this measure is also NP-hard. Therefore, we will aim to find a filter which has worst case performance close to that of the optimal filter, using the following definition.

Definition 1 *Let ψ to be a function of matrix gain $\mathbf{K}$, and $\mathcal{K}$ be a convex set. Let $\psi^* = \min_{\mathbf{K}\in\mathcal{K}} \psi(\mathbf{K})$. a gain matrix $\mathbf{K}^\diamond \in \mathcal{K}$ is a ρ-*approximation filter of ψ^* *if $\frac{\psi^*}{\psi(\mathbf{K}^\diamond)} \geq \rho$ where $0 << \rho < 1$.*

Rewriting $\|\boldsymbol{\chi}_i\|^2 = \boldsymbol{\chi}'\boldsymbol{\mathcal{I}}_i\boldsymbol{\chi}$, squaring the objective function, the SDR for problem (WCE) is given by the following.

$$\psi_{SDP}(\mathbf{K})^2 = \max_{\boldsymbol{\mathcal{X}}} \{\mathrm{Tr}\left((\mathbf{K}\mathbf{C}+\mathbf{D})'(\mathbf{K}\mathbf{C}+\mathbf{D})\boldsymbol{\mathcal{X}}\right) \;:\; \boldsymbol{\mathcal{X}} \succeq 0,\;\; \mathrm{Tr}\left(\boldsymbol{\mathcal{I}}_i\boldsymbol{\mathcal{X}}\right) \leq 1,\;\; i = 1,\dots m\}. \tag{12}$$

From the results in [28,21] it follows that the value of the SDR for problem (WCE) for any given $\mathbf{K}$ (the inner maximization in problem (12)), satisfies:

$$\psi(\mathbf{K}) \leq \psi_{SDP}(\mathbf{K}) \leq \sqrt{\frac{\pi}{2}}\psi(\mathbf{K}). \tag{13}$$

So an upper bound on the value of problem (P) is given by solving the following relaxation.

$$\min_{\mathbf{K}} \psi_{SDP}(\mathbf{K})^2 \tag{SDR-P}$$

Furthermore, problem (12) is regular and convex for a given $\mathbf{K}$, indeed $\boldsymbol{\mathcal{X}} \equiv 0$ is in the interior of the feasible set, and the problem is bounded. Therefore strong duality holds. Consequently, replacing the inner maximization problem (12) by its dual, will results in the same objective function value. Using this dual representation, problem (SDR-P) is as follows.

$$\begin{aligned}&\min_{\mathbf{K}\in\mathbb{R}^{r\times Tq}}\min_{\boldsymbol{\mu}\in\mathbb{R}^m_+}\sum_{i=1}^m \mu_i\\ &s.t.\quad \sum_{i=1}^m \mu_i\boldsymbol{\mathcal{I}}_i-(\mathbf{KC}+\mathbf{D})'(\mathbf{KC}+\mathbf{D})\succeq 0\end{aligned}$$

We can convert the single matrix inequality to an affine inequality in $\mathbf{K}$. By utilizing the Schur complement lemma this problem reduces to:

$$\begin{aligned}&\min_{\mathbf{K}\in\mathbb{R}^{r\times Tq},\boldsymbol{\mu}\in\mathbb{R}^m_+}\sum_{i=1}^m \mu_i\\ &s.t.\\ &\begin{bmatrix}\sum\limits_{i=1}^m \mu_i\boldsymbol{\mathcal{I}}_i & (\mathbf{KC}+\mathbf{D})'\\ (\mathbf{KC}+\mathbf{D}) & \mathbf{I}\end{bmatrix}\succeq 0\end{aligned}\tag{DSDR-P}$$

The resulting problem is convex in both $\boldsymbol{\mu}$ and $\mathbf{K}$ and is consequently solvable. Moreover any optimal solution $\mathbf{K}^\diamond$ to problem (DSDR-P) is also optimal for problem (SDR-P).

The saddle point problem (SDR-P) can be solved directly, by applying algorithms which solve the equivalent variational inequality, such as the ones presented in [20] and [22]. Alternatively fast first order methods can be used to solve problem (DSDR-P) [4]. In both cases projection on the semidefinite cone must be computed, at the computational cost of $O(n^3)$ per iteration, and the number of iterations needed to achieve an accuracy of ϵ is of order $O(1/\sqrt{\epsilon})$. There are also methods which do not involve such a projection step and therefore have lower computational cost per iteration. An example for such algorithm is the sampling algorithm, shown in [11], which requires $O(\log(m)/\epsilon^2)$ iterations, each taking $O(n^2/\epsilon^{1.5})$ operations to achieve accuracy ϵ and ϵ-feasibility. Such algorithms may result in an infeasible solution, and are inefficient for small ϵ values. The alternative to these methods is to use the interior point algorithm which converges rapidly but might have high computational cost per iteration. The choice of which method to use is then dependent on the trade-off between the problem size and the required precision.

We chose to implement the interior point methods for SDP described in [13], which is able to solve problem (12) efficiently (See [28]). In this case, however, it is not trivial to show that each iteration requires the same computational complexity as that required to solve the same problem for fixed $\mathbf{K}$, that is $O(n^3)$. Recalling that $\mathbf{K}\in\mathbb{R}^{r\times(qT+1)}$, and $r << n$ is the constant size of the estimated state, we can arrive at the following theorem:

Theorem 1 *Let (DSDR-P) be the SDP relaxation of problem (P), with variables $\mathbf{K}\in\mathbb{R}^{r\times\tilde{q}}$ and $\boldsymbol{\mu}\in\mathbb{R}^m$ where $\tilde{q}\leq qT+1$ and $m=2T+p+1$ for constant q, p and r. Then problem (DSDR-P) can be solved with accuracy ϵ by an interior point algorithm, within $O(\sqrt{n}\log(\frac{1}{\epsilon}))$ iterations each requiring $O(n^3)$ operations.*

The proof of this theorem and full computational considerations as well as implementation are given in appendix A. The following result shows that the filter obtained by solving problem (DSDR-P) is actually a $\sqrt{2/\pi}$-approximation filter.

Proposition 1 *Let $\mathbf{K}^\diamond$ be an optimal solution of problem (DSDR-P), then gain $\mathbf{K}^\diamond$ defines a ρ-approximation filter of ψ^*, with $\rho=\sqrt{\frac{2}{\pi}}$.*

Proof We use the notation $\psi^{SDP}(\mathbf{K})$ given above, for the square root of the SDP relaxation value for problem (WCE) given $\mathbf{K}$. Given $\mathbf{K}^*$ is an optimal solution of problem (P), with value ψ^*, the following relations hold:

$$\psi(\mathbf{K}^*) \leq \psi(\mathbf{K}^\diamond) \leq \psi^{SDP}(\mathbf{K}^\diamond) \leq \psi^{SDP}(\mathbf{K}^*)$$

The left and right inequalities are the result of $\mathbf{K}^*$ being an optimal solution to problem (4) and $\mathbf{K}^\diamond$ an optimal solution of problem (DSDR-P), the middle inequality is a result of the SDP being a relaxation to (9) and therefore achieving a higher value. From the SDP relaxation properties given in (13), we have that $\sqrt{\frac{\pi}{2}}\psi(\mathbf{K}^*) \geq \psi^{SDP}(\mathbf{K}^*)$, which leads directly to

$$\frac{\psi^*}{\psi(\mathbf{K}^\diamond)} = \frac{\psi(\mathbf{K}^*)}{\psi(\mathbf{K}^\diamond)} \geq \sqrt{\frac{2}{\pi}}. \qquad \square$$

We will refer to the optimal filter derived from problem (DSDR-P), and applied recursively, as the Greedy Affinely Adjustable Robust filter (GAARF).

The following proposition shows that the added variable s, which we used to make the dependence on $\mathbf{K}_T$ linear rather than affine, is redundant. This is due to the fact that the constant term $\mathbf{c}_T$ can always be regarded as zero, which implies that there always exists an optimal filter such that $\mathbf{k}_t = 0$ for any t.

Proposition 2 *Given problem* (P) *where* $\psi(\mathbf{K})$ *is defined with respect to some norm (not necessarily Euclidean), there exists an optimal solution* $\mathbf{K}^* = [\mathbf{k}, \mathbf{K}^1, \ldots, \mathbf{K}^T]$ *such that* $\mathbf{c}_T = 0$.

Proof We will assume to the contrary, that any $\mathbf{K}^*$ for which $c_T = 0$ is not optimal, and therefore there exists $\mathbf{K}^\diamond$ such that $\mathbf{c}^\dagger \neq 0$ such that $\psi(\mathbf{K}^\diamond) < \psi(\mathbf{K}^*)$. We take $\mathbf{K}^\diamond$ to be the optimal solution and construct $\mathbf{K}^*$ which is equal to $\mathbf{K}^\diamond$ in all elements except $\mathbf{k}$. Notice that since $\mathbf{c}_T = \mathbf{k} + \mathbf{d}_T$ where $\mathbf{d}_T$ is some constant vector, and since $\mathbf{A}_T$ defined above is independent of $\mathbf{k}$, then by choosing $\mathbf{k} = -\mathbf{d}_T$ (so $\mathbf{c}_T = 0$). The value of $\mathbf{A}_T$ does not change. Let $\mathbf{c}_T^\dagger \equiv \mathbf{k}^\dagger + \mathbf{d}_T \neq 0$ and since $\mathbf{K}^*$ and $\mathbf{K}^\diamond$ only differ in the value of k then $\mathbf{A}_T \equiv \mathbf{A}_T(K^*) = \mathbf{A}_T(K^\dagger)$. We define $\boldsymbol{\omega} \equiv \boldsymbol{\omega}_T$, the uncertain vector for time T, and $\boldsymbol{\omega}^*$, $\boldsymbol{\omega}^\diamond$ to be the worst case solution for $\mathbf{K}^*$ and $\mathbf{K}^\diamond$ respectively. From symmetry of the uncertainty set both $\boldsymbol{\omega}^*$ and $-\boldsymbol{\omega}^*$ are feasible noise vectors and therefore the following inequalities hold.

$$\begin{aligned}\psi(\mathbf{K}^*) = \|\mathbf{A}_T\boldsymbol{\omega}^*\| &= \frac{1}{2}\|\mathbf{c}_T^\diamond + \mathbf{A}_T\boldsymbol{\omega}^* - (\mathbf{c}_T^\diamond - \mathbf{A}_T\boldsymbol{\omega}^*)\| \\ &\leq \frac{1}{2}\left(\|\mathbf{c}_T^\diamond + \mathbf{A}_T\boldsymbol{\omega}^*\| + \|\mathbf{c}_T^\diamond - \mathbf{A}_T\boldsymbol{\omega}^*\|\right) \leq \|\mathbf{c}_T^\diamond + \mathbf{A}_T\boldsymbol{\omega}^\diamond\| = \psi(\mathbf{K}^\diamond)\end{aligned}$$

where the last inequality is due to $\boldsymbol{\omega}^\diamond$ being the worst case solution for $\mathbf{K}^\diamond$. Thus, $\psi(\mathbf{K}^*) \leq \psi(\mathbf{K}^\diamond)$ in contradiction with the assumption $\psi(\mathbf{K}^\diamond) < \psi(\mathbf{K}^*)$. □

Corollary 1 *There exists an optimal solution to problem* (P) *such that* $\mathbf{k}_t = 0$ *for all* $t = 1, 2, \ldots$

The corollary is a straight forward result of proposition 2, applying it recursively from $t = 1$ to obtain $\mathbf{c}_t = \mathbf{d}_t + \mathbf{k}_t$ where $\mathbf{d}_t$ is zero as a linear function of $\{\mathbf{k}_\tau\}_{\tau=1}^{t-1}$ (see equation (8)).

Finally, we revisit the filter based on limited history, such as the one suggested in (3). This is a good point of reference to show that a limited history filter might not be optimal. The problem of finding the filter whose gain minimizes (9) is as follows.

$$\min_{\mathbf{K}_T \in \mathcal{K}_T} \psi_T(\mathbf{K}_T), \tag{14}$$

where $\mathcal{K}_T = \{\mathbf{K}_T : \mathbf{k}_T = 0,\ \mathbf{K}_T^\tau = 0\ \forall \tau < T\}$. In general, based on Proposition 1 and Definition 1, we can conclude the following.

Corollary 2 *Let $\mathcal{K}$ be a closed convex set, and define the extended value function $\tilde{\psi}$*

$$\tilde{\psi}(\mathbf{K}) = \begin{cases} \psi(\mathbf{K}) & \mathbf{K} \in \mathcal{K} \\ \infty & \textit{otherwise} \end{cases}$$

and let $\mathbf{K}^{\diamond}$ be an optimal solution of problem (DSDR-P) with added constraint $\mathbf{K} \in \mathcal{K}$, then gain $\mathbf{K}^{\diamond}$ defines a ρ-approximation filter of $\tilde{\psi}^* = \min_{K \in \mathcal{K}} \psi(\mathbf{K})$, with $\rho = \sqrt{\frac{2}{\pi}}$.

Corollary 2 and Theorem 1 imply that the filter based on the SDR of problem (14) can be computed efficiently and is a $\sqrt{2/\pi}$-approximation filter of the optimal value of problem (14). We call the filter obtained by this SDR and applied recursively the *Greedy Robust Filter* (GRF).

4.1 Optimizing Expected Worst Case Square Error

In the previous section we discussed the case where the noise elements are all unknown but bounded. However, in tracking applications, while the system noise is usually bounded the measurement noise is often probabilistic. This situation occurs when the sensor has a known statistical error, while the object acts in an adversarial way, and the only knowledge of its movement is given by its physical limitations. In this case the measurement error is independent of both the dynamic system noise and the system's state. Given this motivation, we dedicate this section to discussing the way to obtain a filter which deals well with an uncertain noise vector ω which is partly probabilistic and partly bounded. We start by generalizing the assumptions on ω given in Sect. **??**.

Assume $\boldsymbol{\omega}$ is partitioned into two sets: $\boldsymbol{\omega}^1 \in \mathbb{R}^{\bar{n}_1}$ which is bounded and $\boldsymbol{\omega}^2 \in \mathbb{R}^{\bar{n}_2}$ which is probabilistic. We further assume that $\boldsymbol{\omega}^1$ is partitioned to m partition elements such that element $\boldsymbol{\omega}_i^1$ is bounded in an ellipsoid ${\boldsymbol{\omega}_i^1}'{\mathbf{B}_i^1}^{-1}\boldsymbol{\omega}_i^1 \leq b_i$. The random noise $\boldsymbol{\omega}^2$ has an unknown probability distribution which satisfies $\mathbb{E}(\boldsymbol{\omega}^2) = 0$ and $\mathrm{Cov}(\boldsymbol{\omega}^2) = \mathbf{B}^2$, where $\mathbf{B}^2$ is a PD matrix, and is independent of $\boldsymbol{\omega}^1$ and of the state. We will look at the normalized form achieved by the linear transformations $\boldsymbol{\chi}_i^1 = {\mathbf{B}_i^1}^{-1/2}\boldsymbol{\omega}_i^1/b_i$ and $\boldsymbol{\chi}^2 = {\mathbf{B}^2}^{-1/2}\boldsymbol{\omega}^2$. The set bounding the normalized vector $\boldsymbol{\chi}^1$, and the set of all possible probability distributions functions of the normalized random variable $\boldsymbol{\chi}^2$ are denoted $\tilde{\Omega}^1$ and $\mathcal{P}^2$ respectively, and defined below.

$$\begin{aligned} \tilde{\Omega}^1 &= \left\{\boldsymbol{\chi}^1 : {\boldsymbol{\chi}^1}'\boldsymbol{\mathcal{I}}_i\boldsymbol{\chi}^1 = \left\|\boldsymbol{\chi}_i^1\right\|^2 \leq 1, \quad i = 1, \ldots, m\right\}, \\ \mathcal{P}^2 &= \left\{P : \text{ if } \boldsymbol{\chi}^2 \sim P \text{ then } \mathbb{E}(\boldsymbol{\chi}^2) = 0,\ \mathrm{Cov}(\boldsymbol{\chi}^2) = \mathbf{I}\right\}. \end{aligned} \tag{15}$$

Moreover, assuming $\boldsymbol{\chi}^2 \sim P$, the expected value of some function g of $\boldsymbol{\chi}^2$ is denoted by $\mathbb{E}_p(g(\boldsymbol{\chi}^2))$.

Ideally, our goal would be to find a filter whose gain $\mathbf{K}$ minimizes the *mean worst case square estimation error* measure, which combines the measures used in the Robust and Bayesian frameworks. Given $\mathbf{K}$ we construct this measure in two stages. In the first stage, we represent the worst case estimation error as a function of $\boldsymbol{\chi}^2$, and obtain the following random variable,

$$\varphi(\mathbf{K}, \boldsymbol{\chi}^2) = \max_{\boldsymbol{\chi}^1 \in \tilde{\Omega}^1} \left\| \sum_i \mathbf{A}_i^1 \boldsymbol{\chi}_i^1 + \mathbf{A}^2 \boldsymbol{\chi}^2 \right\|^2, \tag{16}$$

where, as before, $\mathbf{A}^1 \equiv \mathbf{K}\mathbf{C}^1 + \mathbf{D}^1$ and $\mathbf{A}^2 \equiv \mathbf{K}\mathbf{C}^2 + \mathbf{D}^2$ are affine functions of $\mathbf{K}$. In the second stage, the expected value of this random variable is computed

$$\overline{\varphi}_P(\mathbf{K}) = \mathbb{E}_P(\varphi(\mathbf{K}, \boldsymbol{\chi}^2)).$$

Notice that the value of $\overline{\varphi}_P(\mathbf{K})$ is a function of the specific unknown distribution P. The optimal filter gain for each $P \in \mathcal{P}^2$ is then obtained by the following minimization.

$$\overline{\varphi}_P^* = \min_{\mathbf{K}} \overline{\varphi}_P(\mathbf{K}). \tag{EP}$$

Since each P produces a different filter, and P is unknown it is unclear from problem (EP) which filter should be used. Even assuming P is known, computing $\varphi(\mathbf{K}, \boldsymbol{\chi}^2)$ is NP-hard and does not have a closed form (as shown in Sect. 3), leading to the problems of finding the expected value $\overline{\varphi}_P(\mathbf{K})$ and optimal $\mathbf{K}$ being intractable. Therefore, we will try to obtain a bound on $\overline{\varphi}_P(\mathbf{K})$ which has a closed form and is independent of P for any $P \in \mathcal{P}^2$. Optimizing $\mathbf{K}$ over this bound, will produce a filter with guaranteed performance for any $P \in \mathcal{P}^2$.

We first define the value of the SDR of (16) for some given $\mathbf{K}$ and $\boldsymbol{\chi}^2 = 0$ by $(\psi^{SDP}(\mathbf{K}))^2$ and

$$\begin{aligned} \gamma(\mathbf{K}) &= \mathrm{Tr}(\mathbf{A}^2\mathbf{A}^{2\prime}) = \mathbb{E}_P(\left\|\mathbf{A}^2\boldsymbol{\chi}^2\right\|^2),\ \forall P \in \tilde{\Omega}_P^2 \\ \varphi(\mathbf{K}, 0) &= \max_{\boldsymbol{\chi}^1 \in \tilde{\Omega}^1} \left\|\mathbf{A}^1\boldsymbol{\chi}^1\right\|^2. \end{aligned} \tag{17}$$

We proceed to introduce the following auxiliary problem

$$\min_{\mathbf{K}} \{\varphi(\mathbf{K}, 0) + \gamma(\mathbf{K})\},$$

as well as its α-SDP relaxation,

$$\varphi^*_{SDP_\alpha} = \min_{\mathbf{K}} \left\{(1+\alpha)(\psi^{SDP}(\mathbf{K}))^2 + (1+\frac{1}{\alpha})\gamma(\mathbf{K})\right\}. \tag{α-SDP-EP}$$

Next, we show that problem (α-SDP-EP) is tractable and that it is in fact an approximation of problem (EP) for any $P \in \mathcal{P}^2$.

Proposition 3 *Let $\mathbf{K}^\diamond$ be an optimal solution to problem (α-SDP-EP) with $\alpha = 2/\pi$. Then $\mathbf{K}^\diamond$ is a β-approximation filter of $\overline{\varphi}_P^*$ for any $P \in \mathcal{P}^2$, i.e.*

$$\frac{\overline{\varphi}_P^*}{\overline{\varphi}_P(\mathbf{K}^\diamond)} \geq \beta.$$

Proof Let $\tilde{\boldsymbol{\chi}}^1 \in \arg\max_{\chi^1 \in \tilde{\Omega}^1} \varphi(\mathbf{K}, 0)$ for a given matrix $\mathbf{K}$ then by triangular inequality and the optimality of $\tilde{\boldsymbol{\chi}}^1$ that:

$$\left\|\sum_i \mathbf{A}_i^1\boldsymbol{\chi}_i^1 + \mathbf{A}^2\boldsymbol{\chi}^2\right\| \leq \left\|\sum \mathbf{A}_i^1\boldsymbol{\chi}_i^1\right\| + \left\|\mathbf{A}^2\boldsymbol{\chi}^2\right\| \leq \left\|\sum \mathbf{A}_i^1\tilde{\boldsymbol{\chi}}_i^1\right\| + \left\|\mathbf{A}^2\boldsymbol{\chi}^2\right\| \quad \forall \boldsymbol{\chi}^1 \in \tilde{\Omega}^1$$

and therefore:

$$\varphi(\mathbf{K}, \boldsymbol{\chi}^2) \leq \left(\left\|\sum_i \mathbf{A}_i^1\tilde{\boldsymbol{\chi}}_i^1\right\| + \left\|\mathbf{A}^2\boldsymbol{\chi}^2\right\|\right)^2 = \left\|\mathbf{A}^1\tilde{\boldsymbol{\chi}}^1\right\|^2 + 2\left\|\mathbf{A}^1\tilde{\boldsymbol{\chi}}^1\right\|\left\|\mathbf{A}^2\boldsymbol{\chi}^2\right\| + \left\|\mathbf{A}^2\boldsymbol{\chi}^2\right\|^2$$

On the other hand we can bound the function from below:

$$\begin{aligned} \varphi(\mathbf{K}, \boldsymbol{\chi}^2) = \max_{\boldsymbol{\chi}^1 \in \tilde{\Omega}^1} \left\|\sum_i \mathbf{A}_i^1\boldsymbol{\chi}_i^1 + \mathbf{A}^2\boldsymbol{\chi}^2\right\|^2 &= \max_{\boldsymbol{\chi}^1 \in \tilde{\Omega}^1} \{\left\|\mathbf{A}^1\boldsymbol{\chi}^1\right\|^2 + 2(\mathbf{A}^1\boldsymbol{\chi}^1)'\mathbf{A}^2\boldsymbol{\chi}^2 + \left\|\mathbf{A}^2\boldsymbol{\chi}^2\right\|^2\} \\ &\geq \left\|\mathbf{A}^1\tilde{\boldsymbol{\chi}}^1\right\|^2 + 2(\mathbf{A}^1\tilde{\boldsymbol{\chi}}^1)'\mathbf{A}^2\boldsymbol{\chi}^2 + \left\|\mathbf{A}^2\boldsymbol{\chi}^2\right\|^2. \end{aligned}$$

Moreover, for any $P \in \mathcal{P}^2$, the following holds true

$$\mathbb{E}_P((\mathbf{A}^1\tilde{\boldsymbol{\chi}}^1)'\mathbf{A}^2\boldsymbol{\chi}^2) = (\mathbf{A}^1\tilde{\boldsymbol{\chi}}^1)'\mathbf{A}^2\mathbb{E}_P(\boldsymbol{\chi}^2) = 0$$

$$\mathbb{E}_P\left(\|\mathbf{A}^1\tilde{\boldsymbol{\chi}}^1\|\,\|\mathbf{A}^2\boldsymbol{\chi}^2\|\right) = \|\mathbf{A}^1\tilde{\boldsymbol{\chi}}^1\|\,\mathbb{E}_P\left(\left\|\boldsymbol{\chi}^{2\prime}\mathbf{A}^2\right\|\right) \leq \sqrt{\varphi(\mathbf{K},0)\mathbb{E}_P\left(\left\|\boldsymbol{\chi}^{2\prime}\mathbf{A}^2\right\|^2\right)} = \sqrt{\gamma(\mathbf{K})\varphi(\mathbf{K},0)},$$

where the inequality in the second line is due to the concavity of the square root function. Combining this result with the definition of $\gamma(\mathbf{K})$ and $\varphi(\mathbf{K},0)$ we obtain the following.

$$\begin{aligned}\varphi(\mathbf{K},0)+\gamma(\mathbf{K}) \leq \mathbb{E}_P(\varphi(\mathbf{K},\boldsymbol{\chi}^2)) = \overline{\varphi}_P(\mathbf{K}) &\leq \varphi(\mathbf{K},0)+\gamma(\mathbf{K})+2\sqrt{\varphi(\mathbf{K},0)\gamma(\mathbf{K})}\\ &\leq ((1+\alpha)\varphi(\mathbf{K},0)+(1+\frac{1}{\alpha}\gamma(\mathbf{K})) \quad \forall\alpha>0,\end{aligned} \tag{18}$$

where the last inequality is derives from the fact that $\sqrt{\varphi(\mathbf{K},0)\gamma(\mathbf{K})} = \sqrt{\alpha\varphi(\mathbf{K},0)\frac{1}{\alpha}\gamma(\mathbf{K})}$ and the inequality of arithmetic and geometric means.

Given P, let $\mathbf{K}^*_P$ denote an optimal solution for problem (EP). Combining the inequalities in (18) with the fact that $\varphi(\mathbf{K},0) \geq \frac{2}{\pi}(\psi^{SDP}(\mathbf{K}))^2$ for any $\mathbf{K}$, as a result of the SDR approximation (see equation (13)), we obtain the following inequalities,

$$\overline{\varphi}^*_P = \overline{\varphi}_P(\mathbf{K}^*_P) \geq \varphi(\mathbf{K}^*_P,0)+\gamma(\mathbf{K}^*_P) \geq \min_{\mathbf{K}}\{\varphi(\mathbf{K},0)+\gamma(\mathbf{K})\} \geq \min_{\mathbf{K}}\left\{\frac{2}{\pi}(\psi^{SDP}(\mathbf{K}))^2+\gamma(\mathbf{K})\right\}$$

$$\begin{aligned}\overline{\varphi}^*_P = \min_{\mathbf{K}}\overline{\varphi}_P(\mathbf{K}) &\leq \min_{\mathbf{K}}\left\{(1+\alpha)\varphi(\mathbf{K},0)+(1+\frac{1}{\alpha})\gamma(\mathbf{K})\right\}\\ &\leq \min_{\mathbf{K}}\left\{(1+\alpha)(\psi^{SDP}(\mathbf{K}))^2+(1+\frac{1}{\alpha})\gamma(\mathbf{K})\right\} \triangleq \varphi^*_{SDP_\alpha}.\end{aligned}$$

Choosing $\alpha = 2/\pi$, it follows that for $\beta = (1+\pi/2)^{-1}$ results in

$$\varphi^*_{SDP_{\frac{2}{\pi}}}\beta \leq \overline{\varphi}^*_P \leq \varphi^*_{SDP_{\frac{2}{\pi}}}.$$

Finally, using the optimality of $\mathbf{K}^\diamond$ it follows that $\mathbf{K}^\diamond$ defines a β-approximation filter of $\overline{\varphi}^*_P$:

$$\overline{\varphi}^*_P \leq \overline{\varphi}_P(\mathbf{K}^\diamond) \leq \varphi^*_{SDP_{\frac{2}{\pi}}} \leq \frac{1}{\beta}\overline{\varphi}^*_P, \quad \forall P \in \mathcal{P}^2.$$ □

This proof implies that if the optimal solution $\mathbf{K}^\diamond$ satisfies $\psi^{SDP}(\mathbf{K}^\diamond) \leq \gamma\varphi(\mathbf{K}^\diamond,0)$, for some known $1 \leq \gamma < \pi/2$, then $\varphi^*_{SDP_{\frac{1}{\gamma}}}$ is a $(1+\gamma)$ approximation of $\overline{\varphi}^*_P$. This means that if the SDR is accurate (meaning $\gamma = 1$) then $\mathbf{K}^\diamond$ provides a 2-approximation filter.

We will now show that problem (α-SDP-EP) is equivalent to a convex optimization problem, more specifically an SDP, and therefore tractable. We will then proceed to prove that this SDP can be solved via an interior point method, similarly to problem (DSDR-P), without additional computational expense.

Proposition 4 *Solving problem* (α-SDP-EP) *is equivalent to solving the following SDP:*

$$\begin{aligned}\varphi^*_{SDP_\alpha} = &\min_{\mathbf{K},\boldsymbol{\mu},\boldsymbol{\mathcal{U}}}(1+\alpha)\boldsymbol{\mu}'\mathbf{e}+(1+\frac{1}{\alpha})Tr(\boldsymbol{\mathcal{U}})\\ &s.t.\\ &\begin{bmatrix}\sum_{i=1}^{m}\mu_i\boldsymbol{\mathcal{I}}_i & (\mathbf{K}\mathbf{C}^1+\mathbf{D}^1)'\\ (\mathbf{K}\mathbf{C}^1+\mathbf{D}^1) & \mathbf{I}\end{bmatrix} \succeq 0\\ &\begin{bmatrix}\boldsymbol{\mathcal{U}} & (\mathbf{K}\mathbf{C}^2+\mathbf{D}^2)\\ (\mathbf{K}\mathbf{C}^1+\mathbf{D}^1)' & \mathbf{I}\end{bmatrix} \succeq 0\end{aligned} \tag{α-D-SDP}$$

Proof Using the fact that of $\mathbf{A}^1$ and $\mathbf{A}^2$ are affine functions of $\mathbf{K}$, the definition in (17) as well as using the SDR dual, given in (DSDR-P), we can reformulate problem (α-SDP-EP) as follows:

$$\begin{aligned} &\min_{\mathbf{K},\boldsymbol{\mu}} (1+\alpha)\boldsymbol{\mu}'\mathbf{e} + (1+\frac{1}{\alpha})\gamma(\mathbf{K}) \\ &s.t. \quad \begin{bmatrix} \sum_{i=1}^{m} \mu_i \boldsymbol{\mathcal{I}}_i & (\mathbf{KC}^1+\mathbf{D}^1)' \\ (\mathbf{KC}^1+\mathbf{D}^1) & \mathbf{I} \end{bmatrix} \succeq 0. \end{aligned} \tag{19}$$

In order to eliminate $\gamma(\mathbf{K})$ from the objective function we use the following equivalence:

$$\begin{aligned} \gamma(\mathbf{K}) &= Tr\left((\mathbf{KC}^2+\mathbf{D}^2)(\mathbf{KC}^2+\mathbf{D}^2)'\right) \\ &= \min_{\boldsymbol{\mathcal{U}}\in\mathbb{R}^{r\times r}} \left\{\mathrm{Tr}(\boldsymbol{\mathcal{U}}) : \boldsymbol{\mathcal{U}} \succeq (\mathbf{KC}^2+\mathbf{D}^2)(\mathbf{KC}^2+\mathbf{D}^2)'\right\}, \end{aligned}$$

and then apply the Schur complement to the problem's constraint to obtain this alternative linear matrix inequality

$$\begin{bmatrix} \boldsymbol{\mathcal{U}} & (\mathbf{KC}^2+\mathbf{D}^2) \\ (\mathbf{KC}^2+\mathbf{D}^2)' & I \end{bmatrix} \succeq 0.$$

Substituting $\gamma(\mathbf{K})$, in formulation (19), with the equivalent optimization problem, results in problem (α-D-SDP). □

Solving problem (α-D-SDP) is a bit more complex than (DSDR-P) because of the additional constraint. Nonetheless, assuming r is fixed, the same interior point algorithm can be applied, with approximately the same computational cost and a similar convergence rate.

Theorem 2 *Let* $\mathbf{K} \in \mathbb{R}^{r\times\tilde{q}}$ *where* r *is a fixed constant and* $O(\tilde{q}) = O(n)$ *then problem* (α-SDP-EP) *can be solved with accuracy* ϵ *via an interior point algorithm with cost of* $O(n^3)$ *operations per iteration, and* $O(\sqrt{n}\log(\frac{1}{\epsilon}))$ *iterations.*

The proof and full implementation details is given in appendix B.

This approach of combining probabilistic and bounded noise, unifies the frameworks underlying both the KF and SVE. Moreover, if $\bar{n}_1 = 0$, i.e. the entire uncertainty vector is of a probabilistic nature, the optimal filter, generated by this approach, will coincide with that of the KF. Alternatively, if $\bar{n}_2 = 0$, i.e. the entire uncertainty is unknown but bounded, we revert to the formulation presented in Sect. **??**, i.e. the optimal filter will be the *GAARF*.

4.2 Ellipsoid Estimation Set and Rolling Horizon

Both the *GAARF* and the *GRF* presented in Sect. **??**, are *offline* filters, in the sense that the gain matrices $\mathbf{K}_T$ can be computed in advance for each T. However, in both cases, the dimension of the optimization problem solved to obtain these gain matrices grows linearly in time, leading to computational difficulties as the value of T increases. This is due to the fact that the entire noise history is considered explicitly when computing worst case performance. Yet not explicitly considering the entire noise history, e.g., in the one-step update approach, will result in a filter which is either too conservative and does not approximate the optimal filter well, or one which is not sufficiently robust.

In this section we present a technique to obtain a more concise representation of the noise history, which enables a constant computational cost per time-step without significant loss of information. To achieve this goal we limit the number of past purified outputs used by the filter at each step. Let S be a predetermined integer, we are going to use a restricted linear filter with gain matrix $\mathbf{K}_{T,S} = [\mathbf{K}_T^{T-S}, \ldots, \mathbf{K}_T^T]$, such that

$$\hat{x}_t = F\hat{x}_{t-1} + \sum_{\tau=\max\{t-S+1,1\}}^{t} \mathbf{K}_t^\tau \mathbf{z}_\tau. \tag{20}$$

However, this formulation still does not restrict the size of the problem, since $\mathbf{z}_t$ is dependent on the entire $\boldsymbol{\omega}_t$ vector, including noise elements from time steps smaller than $T-S+1$. Therefore, the next step is to define a recursive formulation for $\boldsymbol{\delta}_t = \hat{\mathbf{x}}_t - \mathbf{x}_t$ which relies only on limited noise history. For that purpose let $\mathbf{Z}_t^s \equiv [\mathbf{z}_s, \ldots, \mathbf{z}_t]$ and $\boldsymbol{\omega}_t^s \equiv [\boldsymbol{\delta}_s, \mathbf{a}_{s+1}, \ldots, \mathbf{a}_t, \mathbf{v}_{s+1}, \ldots, \mathbf{v}_t]$ to denote the partial purified output and noise vector respectively. Note that $\mathbf{Z}_t \equiv \mathbf{Z}_t^0$ and $\boldsymbol{\omega}_t \equiv \boldsymbol{\omega}_t^0$. Given the filter structure in (20), we can apply the recursive formula, given by

$$\begin{aligned} \boldsymbol{\delta}_t &= \mathbf{F}\boldsymbol{\delta}_{t-1} + \mathbf{F}\mathbf{G}\mathbf{a}_t + \sum_{s=t-S+1}^{t} \mathbf{K}_T^s \mathbf{z}_s \\ \mathbf{z}_t &= \mathbf{H}\mathbf{F}\boldsymbol{\delta}_{t-1} + \mathbf{H}\mathbf{G}\mathbf{a}_t + \mathbf{v}_t, \end{aligned}$$

a fixed $\tilde{S}$ number of times, for $t = T-\tilde{S}+1, \ldots, T$, to derive the following equalities,

$$\begin{aligned} \boldsymbol{\delta}_T &= \mathbf{A}_T^{\boldsymbol{\omega}} \boldsymbol{\omega}_{T-\tilde{S}}^T + \mathbf{A}_T^{\mathbf{Z}} \mathbf{Z}_{T-\tilde{S}}^{T-\tilde{S}-S+2}, \\ \mathbf{z}_T &= \hat{\mathbf{A}}_T^{\boldsymbol{\omega}} \boldsymbol{\omega}_T^{T-S} + \hat{\mathbf{A}}_T^{\mathbf{Z}} \mathbf{Z}_{T-S}^{T-\tilde{S}-S+2}. \end{aligned}$$

The recursive formulation for matrices $\mathbf{A}_T^{\boldsymbol{\omega}} = [\mathbf{A}^{\boldsymbol{\delta}_{T-\tilde{S}}}, \mathbf{A}_T^{\mathbf{a}_{T-\tilde{S}+1}}, \ldots, \mathbf{A}_T^{\mathbf{a}_T}, \mathbf{A}_T^{\mathbf{v}_{T-\tilde{S}+1}}, \ldots, \mathbf{A}_T^{\mathbf{v}_T}]$, $\mathbf{A}_T^{\mathbf{Z}} = [\mathbf{A}_T^{\mathbf{z}_{T-\tilde{S}-S+2}}, \ldots, \mathbf{A}_T^{\mathbf{z}_{T-\tilde{S}}}]$, and similarly for $\hat{\mathbf{A}}_T^{\boldsymbol{\omega}}$ and $\hat{\mathbf{A}}_T^{\mathbf{Z}}$, given $\tilde{S} = S$ is presented in appendix D.

If we could accurately calculate the set in which the vectors $\mathbf{Z}_{T-\tilde{S}}^{T-\tilde{S}-S+2}$ and $\boldsymbol{\delta}_{T-\tilde{S}}$ reside (the so called *estimation set*) given the filters $\{\mathbf{K}_{s,S}\}_{s=1}^{T-1}$ we would retain all the information of historical noise vector $\boldsymbol{\omega}_{T-\tilde{S}}$, thus approximating more accurately the worst case estimation. Unfortunately, the estimation set structure is not known and might be complicated, and so in order to maintain the problem's structure we use ellipsoid estimation sets. Therefore, instead of Ω_T (presented in (6)), we consider the following uncertainty set.

$$\Omega_T^S = \left\{ \left[\boldsymbol{\omega}_T^{T-\tilde{S}}; \mathbf{Z}_{T-\tilde{S}}^{T-S-\tilde{S}+2}\right] \,\middle|\, \begin{array}{rl} \mathbf{a}_t' \mathbf{Q}_{\mathbf{a}}^{-1} \mathbf{a}_t \le \alpha^2 & t = T-\tilde{S}+1, \ldots, T \\ \mathbf{v}_t' R^{-1} \mathbf{v}_t \le \beta^2 & t = T-\tilde{S}+1, \ldots, T \\ \mathbf{z}_t{}' \boldsymbol{\Sigma}_{\mathbf{z}_t}^{-1} \mathbf{z}_t \;\le 1 & t = T-S-\tilde{S}+2, \ldots, T-\tilde{S} \\ \boldsymbol{\delta}_{T-\tilde{S}}{}' \boldsymbol{\Sigma}_{\boldsymbol{\delta}_{T-\tilde{S}}}^{-1} \boldsymbol{\delta}_{T-\tilde{S}} \;\le 1 & \end{array} \right\}. \tag{21}$$

Notice that although the estimation sets of vectors $\mathbf{Z}_{t-\tilde{S}}^{t-\tilde{S}-S+2}$ and $\boldsymbol{\delta}_{T-\tilde{S}}$ are obviously dependent due to the fact they all depend on $\boldsymbol{\omega}_{t-\tilde{S}-S+2}$, we still calculate the estimation set of each vector separately so to eliminate the dependency of the corresponding problem on t. Accordingly, using independent ellipsoid estimation sets will incur some loss of information.

The uncertain vector $[\boldsymbol{\omega}_T^{T-S}; \mathbf{Z}_{T-S}^{T-2S+2}]$ can be normalized, similarly to (10), provided that matrices $\{\boldsymbol{\Sigma}_{\mathbf{z}_t}\}_{t=T-2S+1}^T$ and $\boldsymbol{\Sigma}_{\boldsymbol{\delta}_{T-S}}$ are PD. Choosing $\tilde{S} = S$, the normalized vector $\boldsymbol{\chi}$ resides in the uncertainty set $\tilde{\Omega}$, defined by (11), with $m = 3S$. Moreover, matrices $\mathbf{A}_T$ and $\hat{\mathbf{A}}_T$, are replaced by their normalized version $\tilde{\mathbf{E}}$ and $\tilde{\mathbf{E}}_z$ such that $\boldsymbol{\delta}_t = \tilde{\mathbf{E}}\boldsymbol{\chi}$ and $\mathbf{z}_t = \tilde{\mathbf{E}}_z\boldsymbol{\chi}$, and $\tilde{\mathbf{E}} = \mathbf{K}_{T,S}\mathbf{C} + \mathbf{D}$. Similarly to (WCE), we define $\psi_T^S(\mathbf{K}_{T,S})$ to be the worst case error of $\mathbf{K}_{T,S}$ given the uncertainty set $\tilde{\Omega}$ derived from Ω_T^S. The optimal filter for this setting is given by the solution of the following fixed size (independent of T) optimization problem

$$\min_{\mathbf{K}_{T,S}} \psi_T^S(\mathbf{K}_{T,S}), \tag{RP}$$

which can be solved using the same methods for solving problem (P), shown in Sect. ??.

In order for the uncertainty set Ω_T^S to be well defined we need a procedure for calculating the matrices $\boldsymbol{\Sigma}_{\boldsymbol{\delta}_t}$ and $\boldsymbol{\Sigma}_{\mathbf{z}_t}$ at each point in time. Since the tightness of the worst case approximation $\psi_T^S(\mathbf{K}_{T,S})$ is dependent on the tightness of Ω_T^S, these matrices should define ellipsoids which are as small as possible. One way of representing the size of an ellipsoid is by the trace of its defining matrix. Hence, we employ trace minimization in order to minimize the ellipsoid's size. We now turn our attention to describing a method of finding matrix $\boldsymbol{\Sigma}_{\boldsymbol{\delta}_t}$ (which can also be easily extended for $\boldsymbol{\Sigma}_{\mathbf{z}_t}$). We first show that a close to optimal $\boldsymbol{\Sigma}_{\boldsymbol{\delta}_t}$ can be obtained using a convex problem, and then proving that this convex problem has an analytic solution.

Proposition 5 *Let $\boldsymbol{\chi} \in \tilde{\Omega}$ be the normalized uncertainty vector, and let $\tilde{\mathbf{E}}$ a matrix such that $\boldsymbol{\delta}_T = \tilde{\mathbf{E}}\boldsymbol{\chi}$. Let $\boldsymbol{\Sigma}^*$ denote the matrix with the smallest trace such that $\boldsymbol{\delta}_T {\boldsymbol{\Sigma}^*}^{-1} \boldsymbol{\delta}_T \leq 1$ for all $\boldsymbol{\chi} \in \tilde{\Omega}$ and let $\boldsymbol{\Sigma}^\diamond$ be the solution to the following SDP optimization problem:*

$$\begin{aligned} \min_{\boldsymbol{\Sigma}\in\mathbb{R}^{r\times r},\boldsymbol{\mu}\in\mathbb{R}^m} \quad & \mathrm{Tr}(\boldsymbol{\Sigma}) \\ s.t. \quad & \begin{bmatrix} \sum_{i=1}^m \mu_i \boldsymbol{\mathcal{I}}_i & \tilde{\mathbf{E}}' \\ \tilde{\mathbf{E}} & \boldsymbol{\Sigma} \end{bmatrix} \succeq 0, \ \mathbf{e}'\boldsymbol{\mu} \leq 1. \end{aligned} \tag{SVE}$$

then $\mathrm{Tr}(\boldsymbol{\Sigma}^*) \leq \mathrm{Tr}(\boldsymbol{\Sigma}^\diamond) \leq \frac{\pi}{2}\mathrm{Tr}(\boldsymbol{\Sigma}^*)$.

Proof Matrix $\boldsymbol{\Sigma}^*$ is the solution of the following semi-infinite optimization problem.

$$\begin{aligned} &\min_{\boldsymbol{\Sigma}\succeq 0,} \mathrm{Tr}(\boldsymbol{\Sigma}) \\ &s.t. \quad \boldsymbol{\chi}'\tilde{\mathbf{E}}'\boldsymbol{\Sigma}^{-1}\tilde{\mathbf{E}}\boldsymbol{\chi} \leq 1 \quad \forall \boldsymbol{\chi} \in \tilde{\Omega}. \end{aligned} \tag{22}$$

The constraints can be rewritten as $\sigma(\boldsymbol{\Sigma}) \leq 1$ where

$$\sigma(\boldsymbol{\Sigma}) = \max_{\boldsymbol{\chi}\in\tilde{\Omega}} \boldsymbol{\chi}'\tilde{\mathbf{E}}'\boldsymbol{\Sigma}^{-1}\tilde{\mathbf{E}}\boldsymbol{\chi}.$$

Using the dual of the SDR of the problem, similarly to (DSDR-P), and applying the SDR approximation property presented in (13), we obtain:

$$\sigma(\boldsymbol{\Sigma}) \leq \min_{\mu\in\mathcal{M}(\boldsymbol{\Sigma})} \mathbf{e}'\boldsymbol{\mu} \leq \frac{\pi}{2}\sigma(\boldsymbol{\Sigma}), \quad \mathcal{M}(\boldsymbol{\Sigma}) = \{\boldsymbol{\mu} : \sum_{i=1}^m \mu_i \boldsymbol{\mathcal{I}}_i - \tilde{\mathbf{E}}'\boldsymbol{\Sigma}^{-1}\tilde{\mathbf{E}} \succeq 0\}.$$

Replacing $\sigma(\boldsymbol{\Sigma})$ by $\mathbf{e}'\boldsymbol{\mu}$, adding constrain $\mu \in \mathcal{M}(\boldsymbol{\Sigma})$, and applying the Schur complement lemma results in problem (SVE). If $\mathrm{Tr}(\boldsymbol{\Sigma}^*) = \mathrm{Tr}(\boldsymbol{\Sigma}^\diamond)$ the claim is proven. Otherwise, $\boldsymbol{\Sigma}^*$ is not a feasible solution for problem (SVE), and therefore for $\boldsymbol{\mu}^* \in \arg\min_{\boldsymbol{\mu}\in\mathcal{M}(\boldsymbol{\Sigma}^*)} \mathbf{e}'\boldsymbol{\mu}$ it must hold that $\mathbf{e}'\boldsymbol{\mu}^* > 1$. Moreover, since $\boldsymbol{\Sigma}^*$ is optimal for problem (22) it also satisfies $\sigma(\boldsymbol{\Sigma}^*) \leq 1$ and consequently

$$1 < \mathbf{e}'\boldsymbol{\mu}^* \leq \frac{\pi}{2}\sigma(\boldsymbol{\Sigma}^*) \leq \frac{\pi}{2}.$$

The definition of $\mathcal{M}$ and $\boldsymbol{\mu}^*$ imply $\frac{2}{\pi}\boldsymbol{\mu}^* \in \mathcal{M}(\frac{\pi}{2}\boldsymbol{\Sigma}^*)$ and it follows that $\frac{\pi}{2}\boldsymbol{\Sigma}^*$ is feasible for problem (SVE). Since $\boldsymbol{\Sigma}^\diamond$ is optimal for problem (SVE) it is also feasible for problem (22) and therefore, $\mathrm{Tr}(\boldsymbol{\Sigma}^*) \leq \mathrm{Tr}(\boldsymbol{\Sigma}^\diamond) \leq \mathrm{Tr}(\frac{\pi}{2}\boldsymbol{\Sigma}^*) = \frac{\pi}{2}\,\mathrm{Tr}(\boldsymbol{\Sigma}^*)$. □

Theorem 3 *Let $\tilde{\mathbf{E}} \in \mathbb{R}^{r\times n}$ be a matrix such that $\tilde{\mathbf{E}} \neq 0$, and let $J_0 = \{i \in [m] : \|\boldsymbol{\mathcal{I}}_i\tilde{\mathbf{E}}'\|_F \neq 0\}$ where, $\|\cdot\|_F$ denoted the Frobenius norm. The optimal solution to problem (SVE) with parameter matrix $\tilde{\mathbf{E}}$ is given explicitly by:*

$$\boldsymbol{\Sigma}^{\diamond} = \left(\sum_{i=1}^{m}\left\|\boldsymbol{\mathcal{I}}_i\tilde{\mathbf{E}}'\right\|_F\right)\tilde{\mathbf{E}}\left(\sum_{i\in J_0}\frac{1}{\left\|\boldsymbol{\mathcal{I}}_i\tilde{\mathbf{E}}'\right\|_F}\boldsymbol{\mathcal{I}}_i\right)\tilde{\mathbf{E}}', \quad \mu_i^{\diamond} = \frac{\left\|\boldsymbol{\mathcal{I}}_i\tilde{\mathbf{E}}'\right\|_F}{\sum\limits_{i=1}^{m}\left\|\boldsymbol{\mathcal{I}}_i\tilde{\mathbf{E}}'\right\|_F},\; i \in [m] \tag{23}$$

and can be computed in $O(n^2r^2)$ operations.

The proof is given in appendix C.

We refer to the filter, resulting from the SDR of problem (RP), as the *Rolling GAARF with horizon length S* (*RGAARF-S*). Algorithm 1 summaries the procedure which generated *RGAARF-S* for a given S.

Algorithm 1 *RGAARF-S*

```
For $T = 1, 2, \ldots$
(1) Compute the optimal $\boldsymbol{\Sigma}_{\mathbf{z}_T}$ for problem (SVE) using equation (23) with $\tilde{\mathbf{E}}_z$ derived from $\hat{\mathbf{A}}_T$.
(2) Construct matrices $\mathbf{C}$ and $\mathbf{D}$ using the ellipsoid matrices defined in $\Omega_T^S$ and the appropriate coefficients
    derived from $\mathbf{A}_T$.
(3) Solve the SDR of problem (RP), given in (DSDR-P), and obtain the optimal $\mathbf{K}^*_{T,S}$ using the interior point
    algorithm described in appendix A.
(4) Use $\mathbf{K}^*_{T,S}$ to update $\mathbf{A}_T$ and $\hat{\mathbf{A}}_{T+1}$ as described in appendix D.
(5) Compute the optimal $\boldsymbol{\Sigma}_{\boldsymbol{\delta}_T}$ for problem (SVE) using equation (23) with $\tilde{\mathbf{E}} = \mathbf{K}^*_{T,S}\mathbf{C} + \mathbf{D}$.
```

Notice that for each T three optimization problems (computing $\mathbf{K}_{T,S}$, $\boldsymbol{\Sigma}_{\boldsymbol{\delta}_T}$ and $\boldsymbol{\Sigma}_{\mathbf{z}_T}$) are solved, each with a constant problem size of $n \approx O(S)$. Therefore, this procedure guarantees a good worst case performance of $\mathbf{K}^*_{T,S}$ while ensuring fixed computational cost per time step.

When the solution of problem (SVE) generates a matrix $\boldsymbol{\Sigma}$ which is PSD instead of PD, the ellipsoid obtained is degenerate. We can project the ellipsoid constraint to a lower dimension. Using the normalized vectors, this can be done by eliminating (nullifying) the appropriate indices (corresponding to zero eigenvalues) . The problem can therefore be solved in this lower dimension, and the worst case noise vector can be retrieved by using the Moore-Penrose pseudoinverse instead of the regular inverse. Therefore, both problems (RP) and (SVE) are well defined regardless of the invariability of $\boldsymbol{\Sigma}$.

There is clearly a trade-off between accuracy of our solution and problem size. The main drawback of using this method is the loss of information, which may cause an even higher overestimation of the worst case (in addition to the SDR approximation), thus avoiding the curse of dimensionality comes with a price. Therefore, an crucial part of the implementation is the decision on the value of S. Choosing $S = T$ will result in the *GAARF*, while choosing $S = 1$ will result in the classical one-step update SVE. In the next section we will show experimental results pertaining to the algorithms we presented and investigate the choice of S and its impact on the algorithm's performance.

5 Experimental Results for The Tracking Problem

In this section we address the tracking problem presented in Sect. 2 and compare the performance of the KF to the robust filters: *GRF*, *GAARF* and *RGAARF*. We also investigate the implication of the adjustable horizon S of *RGAARF* on its performance and computational cost.

We consider tracking an object moving in an unconstrained, two dimensional plain. Let $\mathbf{x}_t = (\mathbf{x}_t^1, \mathbf{x}_t^2)$ denote the object's state vector, where $\mathbf{x}_t^1 \in \mathbb{R}^2$ is a two dimensional position of the object at time t, $\mathbf{x}_t^2 \in \mathbb{R}^2$ is a two dimensional velocity of the object at time t, and the system noise $\mathbf{a}_t \in \mathbb{R}^2$ is the object's acceleration in some (unknown) direction. Furthermore, only the location of the object is observed, and at each point in time the measurement $\mathbf{y}_t \in \mathbb{R}^2$ is afflicted with measurement noise $\mathbf{v}_t \in \mathbb{R}^2$. The system's dynamics and observation processes are given by the equations in (1) with matrices:

$$\mathbf{F}_t = \begin{bmatrix} 1 & 0 & \mathit{\Delta} t & 0 \\ 0 & 1 & 0 & \mathit{\Delta} t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{G}_t \quad = \begin{bmatrix} \frac{\mathit{\Delta} t^2}{2} & 0 \\ 0 & \frac{\mathit{\Delta} t^2}{2} \\ \mathit{\Delta} t & 0 \\ 0 & \mathit{\Delta} t \end{bmatrix}, \mathbf{H}_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

where $\mathbf{w}_t = \mathbf{G}_t \mathbf{a}_t$ and $\mathit{\Delta}_t$ is the time difference between the point $t+1$ and point t.

In our example, we aim at estimating the object's location and velocity based on the following assumptions and data:

- Measurements are obtained every $\mathit{\Delta}_t = 1sec$.
- The acceleration $\mathbf{a}_t$ is bounded by $\|\mathbf{a}_t\| \leq a_{max}$ for $\alpha_t = a_{max} = 2m/sec^2$, and therefore $\mathbf{Q}_t^a = \mathbf{I}$ and $\mathbf{Q}_t = \mathbf{G}_t' \mathbf{G}_t$.
- The measurement noise is contained in some known ellipsoid which is defined by matrix $\mathbf{R}_t = \mathbf{I}$ and size $v_{max} = \beta_t = 20m$. Since the assumption of adversarial measurements is not necessarily realistic we may assume alternatively that the measurements are indeed of a statistical nature with mean zero and covariance $\mathbf{R} \equiv \mathbf{R}_t$ and such that $\beta \equiv \beta_t$, ensures that the noise is inside the defined ellipsoid with high probability. This can be explained by using the multivariate Chebyshev inequality [19] for the Mahalanobis distance $\text{Mahal}_{\mathbf{R}}(\mathbf{v}_t) = \mathbf{v}_t \mathbf{R}^{-1} \mathbf{v}_t$ to obtain:
$$\text{Prob}(\mathbf{v}_t \mathbf{R}^{-1} \mathbf{v}_t > \beta^2) \leq \frac{d}{\beta^2}$$
where $\mathbf{v}_t \in \mathbb{R}^d$ (in our case $d = 2$). If we further assume that $\mathbf{v}_t$ has a Gaussian distribution, then we can conclude that $\text{Mahal}_{\mathbf{R}}(\mathbf{v}_t)$ has Chi-square distribution with d degrees of freedom.
- Initially the object starts its movement from some unknown point which is assumed to be within a $20m$ radius of point $(0, 0)$ (initial detection) and moves with unknown velocity which is at most $10m/sec$. Therefore, the initial estimation error is constrained by $\|\boldsymbol{\delta}_0^1\| \leq 20m$ and $\|\boldsymbol{\delta}_0^2\| \leq 10m/sec$.
- The object is observed for a duration of $T = 50$ seconds.

All the experiments hereinafter were conducted on a 64Bit Intel Core i5-M3340 at 2.7GHz with 8GB RAM. The code was ran using MATLAB v2010a.

We will use this example to compare the *GRF*, and *GAARF* to the classical KF. To do so we must first decide which KF to use. We chose five different KFs, each corresponding to covariances matrices $\mathbf{R}$ and $\mathbf{Q}^a$ which ensure (via Mahalanobis distance) that the probability of $\mathbf{a}$ and $\mathbf{v}$ being inside the above defined ellipsoids is 0.5, 0.7, 0.8, 0.9 and 0.99. We first compared the *Theoretical Worst Case Error* (*TWCE*) for each filter, for each time step, given by the SDP relaxation. To verify that this is not a mere upper bound, but the actual worst case performance, we obtained a lower bound using the LGA algorithm (described in [28]). Indeed, for all instances (all filters and all time steps) the *SDP* solution was (up to precision of 10^{-3}) the actual worst case value. The result are presented in Fig. 2.

The *GAARF* worst case error norm was around 23.7 for the entire scenario (reaching a high of 25.6 in the 3rd time step), while all of the KFs had a worst case of more then 48 (in steady state). Furthermore, we see that while both the KF and *GAARF* reach a steady state, where the worst case estimation error is constant, *GRF* does not, and it exhibits high volatility and worst case which is not much better than that of the KFs. We attribute this volatility to the inability of the *GRF* to correct past decisions, due to lack in degrees of freedom.

In order to test this theoretical performance we conducted two sets of experiments, so that at each point in time the acceleration and measurement noise are chosen at random:

1. Adversarial object and measurement behavior - The directions of the acceleration, measurement error and initial estimation error are chosen uniformly at random at each point in time, while the magnitude is the maximum allowed. For example, $\|\mathbf{a}_t\| = a_{max}$ for all t but at each point in time we chose the acceleration direction.
2. Adversarial and purposeful object and random measurements - This Scenario depicts an adversarial object with a specific purpose, accelerating in a certain direction for the entire scenario. The directions and magnitude of the measurement error and initial estimation error are chosen uniformly at random for each point in time. The acceleration direction is chosen uniformly at random once, at the beginning of the scenario, and proceeds in that direction for the rest of the scenario with maximal magnitude.

These scenarios refer to cases which are not purely random, since purely random scenarios will correspond with the assumptions for which KF is optimal. Furthermore, in this scenario, the filter will rarely encounter a realization of the noise vector which will produce a high estimation error. Moreover, as we stated in the introduction, assuming a randomly moving target is not realistic for the problem of target tracking.

We compare the simulation results for each filter f, each time step t and each simulation i. Denoting the real state of the system (*ground truth*) as $\mathbf{x}_{t,i}$ and the filter estimation as $\hat{\mathbf{x}}_{t,i}(filter)$ we calculated the *Simulated Error* (SE):

$$SE_t^i(f) = \|\mathbf{x}_{t,i} - \hat{\mathbf{x}}_{t,i}(f)\|.$$

The KF performance was consistent and similar for all values of the five chosen probabilities. Therefore, only the filter with the best performance, which is the KF with probability of $p = 0.8$, is shown from now on. The mean and maximal SE values for each filter in each time step are

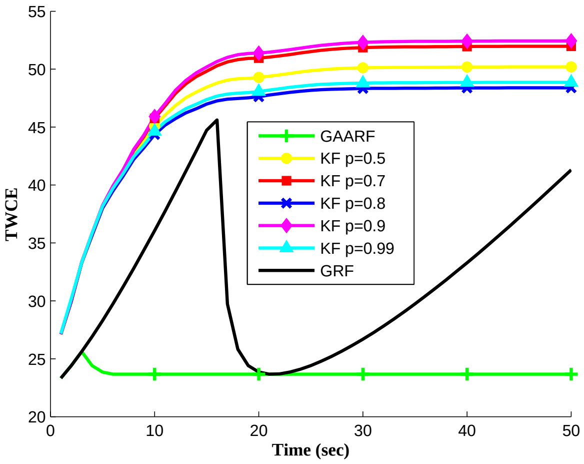
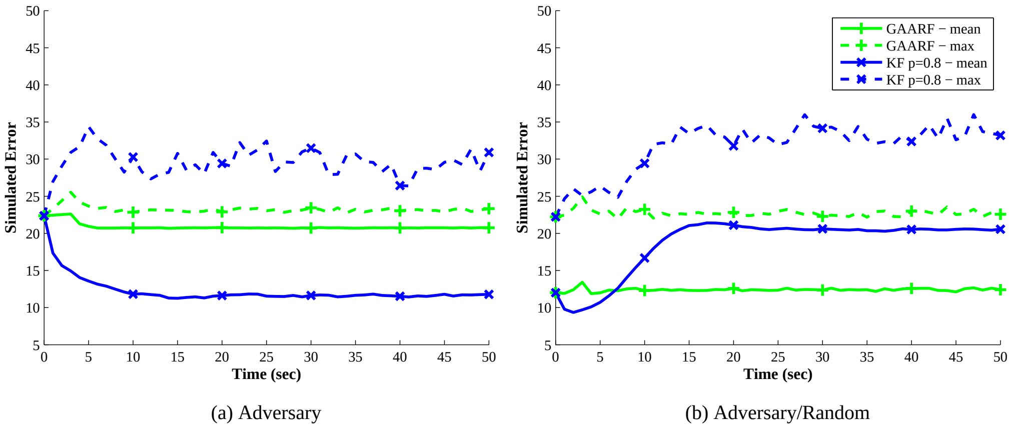


Fig. 2 Kalman vs. Robust - Theoretical Worst Case Error (TWCE)

presented in Fig. 3. The *GRF* exhibited the same unstable behavior as in the theoretical results and was generally inferior both to the KF and *GAARF* and therefore was omitted from the figures. The *GAARF* is better at protecting against *worst case results* for both scenarios, giving it a clear advantage over the KF, with error values 20% lower for the first scenario (Fig. 3a) and up to 30% lower for the second scenario (Fig. 3b). We would expect that the average performance of the KF would be superior, since it is designed to minimize mean performance. However, in the second scenario, presented in Fig. 3b, the KF exhibits inferior performance with mean error of 20 vs. 12 for *GAARF*. This is due to the fact that the KF noise assumptions do not hold in this case, and therefore the KF is not optimal even for the average case.

To better understand the differences between the KF and *GAARF* performance we looked at the KF worst case scenario. In Fig. 4 we see the spatial 2D location of the object and the filters' location estimation for various time steps. The dot indicates the actual location of the object, the 'x' marks the KF predicted location and the '+' the *GAARF* prediction. Furthermore, we calculated the 95% confidence level for the KF, derived from its estimation error covariance matrix, shown as a dashed line, and for both filters we calculated the true estimation set derived from the worst case (through SDP relaxation as shown in Sect. 4.2), presented as a full line. Firstly, observe that the KF 95% confidence interval until time $t = 8$ is an over-estimation of the true estimation set, and later an under-estimation. Therefore, any arbitrary confidence level, chosen in advance, may lead to either an over-estimation or an under-estimation, and in some cases such as at time steps 22, 36 and 50, an incorrect confidence interval may cause us to lose the object. Moreover, both the confidence interval and the true estimation set of the KF are much larger than the estimation set of the *GAARF*, which always captures the ground truth. Specifically, for $t = 50$ the Kalman filter has a confidence interval and estimation set radius of about 40 and $47m$ respectively, while for the *GAARF* it is only $20m$. We can conclude that the *GAARF* worst case estimation is closer to the true object location, and that the estimation set it provides is equivalent to a 100% confidence interval, guaranteeing that the true object location is within the set.

As mentioned earlier, the problem with *GAARF* is that its computational complexity increases with time. Looking at the rolling horizon filter *RGAARF* we examine the trade off between computational complexity per iteration and performance. *RGAARF* was ran using five rolling time-window sizes of 1,2,5,10, and 20 seconds. Recall that *RGAARF*-50 is equivalent to



(a) Adversary

(b) Adversary/Random

Fig. 3 Kalman vs. Robust - Simulated Error

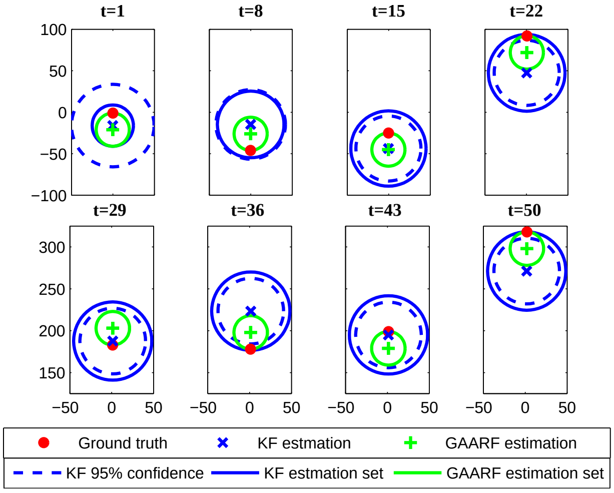


Fig. 4 Location Estimation KF vs. *GAARF*

GAARF, and that *RGAARF*-1 is equivalent to classical SVE. Fig. 5 presents the computational time per time step for each of these filters. As we expected the steady state time (starting from $2S$) increase as the horizon becomes longer, and the *GAARF* takes at least an order of magnitude longer (for time step 50) than the all the *RGAARF*. We omitted the running times for $S = 2$ since they were similar to that of $S = 1$.

To investigate whether the algorithm's performance deteriorates with the length of the horizon we look at the *Theoretical Worst Case Error Ratio* (TWCER) for each filter f at each time step t, defined by:

$$TWCER_t(f) = \frac{TWCE_t(f)}{TWCE_t(GAARF)}.$$

More specifically, for each filter we looked at the TWCE upper bound, which is generated while computing the filter, and at the actual TWCE, taking into account the entire noise vector. Notice that for *GAARF* both bounds are identical. The results are given in Fig. 6. It is not clear that

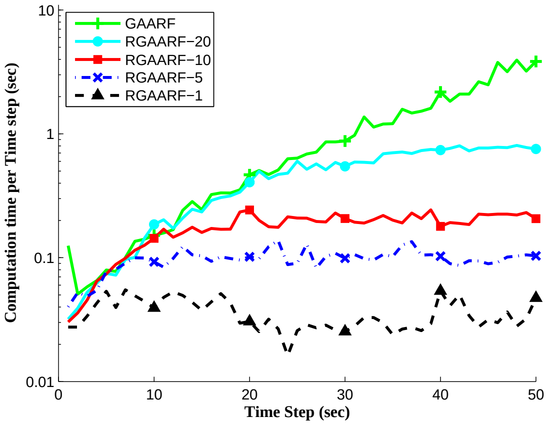


Fig. 5 *RGAARF* running times

longer horizons are indeed better, since it is debatable weather $RGAARF$-20 indeed performs better than $RGAARF$-10. Comparing Fig. 6a and Fig. 6b, we observe that the upper bounds obtained for TWCE are not tight and, as expected, is looser as the horizon length becomes shorter. Moreover, these filters exhibit a recurring pattern in their performance where the cycle length is determined by the horizon length. The seemingly stable behavior of the filters with horizon length 1 and 2 is actually a pattern with short cycle length. Surprisingly, $RGAARF$-1 is more stable and performs better than the GRF, although it uses less information.

Therefore in theory, $RGAARF$-10 achieves the best balance between estimation set accuracy and information loss, thus obtaining a relatively stable filter. This filter is computed in less than a quarter of a second per time step, and it guarantees theoretical worst case performance which deviates at most 9% from that of the $GAARF$.

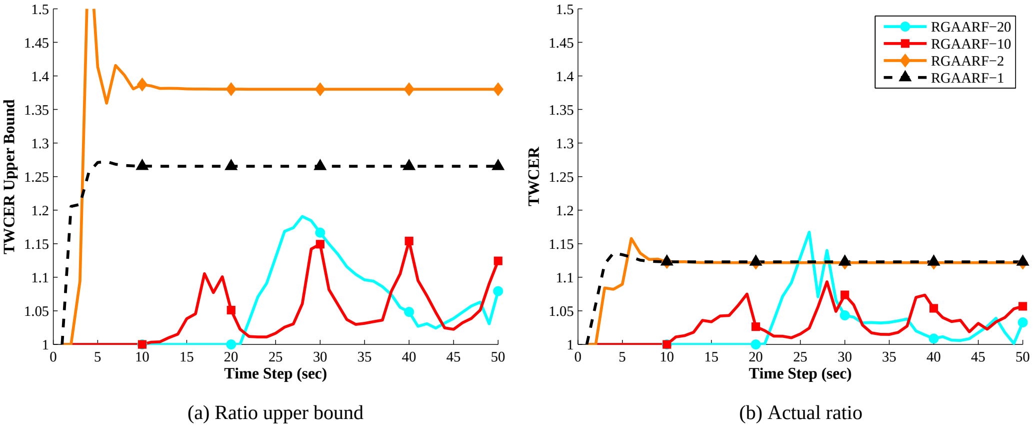
Simulation results comparing the various rolling horizon filters and $GAARD$ support this assertion. More specifically, in the adversarial and random scenarios the mean performance of all $RGAARF$ was close to that of $GAARF$ and for the adversarial-random scenario $RGAARF$-1 and 2 actually had better average performance while the average performance of $RGAARF$-10 was up to 14% worse, and $RGAARD$-20 up to 40% worse. However, worst case performance of short horizon filters was $5\% - 10\%$ worse than the $GAARF$ while longer horizon filters, displayed performance close to that of $GAARF$.

In conclusion, for the tracking application, using $RGAARF$-10 rather than $GAARF$ will have the advantage of maintaining the same performance both theoretically and practically, while reducing the running time significantly. The worst case performance for both adversarial or partly adversarial scenarios is better than that of the KF.

The results presented here were essentially replicated when we changed the maximum measurement error to $30m$ and the maximum initial velocity error to $5\frac{m}{sec}$.

6 Summary

In this paper we presented a new SDP based set-estimation approach for robust estimation focusing mainly on for the case of *tracking*. We developed numerical algorithms to solve the resulting optimization problems for the case of bounded uncertainty. These solutions provide



(a) Ratio upper bound

(b) Actual ratio

Fig. 6 $RGAARF$ Theoretical Performance good approximations for worst case performance, when the filter matrices are given, and for the optimal robust filter. Our basic approach works well for multi-period problems, however the computational cost increases with the time step. Therefore, a rolling horizon, with fixed window size, is used to achieve a fixed computational cost per time step, and can be tuned to work well both in theory and in practice. Finally, we showed how to extend this approach to include both random and bounded noise, thus generalizing the KF for expected worst case.

Future research may consider incorporating knowledge on a partially bounded state vector into this scheme, and exploring the advantages and disadvantages of using different estimation error measures.

Appendix A Implementation of HRVW-alg to saddle point problem

Consider problem (DSDR-P) with parameters $\mathbf{C} \in \mathbb{R}^{\tilde{q}\times n}$ and $\mathbf{D} \in \mathbb{R}^{r\times n}$, and $\boldsymbol{\mathcal{I}}_i \in \mathbb{R}^{n\times n}$ $i = 1,\dots,m$, and decision variables Let $\mathbf{K} \in \mathbb{R}^{r\times\tilde{q}}$, $\boldsymbol{\mu} \in \mathbb{R}^m$ (we negate non-negativeness of μ since μ must be non-negative in order to satisfy the constraints), where r and q are and therefore $O(\tilde{q}) = O(m) = O(n) = O(T)$. We add variable $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{(n+r)\times(n+r)}$ such that:

$$\boldsymbol{\mathcal{Z}} = \begin{bmatrix} \sum_{i=1}^m \mu_i \boldsymbol{\mathcal{I}}_i & \mathbf{D}' + \mathbf{C}'\mathbf{K}' \\ \mathbf{KC} + \mathbf{D} & \mathbf{I} \end{bmatrix}, \ \boldsymbol{\mathcal{Z}} \succeq 0.$$

This problem satisfies the Slater condition for $\mathbf{K} = \mathbf{0}$ and $\mu_i \equiv \alpha = \|\mathbf{D}\|^2 + 1$ $\forall i = 1,\dots,m$ since appropriate $\boldsymbol{\mathcal{Z}}$ is PD and so the problem is regular.

Defining the dual variable $\mathbf{Y}$ to be a symmetric matrix such that

$$Y = \begin{bmatrix} \mathbf{Y}^{11} & \mathbf{Y}^{12} \\ \mathbf{Y}^{21} & \mathbf{Y}^{22} \end{bmatrix} \succeq 0,$$

we can then formulate the following dual problem.

$$\begin{aligned} \max_{\mathbf{Y}} \quad & \mathbf{Y} \circ \begin{bmatrix} \mathbf{0} & -\mathbf{D}' \\ -\mathbf{D} & -\mathbf{I} \end{bmatrix} = -2\mathbf{Y}^{12} \circ \mathbf{D}' - \mathbf{I} \circ \mathbf{Y}^{22} \\ s.t. \quad & \\ & \tilde{\boldsymbol{\mathcal{I}}}_i \circ \mathbf{Y} = \boldsymbol{\mathcal{I}}_i \circ \mathbf{Y}^{11} = 1 \qquad\qquad i = 1,\dots,m \\ & \tilde{\boldsymbol{\mathcal{C}}}^{ij} \circ \mathbf{Y} = 2\tilde{\mathbf{C}}^{ij} \circ \mathbf{Y}^{12} = 2\mathbf{C}_{i\cdot}\mathbf{Y}^{12}_{\cdot j} = 0 \qquad i = 1,\dots,\tilde{q}, \ j = 1,\dots,r \\ & \mathbf{Y} \succeq 0 \end{aligned} \tag{24}$$

where matrices $\tilde{\boldsymbol{\mathcal{I}}}_i$ and $\tilde{\boldsymbol{\mathcal{C}}}^{ij}$ are given by:

$$\tilde{\boldsymbol{\mathcal{I}}}_i = \begin{bmatrix} \boldsymbol{\mathcal{I}}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \ \tilde{\boldsymbol{\mathcal{C}}}^{ij} = \begin{bmatrix} \mathbf{0} & \tilde{\mathbf{C}}^{ij} \\ (\tilde{\mathbf{C}}^{ij})' & \mathbf{0} \end{bmatrix}, \ \tilde{\mathbf{C}}^{ij} \in \mathbb{R}^{n\times r} : \tilde{C}^{ij}_{kl} = \begin{cases} C_{ik} & l = j \\ 0 & \text{otherwise} \end{cases}.$$

We can see that this problem also satisfies the Slater conditions, since strict feasibility hold for $\mathbf{Y}$ such that $\mathbf{Y}^{12} = 0$, $\mathbf{Y}^{11} = \mathrm{Diag}(\sum_{i=1}^m \frac{1}{n_i}\boldsymbol{\mathcal{I}}_i\mathbf{e})$ and $\mathbf{Y}^{22} = \mathbf{I}$. Strict feasibility of both primal and dual problem ensure strong duality, and therefore a duality gap of zero.

Notice that the second set of constraints is actually equivalent to $\mathbf{C}' \cdot \mathbf{Y} = \mathbf{0}$, where $\mathbf{0}$ here denotes an $\tilde{q} \times r$ matrix of all zeros. Therefore, if $\tilde{q} \geq n$ and $rank(\mathbf{C}) = n$ for any feasible solution $\mathbf{Y}^{12} = \mathbf{0}$ holds and the optimal solution satisfies $\mathbf{Y}^{22} = \mathbf{0}$, for any feasible $\mathbf{Y}^{11}$ and the problem value is zero. This also means that $\boldsymbol{\mu} = \mathbf{0}$ and $\mathbf{K}$ is simply some solution for the equation $\mathbf{KC} + \mathbf{D} = 0$. If $rank(\mathbf{C}) < n$, we can take only the set of linearly independent rows of

$\mathbf{C}$ and corresponding columns of $\mathbf{K}$ thus having $\mathbf{C}$ of full row rank and $\tilde{q} < n$. Therefore, without loss of generality we can assume $\tilde{q} < n$.

Since strong duality holds an optimal solution can be found using the $HRVW-alg$ algorithm based on [13]. $HRVW-alg$ conducts interior point iterations in order to reduce the duality gap. In order to do so we have to define a linear operator $B(\mathbf{Y}) : \mathbb{R}^{(n+r)\times(n+r)} \to \mathbb{R}^{m+rq}$ such that $B(\mathbf{Y}) = [B_1(\mathbf{Y}), \dots, B_{m+rq}(\mathbf{Y})]$ where $B_j(\mathbf{Y}) = \mathbf{B}_j \circ \mathbf{Y}$. In our case, denoting the subsets of indexes $J_1 = \{1, \dots, m\}$, $J_2 = \{m+1, \dots, m+rq\}$ we have that:

$$\mathbf{B}_i = \begin{cases} \tilde{\boldsymbol{\mathcal{I}}}_i & i \in J_1 \\ \tilde{\mathbf{C}}^{kl} & i \in J_2, l = \lfloor \frac{i-m-1}{\tilde{q}} \rfloor + 1, k = i - m - (l-1)\tilde{q} \end{cases} . \tag{25}$$

Its adjoint $B'(\mathbf{y}) : \mathbb{R}^{m+rq} \to \mathbb{R}^{(n+r)\times(n+r)}$ where $\mathbf{y} = (\boldsymbol{\mu}, \mathbf{K})$, is then defined as:

$$B'(\mathbf{K}, \boldsymbol{\mu}) = \begin{bmatrix} \sum_{i=1}^{m} \mu_i \boldsymbol{\mathcal{I}}_i & \mathbf{C}'\mathbf{K}' \\ \mathbf{KC} & 0 \end{bmatrix}. \tag{26}$$

We can now write the primal and dual problems as:

$$\begin{array}{llll} & \max\limits_{\mathbf{Y}} & \mathbf{A} \circ \mathbf{Y} & & & \max\limits_{\mathbf{y}} \mathbf{b}'\mathbf{y} \\ (P) & s.t. & B(\mathbf{Y}) = \mathbf{b} & & (D) & s.t. \quad B'(\mathbf{y}) - \mathbf{A} \succeq 0 \\ & & \mathbf{Y} \succeq 0 & , & & \end{array}$$

The algorithms consists of several stages:

Algorithm 2 Obtaining Saddle Point by HRVW Scheme

```
Initialization $\mathbf{Y}^0$, $\mathbf{K}^0$, $\boldsymbol{\mu}^0$, $\mathbf{y}^0 = (\boldsymbol{\mu}^0, \mathbf{K}^0)$ and $\boldsymbol{\mathcal{Z}}^0$ such that they are strictly feasible for primal and dual problem respectively . Set $k = 0$.
While $\boldsymbol{\mathcal{Z}}^k \circ \mathbf{Y}^k > \epsilon$
  (1) Calculate $\gamma^k = \frac{\boldsymbol{\mathcal{Z}}^k \circ \mathbf{Y}^k}{2(n+r)}$
  (2) Calculate the slack matrix $\boldsymbol{\mathcal{Z}}^k$ and its inverse $\mathbf{W}^k = (\boldsymbol{\mathcal{Z}}^k)^{-1}$.
  (3) Calculate positive definite matrix $\boldsymbol{\mathcal{P}}^k \in \mathbb{R}^{(m+rq)\times(m+rq)}$, where $\mathcal{P}_{ij}^k = \mathrm{Tr}(\mathbf{B}_i \mathbf{W}^k \mathbf{B}_j \mathbf{Y}^k)$.
  (4) Calculate vector $\mathbf{v}^k = \gamma^k B(\mathbf{W}^k) - \mathbf{b}$.
  (5) Calculate $\boldsymbol{\Delta}_{\mathbf{y}} = (\boldsymbol{\Delta}_{\boldsymbol{\mu}}, \boldsymbol{\Delta}_{\mathbf{K}})$ be the solution of the system $\mathbf{v}^k = \boldsymbol{\mathcal{P}}^k \boldsymbol{\Delta}_{\mathbf{y}}$.
  (6) Update $\boldsymbol{\Delta}_{\mathbf{Z}} = B'(\boldsymbol{\Delta}_{\mathbf{y}})$ and $\boldsymbol{\Delta}_{\mathbf{Y}} = \gamma^k \mathbf{W}^k - \mathbf{Y}^k + \mathbf{W}^k B'(\boldsymbol{\Delta}_{\mathbf{y}}) \mathbf{Y}^k$.
  (7) Find $\alpha_P$ and $\alpha_D$ such that: $\mathbf{Y}^{k+1} + \alpha_P \boldsymbol{\Delta}_Y \succ 0$ and $\boldsymbol{\mathcal{Z}}^{k+1} = \boldsymbol{\mathcal{Z}}^k + \alpha_D \boldsymbol{\Delta}_{\mathbf{Z}} \succ 0$ by line search.
  (8) Update $\mathbf{K}^{k+1} = \mathbf{K}^k + \alpha_D \boldsymbol{\Delta}_{\mathbf{K}}$ and $\boldsymbol{\mu}^{k+1} = \boldsymbol{\mu}^k + \alpha_D \boldsymbol{\Delta}_{\boldsymbol{\mu}}$
  (9) Update $k \leftarrow k + 1$.
End While
Return $(\mathbf{Y}^*, \boldsymbol{\mu}^*, \mathbf{K}^*) = (\mathbf{Y}^k, \boldsymbol{\mu}^k, \mathbf{K}^k)$.
```

We can clearly see that the computational costs consist of inverting $\boldsymbol{\mathcal{Z}}$, computing $\boldsymbol{\mathcal{P}}$, calculating $B(\mathbf{W})$, inverting $\boldsymbol{\mathcal{P}}$, computing $B'(\mathbf{y})$, checking whether $\mathbf{Y}$ and $\boldsymbol{\mathcal{Z}}$ are positive definite and finally computing the duality gap. The inversion of $\boldsymbol{\mathcal{Z}}$ as well as checking whether $\mathbf{Y}$ and $\boldsymbol{\mathcal{Z}}$ are positive definite, can be done by Cholesky decomposition and takes $O((n+r)^3)$ operations. Calculation of $B(\mathbf{W})$ takes only $O(n + r\tilde{q}n)$ operations, since $B_i(\mathbf{W})$ for each $i \in J_1$ takes $O(n_i)$ operations and $B_i(\mathbf{W})$ for each $i \in J_2$ takes $O(n)$ operations. Calculation of $B'(\mathbf{y})$ takes $O(r\tilde{q}n)$ to calculate the off diagonal blocks and $O(n)$ to compute the diagonal. Calculating the duality gap takes $O((n+r)^2)$ operations and inverting $\boldsymbol{\mathcal{P}}$ takes $O((m + r\tilde{q})^3)$. Since $O(\tilde{q}) = O(n)$ this is equivalent to all these operations being done in $O(n^3)$. The only potentially computational problematic operation in computing $\boldsymbol{\mathcal{P}}$, which consists of computing the matrices $\mathbf{B}_i\mathbf{W}$ and $\mathbf{B}_i\mathbf{Y}$ for every $i \in J_1 \bigcup J_2$. Computing $\mathbf{B}_i\mathbf{W}$ takes up to $O((n+r)^3)$ operations for each i, and so we have $O(m + r\tilde{q})$ such multiplications, resulting in potentially up to $O(n^4)$ operations. We will now prove that the special structure of $\mathbf{B}_i$ ensure that this is not the case and $\boldsymbol{\mathcal{P}}$ can be computed in $O(n^3)$ as well.

From symmetry of $\mathbf{W}$, $\mathbf{B}_i$ and $\mathbf{Y}$ can conclude that $\boldsymbol{\mathcal{P}}$ is a symmetric matrix with entries defined as follows.

$$\mathcal{P}_{ij} = \mathrm{Tr}(\mathbf{B}_i\mathbf{W}\mathbf{B}_j\mathbf{Y}) = \mathbf{W} \circ \mathbf{B}_j\mathbf{Y}\mathbf{B}_i = \mathrm{Tr}(\mathbf{W}(\mathbf{B}_j\mathbf{Y}\mathbf{B}_i)') = \mathbf{W} \circ \mathbf{B}_i\mathbf{Y}\mathbf{B}_j.$$

We will now compute $\mathbf{B}_j\mathbf{Y}\mathbf{B}_i$ for each i and j and calculate the computational complexity. We look at each combination of i, j:

1. $i, j \in J_1$: we have that $\mathcal{P}_{ij} = \boldsymbol{\mathcal{I}}_i\mathbf{W}^{11}\boldsymbol{\mathcal{I}}_j \circ \boldsymbol{\mathcal{I}}_i\mathbf{Y}^{11}\boldsymbol{\mathcal{I}}_j$ which is the inner product of block i, j of the $(\cdot)^{11}$ block in each of the matrices. So to compute the whole $m \times m$ sub-matrix $\{\mathcal{P}_{ij}\}_{i,j\in J_1}$ we need $O(n^2)$ operations.
2. $i \in J_1$ and $j \in J_2$: We denote $l = \lfloor \frac{j-m-1}{\tilde{q}} \rfloor + 1, k = j - m - (l-1)\tilde{q}$ (or vice versa) we have that
$$\tilde{\boldsymbol{\mathcal{I}}}_i\mathbf{Y}\tilde{\boldsymbol{\mathcal{C}}}^{kl} = \begin{bmatrix} \boldsymbol{\mathcal{I}}_i\mathbf{Y}^{11} & \boldsymbol{\mathcal{I}}_i\mathbf{Y}^{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tilde{\boldsymbol{\mathcal{C}}}^{kl} = \begin{bmatrix} \boldsymbol{\mathcal{I}}_i\mathbf{Y}^{12}(\tilde{\mathbf{C}}^{kl})' & \boldsymbol{\mathcal{I}}_i\mathbf{Y}^{11}\tilde{\mathbf{C}}^{kl} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$
The matrix $\boldsymbol{\mathcal{I}}_i\mathbf{Y}^{11}\tilde{\mathbf{C}}^{kl}$ has only n_i non zeros elements, in indexes corresponding to rows in partition element i and column l. Each of these elements can be calculated by $\mathbf{Y}^{11}_{s\cdot}\mathbf{C}'_{k\cdot}$, where s is in partition element i, and so the total number of operations needed for the computation is $O(nn_i)$. The matrix $\boldsymbol{\mathcal{I}}_i\mathbf{Y}^{12}(\tilde{\mathbf{C}}^{kl})'$ has only n_i non zero rows of length n each, where each row can be calculated as the product: $Y^{12}_{sl}\mathbf{C}_{k\cdot}$. Therefore, the total number of operation to compute this matrix is $O(nn_i)$. The multiplication with $\mathbf{W}$ includes multiplying only the non zero elements and therefore takes $O(n_i(n+r))$ operations. Consequently computing $\{\mathcal{P}_{ij}\}_{i\in J_1, j\in J_2}$ requires $\sum_{i=1}^{m}\sum_{j=m+1}^{m+r\tilde{q}} O(n_i(n+r)) = O(n(n+r)r\tilde{q}) = O(n^3)$.
3. $i, j \in J_2$: Denoting $l_1 = \lfloor \frac{i-m-1}{\tilde{q}} \rfloor + 1$, $k_1 = i - m - (l_1 - 1)\tilde{q}$, $l_2 = \lfloor \frac{j-m-1}{\tilde{q}} \rfloor + 1$, and
$k_2 = j - m - (l_2 - 1)\tilde{q}$ we have that:
$$\tilde{\boldsymbol{\mathcal{C}}}^{k_1l_1}\mathbf{Y}\tilde{\boldsymbol{\mathcal{C}}}^{k_2l_2} = \begin{bmatrix} \tilde{\mathbf{C}}^{k_1l_1}\mathbf{Y}^{21} & \tilde{\mathbf{C}}^{k_1l_1}\mathbf{Y}^{22} \\ (\tilde{\mathbf{C}}^{k_1l_1})'\mathbf{Y}^{11} & (\tilde{\mathbf{C}}^{k_1l_1})'\mathbf{Y}^{12} \end{bmatrix} \tilde{\boldsymbol{\mathcal{C}}}^{k_2l_2} = \begin{bmatrix} \tilde{\mathbf{C}}^{k_1l_1}\mathbf{Y}^{22}(\tilde{\mathbf{C}}^{k_2l_2})' & \tilde{\mathbf{C}}^{k_1l_1}\mathbf{Y}^{21}\tilde{\mathbf{C}}^{k_2l_2} \\ (\tilde{\mathbf{C}}^{k_1l_1})'\mathbf{Y}^{12}(\tilde{\mathbf{C}}^{k_2l_2})' & (\tilde{\mathbf{C}}^{k_1l_1})'\mathbf{Y}^{11}\tilde{\mathbf{C}}^{k_2l_2} \end{bmatrix}.$$
Notice the following:
   - The first matrix $\tilde{\mathbf{C}}^{k_1l_1}\mathbf{Y}^{22}(\tilde{\mathbf{C}}^{k_2l_2})' = Y^{22}_{l_1l_2}\mathbf{C}'_{k_1\cdot}\mathbf{C}_{k_2\cdot}$.
   - Matrix $(\tilde{\mathbf{C}}^{k_1l_1})'\mathbf{Y}^{11}\tilde{\mathbf{C}}^{k_2l_2}$ is a zero matrix with one non zero element in (l_1, l_2) with value $\mathbf{C}_{k_1\cdot}\mathbf{Y}^{11}\mathbf{C}'_{k_2\cdot}$.
   - Matrix $\tilde{\mathbf{C}}^{k_1l_1}\mathbf{Y}^{21}\tilde{\mathbf{C}}^{k_2l_2}$ will have non-zero entries only in column l_2 which will equal $\mathbf{C}'_{k_2\cdot}(\mathbf{Y}^{21}_{l_1\cdot}\mathbf{C}'_{k_1\cdot})$. Similarly, we have $(\tilde{\mathbf{C}}^{k_1l_1})'\mathbf{Y}^{12}(\tilde{\mathbf{C}}^{k_2l_2})' = (\tilde{\mathbf{C}}^{k_2l_2}\mathbf{Y}^{21}\tilde{\mathbf{C}}^{k_1l_1})'$ which means that only the l_1 row is non-zero with value $\mathbf{C}_{k_1\cdot}(\mathbf{Y}^{21}_{l_2\cdot}\mathbf{C}'_{k_2\cdot})$.

   Therefore, from symmetry of $\mathbf{W}$ the total calculation sums up to:
$$\begin{aligned}\mathrm{Tr}(\mathbf{W}\tilde{\boldsymbol{\mathcal{C}}}^{k_1l_1}Y\tilde{\boldsymbol{\mathcal{C}}}^{k_2l_2}) &= Y^{22}_{l_1l_2}(\mathbf{C}_{k_2\cdot}\mathbf{W}^{11}\mathbf{C}'_{k_1\cdot}) + W^{22}_{l_1l_2}(\mathbf{C}_{k_1\cdot}\mathbf{Y}^{11}\mathbf{C}'_{k_2\cdot}) \\ &+ (\mathbf{W}^{21}_{l_2\cdot}\mathbf{C}'_{k_2\cdot})(\mathbf{Y}^{21}_{l_1\cdot}\mathbf{C}'_{k_1\cdot}) + (\mathbf{W}^{21}_{l_1\cdot}\mathbf{C}'_{k_1\cdot})(\mathbf{Y}^{21}_{l_2\cdot}\mathbf{C}'_{k_2\cdot})\end{aligned} \tag{27}$$
   Both $\mathbf{W}^{11}\mathbf{C}'_{k_1\cdot}$ and $\mathbf{Y}^{11}\mathbf{C}'_{k_1\cdot}$ can be computed once for every k_1 at a total cost of $O(n^2\tilde{q}) = O(n^3)$, creating $2\tilde{q}$ vectors of length n. Multiplying each of this vectors by each $\mathbf{C}'_{k_2\cdot}$ takes $O(n\tilde{q}^2) = O(n^3)$. Finally multiplying these $\tilde{q}^2$ results by the appropriate scalars $Y^{22}_{l_1l_2}$ and $W^{22}_{l_1l_2}$ for each combination of (l_1, l_2) takes $O(r^2\tilde{q}^2)$ operations. Similarly, computing $\mathbf{W}^{21}_{l\cdot}\mathbf{C}'_{k\cdot}$

and $\mathbf{Y}^{21}_{l\cdot}\mathbf{C}'_{k\cdot}$ takes total of $O(r\tilde{q}n)$ for all combinations of l and k, resulting in $r\tilde{q}$ scalars each. The multiplication of all combinations of this scalars costs $O(r^2\tilde{q}^2) = O(n^2)$. Therefore, we can compute $\{\mathcal{P}_{ij}\}_{i,j\in J_2}$ in $O(n^3)$ operations.

Accordingly, the total computational cost of constructing $\boldsymbol{\mathcal{P}}$ is $O(n^3)$.

In conclusion, the computational cost of solving problem (DSDR-P) each step of the interior point algorithm 2 is $O((n+r\tilde{q})^3)$ operations per iterations and $O(\sqrt{n+r}\log\left(\frac{\mathrm{Tr}(\boldsymbol{\mathcal{Z}}_0\mathbf{Y}_0)}{\epsilon}\right))$ iteration to obtain duality gap of ϵ.

Appendix B Implementation of HRVW-alg to worst case expectation minimization approximation problem

Consider problem (α-SDP-EP) where objective function is divided by the constant $(1+\frac{1}{\alpha})$. We denote

$$\boldsymbol{\mathcal{Z}}^1 = \begin{bmatrix} \sum\limits_{i=1}^{m}\mu_i\boldsymbol{\mathcal{I}}_i & (\mathbf{KC}^1+\mathbf{D}^1)' \\ (\mathbf{KC}^1+\mathbf{D}^1) & \mathbf{I} \end{bmatrix},\ \boldsymbol{\mathcal{Z}}^2 = \begin{bmatrix} \boldsymbol{\mathcal{U}} & (\mathbf{KC}^2+\mathbf{D}^2) \\ (\mathbf{KC}^1+\mathbf{D}^1)' & \mathbf{I} \end{bmatrix},\ \boldsymbol{\mathcal{Z}} = \begin{bmatrix} \boldsymbol{\mathcal{Z}}^1 & 0 \\ 0 & \boldsymbol{\mathcal{Z}}^2 \end{bmatrix},\ \boldsymbol{\mathcal{Z}} \succeq 0$$

Defining the dual variable $\mathbf{Y}$ to be a block diagonal matrix, with blocks $\mathbf{Y}^1 \in \mathbb{R}^{(\bar{n}_1+r)\times(\bar{n}_1+r)}$ and $\mathbf{Y}^2 \in \mathbb{R}^{(r+\bar{n}_2)\times(r+\bar{n}_2)}$ which themselves are separated to blocks.

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}^1 & 0 \\ 0 & \mathbf{Y}^2 \end{bmatrix},\ \mathbf{Y}^1 = \begin{bmatrix} \mathbf{Y}^1_{[11]} & \mathbf{Y}^1_{[12]} \\ \mathbf{Y}^1_{[21]} & \mathbf{Y}^1_{[22]} \end{bmatrix} \succeq 0,\ \mathbf{Y}^2 = \begin{bmatrix} \mathbf{Y}^2_{[11]} & \mathbf{Y}^2_{[12]} \\ \mathbf{Y}^2_{[21]} & \mathbf{Y}^2_{[22]} \end{bmatrix} \succeq 0,$$

we can formulate the following dual problem.

$$\begin{aligned}
&\max_{\mathbf{Y}^1,\mathbf{Y}^2} -2\mathbf{Y}^1_{[12]}\circ\mathbf{D}^{1'} - \mathbf{I}\circ\mathbf{Y}^1_{[22]} - 2\mathbf{Y}^2_{[12]}\circ\mathbf{D}^2 - \mathbf{I}\circ\mathbf{Y}^2_{[22]} \\
&s.t. \\
&\quad \tilde{\boldsymbol{\mathcal{I}}}_i\circ\mathbf{Y} \equiv \boldsymbol{\mathcal{I}}_i\circ\mathbf{Y}^1_{[11]} = \alpha && i=1,\dots,m \\
&\quad \tilde{\boldsymbol{\mathcal{C}}}^{1ij}\circ\mathbf{Y}^1 + \tilde{\boldsymbol{\mathcal{C}}}^{2ij}\circ\mathbf{Y}^2 = 2\tilde{\mathbf{C}}^{1ij}\circ\mathbf{Y}^1_{[12]} + 2\tilde{\mathbf{C}}^{2ij}\circ\mathbf{Y}^2_{[12]} = 0 && i=1,\dots,q,\ j=1,\dots,r \\
&\quad \underline{\boldsymbol{\mathcal{I}}}^{ij}\circ\mathbf{Y}^2 \equiv (Y^2_{[11]})_{ij} = \mathbf{1}_j(i) && i,j=1,\dots,r \\
&\quad \mathbf{Y}^1 \succeq 0 \\
&\quad \mathbf{Y}^2 \succeq 0
\end{aligned} \tag{28}$$

where $\mathbf{1}_j(i)$ is the Dirac measure, and matrices $\underline{\boldsymbol{\mathcal{I}}}^{ij}$, $\tilde{\boldsymbol{\mathcal{I}}}_i$ and $\tilde{\boldsymbol{\mathcal{C}}}^{\ell ij}$, $\tilde{\mathbf{C}}^{1ij} \in \mathbb{R}^{\bar{n}_1\times r}$ and $\tilde{\mathbf{C}}^{2ij} \in \mathbb{R}^{r\times\bar{n}_2}$ are given by:

$$\tilde{\boldsymbol{\mathcal{I}}}_i = \begin{bmatrix} \boldsymbol{\mathcal{I}}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},\ \tilde{\boldsymbol{\mathcal{C}}}^{\ell ij} = \begin{bmatrix} \mathbf{0} & \tilde{\mathbf{C}}^{\ell ij} \\ (\tilde{\mathbf{C}}^{\ell ij})' & \mathbf{0} \end{bmatrix},\ \ell = 1,2$$

$$\underline{\boldsymbol{\mathcal{I}}}^{ij}_{kl} = \begin{cases} 1 & k=i,\ l=j \\ 0 & \text{otherwise} \end{cases},\ \tilde{C}^{1ij}_{kl} = \begin{cases} C^1_{ik} & l=j \\ 0 & \text{otherwise} \end{cases},\quad \tilde{C}^{2ij}_{kl} = \begin{cases} C^2_{il} & k=j \\ 0 & \text{otherwise} \end{cases}.$$

Both primal and dual problem are strictly feasible with $\mathbf{K} = 0$, $\boldsymbol{\mu} = (\|\mathbf{D}^1\|^2+1)\mathbf{e}$ and $\boldsymbol{\mathcal{U}} = (\|\mathbf{D}^2\|^2+1)\mathbf{I}$ for the primal and $\mathbf{Y}^1_{[11]} = \sum\limits_{i=1}^{m}\frac{\alpha}{n_i}\boldsymbol{\mathcal{I}}_i$, $\mathbf{Y}^1_{[22]} = \mathbf{I}$, $\mathbf{Y}^1_{[12]} = 0$, $\mathbf{Y}^2 = \mathbf{I}$ for the dual problem, and therefore strong duality holds, and the interior point method can be utilized.

In order to solve this problem via interior point method, we need to formulate linear operator B and its adjoint B'. To do so, we will first define three index sets $J_1 = \{1, \ldots, m\}$, $J_2 = \{m+1, \ldots, m+\tilde{q}r\}$ and $J_3 = \{m+\tilde{q}r+1, \ldots, m+\tilde{q}r+r^2\}$. The linear operator can then be written as

$$B(Y) = B(\mathbf{Y}^1, \mathbf{Y}^2) = [B_1(\mathbf{Y}^1, \mathbf{Y}^2), \ldots, B_{m+\tilde{q}r+r^2}(\mathbf{Y}^1, \mathbf{Y}^2)]$$

where $B_i(\mathbf{Y}^1, \mathbf{Y}^2) = Tr(\mathbf{B}_i^1 \mathbf{Y}^1) + Tr(\mathbf{B}_i^2 \mathbf{Y}^2)$ such that

$$\mathbf{B}_i^1 = \begin{cases} \tilde{\boldsymbol{\mathcal{I}}}_i & i \in J_1 \\ \tilde{\boldsymbol{\mathcal{C}}}^{1kl} & i \in J_2, l = \lfloor \frac{i-m-1}{\tilde{q}} \rfloor + 1, k = i - m - (l-1)\tilde{q} \\ 0 & i \in J_3 \end{cases} . \tag{29}$$

$$\mathbf{B}_i^2 = \begin{cases} 0 & i \in J_1 \\ \tilde{\boldsymbol{\mathcal{C}}}^{2kl} & i \in J_2, l = \lfloor \frac{i-m-1}{\tilde{q}} \rfloor + 1, k = i - m - (l-1)\tilde{q} \\ \underline{\boldsymbol{\mathcal{I}}}^{kl} & i \in J_3, l = \lfloor \frac{i-m-\tilde{q}r-1}{r} \rfloor + 1, k = i - m - (\tilde{q}+l-1)r \end{cases} . \tag{30}$$

The adjoint operator is $B'(\mathbf{K}, \boldsymbol{\mathcal{U}}, \boldsymbol{\mu}) \equiv \boldsymbol{\mathcal{Z}}$ where $\boldsymbol{\mathcal{Z}}$ is defined above.

According to algorithm 2 the most computationally expensive stage in the interior point algorithm consist of constructing matrix $\boldsymbol{\mathcal{P}}$ and inverting it, since all other calculation take at most $O((\bar{n}_1 + \bar{n}_2 + 2r)^3) = O(n+2r)^3$ operations. We will show that this construction also takes $O(n)^3$ operations.

We can separate the constructing of matrix $\boldsymbol{\mathcal{P}}$ to several parts. For this purpose we will first denote $\mathbf{W} = \boldsymbol{\mathcal{Z}}^{-1}$ as a block diagonal matrix with blocks $\mathbf{W}^1 = \boldsymbol{\mathcal{Z}}^{1^{-1}}$ and $\mathbf{W}^2 = \boldsymbol{\mathcal{Z}}^{2^{-1}}$. An important observation is that the construction of matrix $\boldsymbol{\mathcal{P}}$ can be separated into two independent calculations:

$$\mathcal{P}_{ij} = \mathbf{W} \circ \mathbf{B}_j \mathbf{Y} \mathbf{B}_i = \mathbf{W}^1 \circ \mathbf{B}_j^1 \mathbf{Y}^1 \mathbf{B}_i^1 + \mathbf{W}^2 \circ \mathbf{B}_j^2 \mathbf{Y}^2 \mathbf{B}_i^2 = \mathbf{W} \circ \mathbf{B}_i \mathbf{Y} \mathbf{B}_j = \mathcal{P}_{ji}.$$

The first term of the calculation is identical to the terms calculated in appendix A (or is equal to zero when either i or j are in $J3$) and therefore costs $O(\bar{n}_1^3)$ to calculate for all i,j. Consequently, we will focus on the computational complexity of the second term:

1. $\forall i \in J_1$ or $j \in J_1$. The term is identically zero therefore doesn't need to be computed.
2. $\forall i, j \in J_2$. We start by calculating $\mathbf{B}_i^2 \mathbf{Y}^2 \mathbf{B}_j^2$. Denoting, as before, $l_1 = \lfloor \frac{i-m-1}{\tilde{q}} \rfloor + 1,\ k_1 = i - m - l_1\tilde{q},\ l_2 = \lfloor \frac{j-m-1}{\tilde{q}} \rfloor + 1,\ k_2 = j - m - l_2\tilde{q}$ We have that:

   $$\tilde{\boldsymbol{\mathcal{C}}}^{2ij} \mathbf{Y}^2 = \begin{bmatrix} \tilde{\mathbf{C}}^{2k_1l_1} \mathbf{Y}^2_{[21]} & \tilde{\mathbf{C}}^{2k_1l_1} \mathbf{Y}^2_{[22]} \\ (\tilde{\mathbf{C}}^{2k_1l_1})' \mathbf{Y}^2_{[11]} & (\tilde{\mathbf{C}}^{2k_1l_1})' \mathbf{Y}^2_{[12]} \end{bmatrix}$$

   and consequently

   $$\tilde{\boldsymbol{\mathcal{C}}}^{2k_1l_1} \mathbf{Y}^2 \tilde{\boldsymbol{\mathcal{C}}}^{2k_2l_2} = \begin{bmatrix} \tilde{\mathbf{C}}^{2k_1l_1} \mathbf{Y}^2_{[22]} (\tilde{\mathbf{C}}^{2k_2l_2})' & \tilde{\mathbf{C}}^{2k_1l_1} \mathbf{Y}^2_{[21]} \tilde{\mathbf{C}}^{2k_2l_2} \\ (\tilde{\mathbf{C}}^{2k_1l_1})' \mathbf{Y}^2_{[12]} (\tilde{\mathbf{C}}^{2k_2l_2})' & (\tilde{\mathbf{C}}^{2k_1l_1})' \mathbf{Y}^2_{[11]} \tilde{\mathbf{C}}^{2k_2l_2} \end{bmatrix}$$

   $\tilde{\mathbf{C}}^{2k_1l_1} \mathbf{Y}^2_{[22]} (\tilde{C}^{2k_2l_2})'$ has only one component which is not zero in index (l_1, l_2) and its value is $C^2_{k_1\cdot} \mathbf{Y}^2_{[22]} (C^2_{k_2\cdot})'$. On the other hand $(\tilde{\mathbf{C}}^{2k_1l_1})' \mathbf{Y}^2_{[11]} \tilde{C}^{2k_2l_2} = (\mathbf{Y}^2_{[11]})_{l_1l_2} (C^2_{k_1\cdot})' C^2_{k_2\cdot}$. Furthermore, only row l_1 of matrix $\tilde{\mathbf{C}}^{2k_1l_1} \mathbf{Y}^2_{[21]} \tilde{\mathbf{C}}^{2k_2l_2}$ is not zero, with value of $(\mathbf{C}^2_{k_1\cdot} (\mathbf{Y}^2_{[21]})_{\cdot l_2}) C^2_{k_2\cdot}$, and similarly only column l_2 of matrix $(\tilde{\mathbf{C}}^{2k_1l_1})'\mathbf{Y}^2_{[12]}(\tilde{\mathbf{C}}^{2k_2l_2})'$ is not zero, with value $(\mathbf{C}^2_{k_1\cdot})'(\mathbf{Y}^2_{[12]})_{l_2\cdot})(\mathbf{C}^2_{k_2\cdot})'$. The inner product is, therefore, given by:

$$\begin{aligned}\mathbf{W}^2 \circ \tilde{\boldsymbol{\mathcal{C}}}^{2k_1l_1}\mathbf{Y}^2\tilde{\boldsymbol{\mathcal{C}}}^{2k_2l_2} &= (W^2_{[11]})_{l_1l_2}\mathbf{C}^2_{k_1\cdot}\mathbf{Y}^2_{[22]}(\mathbf{C}^2_{k_2\cdot})' + (Y^2_{[11]})_{l_1l_2}\mathbf{C}^2_{k_2\cdot}\mathbf{W}^2_{[22]}(C^2_{k_1\cdot})' \\ &+ (\mathbf{C}^2_{k_1\cdot}(\mathbf{Y}^2_{[12]})'_{l_2\cdot})(\mathbf{C}^2_{k_2\cdot}(\mathbf{W}^2_{[12]})'_{l_1\cdot}) + ((\mathbf{W}^2_{[12]})_{l_1\cdot}(\mathbf{C}^2_{k_1\cdot})')((\mathbf{Y}^2_{[12]})_{l_2\cdot}(\mathbf{C}^2_{k_2\cdot})')\end{aligned}$$

Calculating the first term for all k_1 takes $O(\bar{n_2}^2\tilde{q})$ operations resulting in $2\tilde{q}$ vectors of dimension $\bar{n_2}$. Multiplying these vectors with $\mathbf{C}^2_{k_1\cdot}$ for all values of k_1 takes $O(\tilde{q}^2\bar{n_2})$ operations. The last two terms can also be calculated in two stages, first constructing $2\tilde{q}r$ scalars for each combination of k and l by multiplying the two appropriate vectors, at total cost of $O(r\tilde{q}\bar{n}_2)$, and then multiplying all combinations of these scalars costing $O(\tilde{q}^2r^2)$ operations. Therefore the total computational cost for this sub-matrix is therefore $O(\tilde{q}\bar{n}_2^2 + \bar{n}_2\tilde{q}^2 + r^2\tilde{q}^2 + r\tilde{q}\bar{n}_2)$. Since $\bar{n}_2 \leq n$ and since $O(\tilde{q}) = O(n)$ we have that this is equivalent to $O(n^3)$

3. $\forall i \in J_2\, j \in J_3$. Denoting, as before, $l_1 = \lfloor\frac{i-m-1}{\tilde{q}}\rfloor+1,\ k_1 = i-m-(l_1-1)\tilde{q},\ l_2 = \lfloor\frac{j-m-\tilde{q}r-1}{r}\rfloor+1,\ k_2 = j-m-(\tilde{q}+l_2-1)r$, to obtain $l_1,\ l_2,\ k_2 \in [r]$ and $k_1 \in [\tilde{q}]$. Notice that only column l_2 of matrix $\mathbf{Y}^2\underline{\boldsymbol{\mathcal{I}}}^{k_2l_2}$ is non zero, with value $\left[(\mathbf{Y}^2_{[11]})_{\cdot k_2}; (\mathbf{Y}^2_{[21]})_{\cdot k_2}\right]$ and therefore only the l_2 column of matrix $\tilde{\boldsymbol{\mathcal{C}}}^{2k_1l_1}\mathbf{Y}^2\underline{\boldsymbol{\mathcal{I}}}^{k_2l_2}$ is non zero. Consequently,

$$\mathbf{W} \circ \tilde{\boldsymbol{\mathcal{C}}}^{2k_1l_1}\mathbf{Y}^2\underline{\boldsymbol{\mathcal{I}}}^{k_2l_2} = (W^2_{[11]})_{l_1l_2}\mathbf{C}^2_{k_1\cdot}(\mathbf{Y}^2_{[21]})_{\cdot k_2} + \mathbf{C}^2_{k_1\cdot}(\mathbf{W}^2_{[21]})_{\cdot l_2}(Y^2_{[11]})_{l_1k_2}.$$

   Similarly to the previous case, assuming $O(\tilde{q}) = O(n)$, this calculation can be done for all such indexes in $O(r\bar{n}_2\tilde{q} + r^3\tilde{q}) = O(n^2)$ operations.

4. $\forall i, j \in J_3$. In this case we denote $l_1 = \lfloor\frac{i-m-\tilde{q}r-1}{r}\rfloor+1,\ k_1 = i-m-(\tilde{q}+l_1-1)r,\ l_2 = \lfloor\frac{j-m-\tilde{q}r-1}{r}\rfloor+1,\ k_2 = j-m-(\tilde{q}+l_2-1)r$. According to the previous case we can conclude that:

$$\mathbf{W} \circ \underline{\boldsymbol{\mathcal{I}}}^{k_1l_1}\mathbf{Y}^2\underline{\boldsymbol{\mathcal{I}}}^{k_2l_2} = (W^2_{[11]})_{l_2k_1}(Y^2_{[11]})_{l_1k_2}.$$

   Obviously computing this for all relevant indexes takes $O(r^4)$.

Therefore, computing $\boldsymbol{\mathcal{P}}$ in this case takes $O(n^3)$ operations.

Consequently, the interior point algorithm 2 implemented on problem (α-SDP-EP) takes $O(n^3)$ per iteration and obtains an ϵ accurate solution in $O(\sqrt{n+r}\log(\frac{1}{\epsilon}))$ iterations.

Appendix C Obtaining a solution to problem (SVE)

Given problem (SVE) we define additional variable $\boldsymbol{\mathcal{Z}}$ as:

$$\boldsymbol{\mathcal{Z}}^1 = \begin{bmatrix}\sum_{i=1}^m \mu_i\boldsymbol{\mathcal{I}}_i & \tilde{\mathbf{E}}' \\ \tilde{\mathbf{E}} & \boldsymbol{\Sigma}\end{bmatrix},\ \mathcal{Z}^2 = 1 - \sum_{i=1}^m \mu_i,\ \boldsymbol{\mathcal{Z}} = \begin{bmatrix}\boldsymbol{\mathcal{Z}}^1 & \mathbf{0} \\ \mathbf{0} & \mathcal{Z}^2\end{bmatrix},\ \boldsymbol{\mathcal{Z}} \succeq 0,$$

Since $\boldsymbol{\Sigma}$ is defined to be symmetric it is represented by $r(r+1)/2$ variables. Constructing the problem dual via its Lagrangian.

The dual problem is given by:

$$\begin{aligned}
&\max_{\mathbf{Y}\in\mathbb{R}^{(n+r)\times(n+r)},\beta\in\mathbb{R}_+} \mathbf{Y}^1 \circ \begin{bmatrix} \mathbf{0} & -\tilde{\mathbf{E}}' \\ -\tilde{\mathbf{E}} & \mathbf{0}\end{bmatrix} - \beta = -2\mathbf{Y}^{12} \circ \tilde{\mathbf{E}}' - \beta \\
&s.t. \\
&\quad \boldsymbol{\mathcal{I}}_i \circ \mathbf{Y}^{11} - \beta = 0 \qquad\qquad i = 1, \ldots, m \\
&\quad \mathbf{Y}^{22} = \mathbf{I} \\
&\quad \mathbf{Y} = \begin{bmatrix} \mathbf{Y}^{11} & \mathbf{Y}^{12} \\ \mathbf{Y}^{12\prime} & \mathbf{Y}^{22}\end{bmatrix} \succeq 0
\end{aligned} \tag{31}$$

Both primal and dual problem are strictly feasible with $\boldsymbol{\mu} = \mathbf{e}/n$ and $\boldsymbol{\Sigma} = \tilde{\mathbf{E}}'(\sum_i \mu_i \boldsymbol{\mathcal{I}}_i)^{-1}\tilde{\mathbf{E}} + \mathbf{I}$ for the primal and $\mathbf{Y}^{11} = \sum_i \boldsymbol{\mathcal{I}}_i/n_i$, $\mathbf{Y}^{22} = \mathbf{I}$, $\mathbf{Y}^{12} = \mathbf{0}$, $\beta = 1$ for the dual problem. Therefore, strong duality holds for these problems.

Observe that since $\mathbf{Y}^{22} = \mathbf{I}$ the PSD constraint on $\mathbf{Y}$ can be reformulated, using the Schur complement, as $\boldsymbol{\mathcal{Y}} = (\mathbf{Y}^{11} - \mathbf{Y}^{12}\mathbf{Y}^{12\prime}) \succeq 0$. Moreover, any square sub-matrix on the diagonal of $\boldsymbol{\mathcal{Y}}$ is also PSD and therefore the first constraints can be reformulated as:

$$\mathrm{Tr}(\mathbf{Y}^{12\prime}\boldsymbol{\mathcal{I}}_i\mathbf{Y}^{12}) = \mathbf{Y}^{12}\mathbf{Y}^{12\prime} \circ \boldsymbol{\mathcal{I}}_i \leq \beta.$$

Defining k^i_j $j = 1, \ldots, n_i$ to be the jth index in the index set corresponding to partition element i, we can define vector $\boldsymbol{\zeta}_i = [\zeta_i^1, \ldots, \zeta_i^{n_i}]'$ where $\boldsymbol{\zeta}_i^j = \mathbf{Y}^{12}_{k^i_j\cdot}$ is the k^i_j row of matrix $\mathbf{Y}^{12}$. Similarly, we define $\tilde{\mathbf{e}}_i = [\tilde{\mathbf{e}}_i^1; \ldots; \tilde{\mathbf{e}}_i^{n_i}]$ where $\tilde{\mathbf{e}}_i^j = \tilde{\mathbf{E}}_{\cdot k^i_j}$ is the k^i_j column of matrix $\tilde{\mathbf{E}}$. Therefor, problem (31) can be reformulates as follows.

$$\begin{aligned}
&\max_{\beta\geq 0}\max_{\boldsymbol{\zeta}} \quad -2\sum_{i=1}^m \boldsymbol{\zeta}_i'\tilde{\mathbf{e}}_i - \beta \\
&s.t. \qquad\qquad \|\boldsymbol{\zeta}_i\|^2 \leq \beta \quad i = 1, \ldots, m
\end{aligned} \tag{32}$$

Since $\|e_i\| = \|\boldsymbol{\mathcal{I}}_i\tilde{\mathbf{E}}\|_F$ where $\|\cdot\|_F$ denotes the Frobenius norm, the definition of J_0 can be updated to $J_0 = \{i \in \{1, \ldots, m\} : \|\tilde{\mathbf{e}}_i\| \neq 0\}$. Given known β the problem is separable in $\boldsymbol{\zeta}_i$, and an optimal solution (not necessarily unique) is known to be

$$\boldsymbol{\zeta}_i^* = \begin{cases} -\sqrt{\beta}\frac{\tilde{\mathbf{e}}_i}{\|\tilde{\mathbf{e}}_i\|} & i \in J_0 \\ \sqrt{\frac{\beta}{n_i}} & \text{otherwise} \end{cases}. \tag{33}$$

Plugging this solution we obtain the following problem in variable β.

$$\max_{\beta\geq 0} 2\sqrt{\beta}\sum_{i=1}^m \|\tilde{\mathbf{e}}_i\| - \beta.$$

Since $\tilde{\mathbf{E}} \neq 0$ the solution is obtained in a stationary point, i.e. when $\beta = 1/(\sum_{i=1}^m \|\tilde{\mathbf{e}}_i\|)^2$, in either case the optimal objective function value is $\sum_{i=1}^m \|\tilde{\mathbf{e}}_i\|$. Notice that if $\tilde{\mathbf{E}} = 0$ the problem becomes trivial with solution $\boldsymbol{\Sigma} = 0$

In order to retrieve the solution to the primal problem, we first look at the dual of problem (32), which is given by:

$$\begin{aligned} \min_{\mu \in \mathbb{R}^m_+} \quad & \sum_{i \in J_0} \frac{\|\tilde{\mathbf{e}}_i\|^2}{\mu_i} \\ s.t. \quad & \sum_{i=1}^{m} \mu_i = 1 \\ & \mu_i = 0 \quad i \notin J_0. \end{aligned} \tag{34}$$

The optimal solution to (34) is given, by the primal dual connection $\tilde{\mathbf{e}}_i = \boldsymbol{\zeta}_i \mu_i$ and the optimal solution (33), to be

$$\mu_i^{\diamond} = \frac{\|\tilde{\mathbf{e}}_i\|}{\sqrt{\beta}} = \frac{\|\tilde{\mathbf{e}}_i\|}{\sum_i \|\tilde{\mathbf{e}}_i\|} \quad i = 1, \ldots, m.$$

Returning to problem (SVE) we obtain $\boldsymbol{\Sigma} \succeq \tilde{\mathbf{E}}'(\sum_i \mu_i \boldsymbol{\mathcal{I}}_i)^{-1}\tilde{\mathbf{E}}$ provided $\mu_i = 0 \ \forall i \notin J_0$, and so $\mathrm{Tr}(\boldsymbol{\Sigma}) \geq \sum_{i \in J_0} \|\tilde{\mathbf{e}}_i\|^2 / \mu_i$. Therefore, the optimal $\boldsymbol{\Sigma}$ for problem (SVE) is obtained when the inequality is satisfied with equality, and is given by:

$$\boldsymbol{\Sigma}^{\diamond} = \left(\sum_{i=1}^{m} \|\boldsymbol{\mathcal{I}}_i \tilde{\mathbf{E}}\|_F \right) \tilde{\mathbf{E}}' \left(\sum_{i \in J_0} \frac{1}{\|\boldsymbol{\mathcal{I}}_i \tilde{\mathbf{E}}\|_F} \boldsymbol{\mathcal{I}}_i \right) \tilde{\mathbf{E}}$$

with optimal value of $\mathrm{Tr}(\boldsymbol{\Sigma}^{\diamond}) = (\sum_{i=1}^{m} \|\boldsymbol{\mathcal{I}}_i \tilde{\mathbf{E}}\|_F)^2$. The most computationally expensive operation is the multiplication $\tilde{\mathbf{E}}'\tilde{\mathbf{E}}$ which costs $O(n^2 r^2)$.

In conclusion, solving problem (SVE), including obtaining the optimal $\boldsymbol{\Sigma}^{\diamond}$ takes at most $O(n^2 r^2)$ operations. Given r is constant and $n \leq S \max_i n_i$ where $\max_i n_i$ is constant, we obtain computational complexity of $O(S^2)$.

Appendix D Rolling horizon objective function coefficients

Representing

$$\begin{aligned} \boldsymbol{\delta}_T &= \mathbf{A}_T^{\boldsymbol{\omega}} \boldsymbol{\omega}_T^{T-S} + \mathbf{A}_T^{\mathbf{Z}} \mathbf{Z}_{T-S}^{T-2S+2} \\ &\equiv \mathbf{A}^{\boldsymbol{\delta}_{T-S}} \boldsymbol{\delta}_{T-S} + \sum_{t=T-S+1}^{T} \mathbf{A}_T^{\mathbf{a}_t} \mathbf{a}_t + \sum_{t=T-S+1}^{T} \mathbf{A}_T^{\mathbf{v}_t} \mathbf{v}_t + \sum_{t=T-2S+2}^{T-S} \mathbf{A}_T^{\mathbf{z}_t} \mathbf{z}_t, \end{aligned}$$

where the coefficient matrices are given by:

$$\mathbf{A}_T^{\mathbf{a}_t} = \begin{cases} \mathbf{F}\mathbf{A}_{T-1}^{\mathbf{a}_t} - \mathbf{K}_T^T \mathbf{H}\mathbf{F}\mathbf{A}_{T-1}^{\mathbf{a}_t} + \mathbf{K}_T^t \mathbf{H}\mathbf{G} - \sum\limits_{\tau=t+1}^{T-1} \mathbf{K}_T^{\tau} \mathbf{H}\mathbf{F}\mathbf{A}_{\tau-1}^{\mathbf{a}_t} & \max(T-S+1, 1) \leq t < T \\ (\mathbf{K}_T^T \mathbf{H}\mathbf{G} - \mathbf{G}) & t = T, \ T \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{A}_T^{\mathbf{v}_t} = \begin{cases} \mathbf{F}\mathbf{A}_{T-1}^{\mathbf{v}_t} - \mathbf{K}_T^T \mathbf{H}\mathbf{F}\mathbf{A}_{T-1}^{\mathbf{v}_t} + \mathbf{K}_T^t - \sum\limits_{\tau=t+1}^{T-1} \mathbf{K}_T^{\tau} \mathbf{H}\mathbf{F}\mathbf{A}_{\tau-1}^{\mathbf{v}_t} & \max(T-S+1, 1) \leq t < T \\ \mathbf{K}_T^T & t = T, \ T \geq 1 \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\mathbf{A}_T^{\boldsymbol{\delta}_t} = \begin{cases} \mathbf{F}\mathbf{A}_{T-1}^{\boldsymbol{\delta}_t} - \sum\limits_{\tau=t+2}^{T} \mathbf{K}_T^{\tau}\mathbf{H}\mathbf{F}\mathbf{A}_{\tau-1}^{\boldsymbol{\delta}_t} & \max(T-S,0) \leq t < T-1 \\ \mathbf{F} - \mathbf{K}_T^{T}\mathbf{H}\mathbf{F} & t = T-1 \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\mathbf{L}_T^{\mathbf{z}_t,j} = \begin{cases} \mathbf{K}_T^{t} - \sum\limits_{\tau=t+j}^{T} \mathbf{H}\mathbf{F}\mathbf{L}_{\tau-1}^{\mathbf{z}_t,j} & max(T-S+1,1) \leq t \leq T-1,\ 1 \leq j \leq T-t \\ \mathbf{F}\mathbf{L}_{T-1}^{\mathbf{z}_t,j} - \sum\limits_{\tau=max(T-S,t+j)}^{T} \mathbf{H}\mathbf{F}\mathbf{L}_{\tau-1}^{\mathbf{z}_t,j} & max(T-2S+2,1) \leq t \leq T-S,\ 1 \leq j \leq S-1 \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\mathbf{A}_T^{\mathbf{z}_t} = \begin{cases} \mathbf{F}\mathbf{L}_{T}^{\mathbf{z}_t,j} - \sum\limits_{\tau=T-S+1}^{T} \mathbf{H}\mathbf{F}\mathbf{L}_{\tau-1}^{\mathbf{z}_t,j} & \max(T-2S+2,1) \leq t \leq T-S,\ j = T-S-t+1 \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

Similarly we can represent

$$\mathbf{z}_T = \hat{\mathbf{A}}_T^{\boldsymbol{\omega}}\boldsymbol{\omega}_T^{T-S} + \hat{\mathbf{A}}_T^{\mathbf{Z}}\mathbf{Z}_{T-S}^{T-2S+2} \equiv \hat{\mathbf{A}}^{\boldsymbol{\delta}_{T-S}}\boldsymbol{\delta}_{T-S} + \sum_{t=T-S+1}^{T} \hat{\mathbf{A}}_T^{\mathbf{a}_t}\mathbf{a}_t + \hat{\mathbf{A}}_T^{\mathbf{v}_t}\mathbf{v}_t$$

where:

$$\hat{\mathbf{A}}_T^{\mathbf{a}_t} = \begin{cases} -\mathbf{H}\mathbf{F}\mathbf{A}_{T-1}^{\mathbf{a}_t} & \max(T-S+1,1) \leq t < T \\ \mathbf{H}\mathbf{G} & t = T,\ T \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{\mathbf{A}}_T^{\mathbf{v}_t} = \begin{cases} -\mathbf{H}\mathbf{F}\mathbf{A}_{T-1}^{\mathbf{v}_t} & \max(T-S+1,1) \leq t < T \\ \mathbf{I} & t = T,\ T \geq 1 \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\hat{\mathbf{A}}_T^{\boldsymbol{\delta}_t} = \begin{cases} -\mathbf{H}\mathbf{F}\mathbf{A}_{T-1}^{\boldsymbol{\delta}_t} & \max(T-S,0) \leq t < T-1 \\ -\mathbf{H}\mathbf{F} & t = T-1,\ T \geq 1 \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\hat{\mathbf{A}}_T^{\mathbf{z}_t} = \begin{cases} -\mathbf{H}\mathbf{F}\mathbf{A}_{T-1}^{\mathbf{z}_t} & \max(T-2S+2,1) \leq t \leq T-S \\ \mathbf{0} & \text{otherwise} \end{cases}$$

References

1. Bar-Shalom, Y., Li, X.R.: Multitarget-Multisensor tracking : principles and techniques. YBS Publishing, Storrs, CT, USA (1995)
2. Bar-Shalom, Y., Li, X.R., Kirubarajan, T.: Estimation with applications to tracking and navigation. Wiley, New York, NY, USA (2001)
3. Başar, T., Bernhard, P.: H-Infinity Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach (Second Edition). Modern Birkhäuser Classics. Birkhäuser, Boston, MA, USA (1995)
4. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging. Sci. **2**(1), 183–202 (2009)
5. Ben-Tal, A., El Ghaoui, L., Nemirovski, A.: Robust optimization Princeton University Press, Princeton, NJ, USA (2009)
6. Bertsekas, D., Rhodes, I.: Recursive state estimation for a set-membership description of uncertainty. IEEE Trans. Autom. Control **16**(2), 117–128 (1971)

7. Dong, Z., You, Z.: Finite-horizon robust kalman filtering for uncertain discrete time-varying systems with uncertain-covariance white noises. Signal Processing Letters, IEEE **13**(8), 493–496 (2006)
8. El Ghaoui, L., Calafiore, G.: Robust filtering for discrete-time systems with bounded noise and parametric uncertainty. IEEE Trans. Autom. Control **46**(7), 1084–1089 (2001)
9. Fu, M., de Souza, C.E., Luo, Z.Q.: Finite-horizon robust kalman filter design. IEEE Trans. Signal Process. **49**(9), 2103–2112 (2001)
10. Garulli, A., Vicino, A., Zappa, G.: Conditional central algorithms for worst case set-membership identification and filtering. IEEE Trans. Autom. Control **45**(1), 14–23 (2000)
11. Hazan, E.: Sparse approximate solutions to semidefinite programs. In: E. Laber, C. Bornstein, L. Nogueira, L. Faria (eds.) LATIN 2008: Theoretical Informatics, *Lecture Notes in Computer Science*, vol. 4957, pp. 306–316. Springer Berlin Heidelberg (2008)
12. Heffes, H.: The effect of erroneous models on the kalman filter response. IEEE Trans. Autom. Control **11**(3), 541–543 (1966)
13. Helmberg, C., Rendl, F., Vanderbei, R.J., Wolkowicz, H.: An interior-point method for semidefinite programming. SIAM J. Optim. **6**, 342–361 (1996)
14. Kalman, R.E.: A new approach to linear filtering and prediction problems. J. Basic Eng-T. ASME **82**(Series D), 35–45 (1960)
15. Kurzhanski, A.B., Vályi, I.: Ellipsoidal calculus for estimation and control. Systems and control : foundations and applications. Birkhäuser, Boston, MA, USA (1997)
16. Masreliez, C.: Approximate non-Gaussian filtering with linear state and observation relations. IEEE Trans. Autom. Control **20**(1), 107–110 (1975)
17. Masreliez, C., Martin, R.: Robust bayesian estimation for the linear model and robustifying the kalman filter. IEEE Trans. Autom. Control **22**(3), 361–371 (1977)
18. Nagpal, K.M., Khargonekar, P.P.: Filtering and smoothing in an H infinity setting. IEEE Trans. Autom. Control **36**(2), 152–166 (1991)
19. Navarro, J.: A very simple proof of the multivariate chebyshev's inequality. Communications in Statistics - Theory and Methods (2013). DOI 10.1080/03610926.2013.873135
20. Nemirovski, A.: Prox-method with rate of convergence o(1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. SIAM J. Optim. **15**, 229–251 (2004)
21. Nesterov, Y.: Semidefinite relaxation and nonconvex quadratic optimization. Optim. Methods Softw. **9**(1-3), 141–160 (1998)
22. Nesterov, Y.: Dual extrapolation and its applications to solving variational inequalities and related problems. Math. Programming **109**(2-3), 319–344 (2007)
23. Nishimura, T.: On the a priori information in sequential estimation problems. IEEE Trans. Autom. Control **11**(2), 197–204 (1966)
24. Sasa, S.: Robustness of a kalman filter against uncertainties of noise covariances. In: American Control Conference, 1998. Proceedings of the 1998, vol. 4, pp. 2344–2348 vol.4 (1998)
25. Sayed, A.: A framework for state-space estimation with uncertain models. IEEE Trans. Autom. Control **46**(7), 998–1013 (2001)
26. Schweppe, F.: Recursive state estimation: Unknown but bounded errors and system inputs. IEEE Trans. Autom. Control **13**(1), 22–28 (1968)
27. Shaked, U., Theodor, Y.: $\mathcal{H}_\infty$-optimal estimation: a tutorial. In: Decision and Control, 1992., Proceedings of the 31st IEEE Conference on, vol. 2, pp. 2278 –2286 (1992)
28. Shtern, S., Ben-Tal, A.: Algorithms for solving nonconvex block constrained quadratic problems. Tech. Rep. Technion Report IE/OR-2013-01, Technion - Israel Instutute of Technology (2013)
29. Simon, D.: Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches. Wiley-Interscience, Hoboken, NJ, USA (2006)
30. Tsai, C., Kurz, L.: An adaptive robustizing approach to kalman filtering. Automatica **19**(3), 279–288 (1983)
31. Willems, J.C.: Deterministic least squares filtering. J. Econometrics **118**(1), 341–373 (2004)
32. Xie, L., Soh, Y.C., de Souza, C.E.: Robust kalman filtering for uncertain discrete-time systems. IEEE Trans. Autom. Control **39**(6), 1310–1314 (1994)
33. Yang, F., Wang, Z., Hung, Y.S.: Robust kalman filtering for discrete time-varying uncertain systems with multiplicative noises. IEEE Trans. Autom. Control **47**(7), 1179–1183 (2002)