Bayesian Inference in Linear Models With a Random Gaussian Matrix: Algorithms and Complexity

Ido Nevat *, Gareth W. Peters† and Jinhong Yuan*

* School of Electrical Engineering and Telecommunications, University of NSW, Sydney, Australia
† School of Mathematics and Statistics, University of NSW and CSIRO Mathematical and Information Sciences, Sydney, Australia
Email: ido@student.unsw.edu.au, peterga@maths.unsw.edu.au, J.Yuan@unsw.edu.au

Abstract—We consider the Bayesian inference of a random Gaussian vector in a linear model with a random Gaussian matrix. We review two approaches to finding the MAP estimator for this model. We propose improved versions of these approaches with reduced complexity. Next we analyze their complexity and convergence properties. Then we derive the MAP estimator in the setting in which the variance of the noise is unknown. Simulation results presented compare the performance in terms of estimation error of the approaches.

I. INTRODUCTION

A generic problem in many different fields is the estimation of a random Gaussian vector \( x \) in the linear model

\[
y = Gx + w, \tag{1}
\]

where \( G \) is a linear transformation matrix and \( w \) is a Gaussian noise vector. This estimation problem arises in a large variety of areas in science and engineering e.g., communication, economics, signal processing, seismology, and control. Three standard methods for estimating \( x \) in this Bayesian framework are the minimum mean squared error (MMSE), the linear minimum mean squared error (LMMSE) and the maximum a-posteriori (MAP) estimators. The first two approaches are based on a quadratic cost function whereas the third minimizes a hit-or-miss risk. From a detection point of view, the MAP based on a quadratic cost function whereas the third minimizes the minimum mean squared error (LMMSE) and the maximum error of the approaches.

Most of the literature concentrates on the simplest case, in which it is assumed that the model matrix \( G \) is completely specified. In this setting, the MMSE, LMMSE and MAP estimators coincide and have a simple closed form solution. However, in our model, the matrix \( G \) is a random matrix with independent and identically distributed elements and known second-order statistics. In many applications the model matrix \( G \) is subject to uncertainties. For example, \( G \) may be estimated from noisy data. If the actual data matrix deviates from the one assumed, then the performance of an estimator designed based on the assumed model alone may deteriorate considerably.

This paper is a continuation of [1] and [2]. We concentrate on the evaluation of the MAP estimator for model (1) using two different methods. First we review an optimal solution using the hidden convexity (HC) property of the underlying model [1]. Using the method developed in [3], we transform the multi-dimensional, nonlinear and nonconvex problem into a simple tractable form which can be efficiently evaluated. We present a more efficient method to solve the problem and assess the complexity involved. Next we review a suboptimal solution using the Bayesian Expectation-Maximization (BEM) methodology [2] which is an extension of the one suggested in [4]. We suggest an efficient implementation which does not require matrix inversion and is therefore appealing to real time applications. Additionally, we assess the sensitivity of the BEM to the initial conditions of the algorithm and analyse how it affects the number of iterations required in order for it to converge.

The main contributions of this paper are:

1) We provide efficient solutions to MAP estimation in model (1) and analyze their complexity.
2) We compare the performance of the HC and the BEM solutions, in terms of MSE.
3) We extend the estimation problem to the case in which the noise variance is unknown.

The following notation is used. Boldface upper case letters denote matrices, boldface lower case letters denote column vectors, and standard lower case letters denote scalars. The superscripts \((\cdot)^T\) denotes the transpose operation. By \( I \) we denote the identity matrix. \(||\cdot||\) is the standard Euclidean norm. \( \lambda_{\text{min}}(X) \) is the smallest eigenvalue of \( X \). The functions \( p(x) \), \( p(x|y) \) and \( E\{\cdot\} \) denote the probability distribution function (PDF) of \( x \), the PDF of \( x \) given \( y \), and the expectation, respectively. \( y[i] \) denotes the \( i^{th} \) element of the vector \( y \).

II. PROBLEM FORMULATION

Consider the problem of estimating a random vector \( x \) in the linear model

\[
y = Gx + w, \tag{2}
\]

where \( x \) is a zero-mean Gaussian vector with independent elements of variance \( \sigma_x^2 > 0 \) and \( w \) is a zero-mean Gaussian vector with independent elements of variance \( \sigma_w^2 > 0 \). The matrix \( G \) is an \( N \times K \) that can be decomposed as

\[
G = H + \Sigma, \tag{3}
\]
where $H$ is a deterministic and known $N \times K$ matrix, and $\Sigma$ is an $N \times K$ matrix with i.i.d random Gaussian zero-mean elements and variance of each element $\sigma_d^2 > 0$. In addition, $x$, $G$ and $w$ are statistically independent. It is desired to find an estimator $\hat{x}(y)$ which is a function of the observation vector $y$ and the given statistics of $G$, that is optimal in some sense. Under the Bayesian framework, a typical procedure for selecting $\hat{x}(y)$ is to define a nonnegative cost function $C(x, \hat{x}(y))$ and to minimize its risk $R$, defined as the expected cost [5].

$$R = E \{ C(x, \hat{x}) \} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} C(x, \hat{x}) p_{xy}(x, y) \, dx \, dy = \int_{-\infty}^{\infty} I(\hat{x}) p_{y}(y) \, dy,$$

where

$$I(\hat{x}) = \int_{-\infty}^{\infty} C(x, \hat{x}) p_{xy}(x|y) \, dx. \quad (5)$$

Since (4) is the expectation of a positive quantity $I(\hat{x})$, it is sufficient to minimize (5). The most common objective function is the quadratic error which is defined as

$$C(x, \hat{x}(y)) = \| x - \hat{x}(y) \|_2^2. \quad (6)$$

Minimizing this objective function leads to the well known MMSE estimator [5]

$$\hat{x}_{MMSE}(y) = E \{ x|y \} = E \{ E \{ x|y, G \} | y \} = E \left\{ \left( G^T G + \frac{\sigma_w^2}{\sigma_x^2} I \right)^{-1} G^T y \right\}. \quad (7)$$

Unfortunately, it is easy to see that the computational complexity involved in solving (7) is too high for practical applications. Instead, a common approach is to consider the LMMSE estimator which satisfies the following closed form solution

$$\hat{x}_{LMMSE}(y) = E \{ x y^T \} E^{-1} \{ y y^T \} y$$

$$= H^T \left( HH^T + K \sigma_g^2 I + \frac{\sigma_w^2}{\sigma_x^2} I \right)^{-1} H^T y, \quad (8)$$

where we have used the fact that $x$ and $y$ are zero mean random vectors. Using the Lemma of Matrix inversion a more convenient form of (8) can be written as

$$\hat{x}_{LMMSE}(y) = \left( H^T H + \left( K \sigma_g^2 + \frac{\sigma_w^2}{\sigma_x^2} I \right) \right)^{-1} H^T y. \quad (9)$$

Alternatively, one may choose to minimize the hit-or-miss cost function given by

$$C(x, \hat{x}(y)) = \begin{cases} 0, & \| x - \hat{x}(y) \|_2 \leq \epsilon, \\ 1, & \text{otherwise} \end{cases}, \quad (10)$$

where $\epsilon \to 0$ is a positive scalar. Optimizing this cost function yields the MAP estimator:

$$\hat{x}_{MAP}(y) = \arg \max_x \{ \log p_{x|y}(x|y) \}$$

$$= \arg \max_x \{ \log p_{y|x}(y|x) + \log p_x(x) \} \quad (11)$$

As a result of the Gaussian assumption, we have that

$$p(y|x) \sim N \left( Hx, \left( \sigma_g^2 \| x \|_2^2 + \sigma_w^2 \right) I \right), \quad (12)$$

and $\hat{x}_{MAP}(y)$ is the solution to

$$\min_x \left\{ \frac{\| y - Hx \|_2^2}{\sigma_g^2 \| x \|_2^2 + \sigma_w^2} + N \log(\sigma_g^2 \| x \|_2^2 + \sigma_w^2) + \frac{\| x \|_2^2}{\sigma_x^2} \right\}. \quad (13)$$

Problem (13) is a K-dimensional, nonlinear and nonconvex optimization program. In the rest of the paper we will discuss two different methods for solving (11) (and equivalently (13)) and analyze their algorithmic complexity.

### III. MAP Estimation Using Hidden Convexity Optimization

In this Section we first review the solution of (13). Next we provide a low complexity method of solving this problem. We have solved the optimization problem in (13) in [1] using the hidden convexity property of the underlined program.

**Theorem 1:** For any $t \geq 0$, let

$$f(t) = \min_{x} \| y - Hx \|_2^2 \quad s.t. \| x \|_2^2 = t \quad (14)$$

and denote the optimal argument by $x(t)$. Then, using a simple change of variables $\| x \|_2^2 = t$, the MAP estimator of $x$ in the model (13) is $x(t^*)$, where $t^*$ is the solution to the following unimodal optimization problem:

$$\arg \min_{t \geq 0} \left\{ \frac{f(t)}{\sigma_g^2 t + \sigma_w^2} + N \log(\sigma_g^2 t + \sigma_w^2) + \frac{t}{\sigma_x^2} \right\}. \quad (15)$$

Therefore, in order to find the solution to (13) two nested line searches are required. We have an outer minimization with respect to $t$ (eq. (15)) and for each fixed $t$ we need to solve (14). The change of variables in Theorem 1 allows for an efficient solution of the MAP problem. This is due to:

1. There are standard methods for evaluating $f(t)$ in (14) for any $t \geq 0$.
2. The unimodality of (15) (the full details can be found in [1]) ensures that an efficient one dimensional search can find the global optimum.

In the rest of this section, we discuss these two properties. We first provide a simple and efficient method for evaluating $f(t)$ in (14). Problem (14) is a quadratically constrained least squares (LS) problem whose solution can be traced back to [6]. Note that this has the same form as a ridge regression estimator.

**Lemma 1:** ([6], [7]): The solution to

$$f(t) = \min_{x: \| x \|_2^2 = t} \| y - Hx \|_2^2 \quad (16)$$

is

$$x(t) = (H^T H + \eta I)^{-1} H^T y, \quad (17)$$
where \( \eta \geq -\lambda_{\min}(H^T H) \) is the unique root of the equation
\[
\|x(t)\|^2 = t.
\] (18)

We now discuss the implementation of Lemma 1. The only issue is the evaluation of \( \eta \). This can be done by trying different values of \( \eta \) in (17) until one that satisfies (18) is found. This is made simpler due to the monotonicity of
\[
\left\| (H^T H + \eta I)^{-1} H^T y \right\|^2 \quad \text{in} \eta \quad \text{which enables us to find a value of} \quad \eta \quad \text{that satisfies} \quad (18) \quad \text{using} \quad \text{a simple line-search, such as bi-section} \quad \text{[9]}. \quad \text{The search range is} \quad -\lambda_{\min}(H^T H) \leq \eta \leq \eta_{\max}, \quad \text{where} \quad \eta_{\max} \quad \text{is some sufficiently large upper bound. Next,} \quad f(t) \quad \text{can be evaluated by plugging the appropriate} \quad x(t) \quad \text{into} \quad \|y - Hx(t)\|^2. \quad \text{However, this procedure may be computationally intensive due to the matrix inversion needed in each trial of} \quad \eta. \quad \text{We now suggest an efficient implementation of} \quad (H^T H + \eta I)^{-1} H^T \quad \text{in (17)}:
\[
\text{Algorithm 1: } \quad \text{The implementation of Lemma 1 is presented in Algorithm 1. Now that we have an efficient method for evaluating} \quad f(t), \quad \text{it remains to solve} \quad (15). \quad \text{The unimodality property ensures}
\]

\[
\text{Proof: } \text{See Appendix}
\]

The implementation of Lemma 1 is presented in Algorithm 1. Now that we have an efficient method for evaluating \( f(t) \),

**Algorithm 1** Constrained Least Squares (Lemma 1)

**Input:** \( t, H, y, \lambda_{\min}(H^T H), A, U, V, \eta_{\max} \)

**Output:** \( f(t), x \)

1: \( \eta_L = -\lambda_{\min}(H^T H) \)
2: \( \eta_R = \eta_{\max} \)
3: 
4: \( \eta_M = \frac{\eta_L + \eta_R}{2} \)
5: \( x = V \Lambda_M U^T y \)
6: \( \epsilon = x^T x - t \)
7: if \( \epsilon > 0 \) then
8: \( \eta_L = \eta_M \)
9: else
10: \( \eta_R = \eta_M \)
11: end if
12: until \( |\epsilon| \leq \epsilon_{\min} \)
13: \( f(t) = \|y - Hx\|^2 \)

\( \text{Algorithm 2 MAP Estimation - Solution of eq. (15)} \)

**Input:** \( y, H, \sigma_x^2, \sigma_w^2, \sigma_y^2, N, t_{\max} \)

**Output:** \( x \)

1: \( t_L = 0 \)
2: \( t_R = t_{\max} \)
3: \( \rho = (\sqrt{5} - 1)^2 \)
4: repeat
5: \( \Delta = t_R - t_L \)
6: \( t_A = t_L + \rho \Delta \)
7: \( t_B = t_R - \rho \Delta \)
8: \( r(t_A) = \frac{t_A}{\sigma_x^2 t_A + \sigma_w^2} + N \log(\sigma_x^2 t_A + \sigma_w^2) + \frac{t_A}{\sigma_y^2} \)
9: \( r(t_B) = \frac{t_B}{\sigma_x^2 t_B + \sigma_w^2} + N \log(\sigma_x^2 t_B + \sigma_w^2) + \frac{t_B}{\sigma_y^2} \)
10: if \( r(t_A) < r(t_B) \) then
11: \( t_R = t_B \)
12: else
13: \( t_L = t_A \)
14: end if
15: until \( |r(t_A) - r(t_B)| < \epsilon \)

**A. Complexity Analysis**

We now assess the complexity of the new algorithm. We define the cost of an algorithm as the total number of floating-point operations (flops) required to carry it out, as a function of problem dimensions [10]. The most intensive part is the SVD operation which needs to be implemented once. Its complexity is \( 4NK^2 + 8NK^2 + 9K^3 \) [flops] [10]. Since the outer line search (Algorithm 2) consists of only scalar operations and doesn’t depend on either \( x \) or \( y \), it is negligible. Next we assess the complexity of the inner line search (Algorithm 1). The vector operations are steps (5),(6) and (13). The complexity of step (5) is \( 2K^2N + 2K^2N \) [flops]. The complexity of step (6) is \( 2K \) [flops] and the complexity of step (13) is \( N(2K + 3) \) [flops]. The overall complexity of Algorithm 1 is
\[
C_{\text{INNER}} = 2(K^2N + 2K^2N + K + 2K + N) \quad \text{ITR} \quad N(2K + 3),
\]
where \( \text{ITR}_1 \) is the number of iterations of the inner line search. The overall complexity is
\[
C_{HD} = (SVD \text{ complexity}) + C_{\text{INNER}} \quad \text{ITR}_2, \quad (22)
\]

where \( \text{ITR}_2 \) is the number of iterations in the outer line search. If \( N \) and \( K \) are of the same order of magnitude, and ignoring all terms except the leading terms, we can simplify the above expression to be:
\[
C_{HD} \approx (21 + 2 \quad \text{ITR}_2 \quad \text{ITR}_1) N^3, \quad (23)
\]

For the inner loop, implemented using bi-section, the convergence rate is linear. The number of iterations required to
achieve a given tolerance in the solution are $ITR_1 = \log_2 \frac{\epsilon}{\epsilon_0}$ [8], where $\epsilon_0 = -\lambda_{\min}(H^T H) + \eta_{\max}$ is the size of the initially bracketing interval, and $\epsilon$ is the desired ending tolerance. For the outer line search, using the Golden Section search, each iteration of the line search approximately reduces the original interval by a factor of 0.618 and therefore, after $ITR_2$ iterations the updated search interval is $0.618^{ITR_2} t_{\max}$ [8], where $t_{\max}$ is the original interval size. To summarize, the overall complexity of the HC algorithm is deterministic, and doesn’t depend on $\sigma_w^2$ and $\sigma_g^2$ which makes it appealing for real-time systems.

IV. MAP ESTIMATION USING BEM

In this section we demonstrate the application of the BEM methodology in solving the MAP estimation problem in context of model (2). At each iteration, the algorithm maximizes the expected log likelihood. The expectation is taken with respect to $G$ to integrate this parameter out of the target posterior. This allows the maximization step to obtain an updated estimate of the MAP of $x$ for the target posterior $p(x \mid y)$.

$$x_{n+1} = \arg \max_x E_{G \mid y; x_n} \{ \log p(y \mid G, x) \} \quad (24)$$

We now present a summary of the solution of (24) (full details can be found in our earlier paper [2]):

$$x_{n+1} = \left( \Phi_2(y, x_n) + \frac{\sigma_w^2}{\sigma_z^2} I \right)^{-1} \Phi_1(y, x_n)^T y, \quad (25)$$

where

$$\Phi_1(y, x_n) = H + \frac{1}{\|x_n\|^2 + \frac{\sigma_w^2}{\sigma_z^2}} (y - Hx_n)x_n^T, \quad (26)$$

$$\Phi_2(y, x_n) = \Phi_1(y, x_n)^T \Phi_1(y, x_n) + \left( I - \frac{\sigma_w^2 x_n x_n^T}{\|x_n\|^2 + \frac{\sigma_w^2}{\sigma_z^2}} \right) \sigma_g^2 N. \quad (27)$$

A. Initial guess of $x_0$

In practice, selection of the initial guess will influence the ability of the algorithm to obtain an optimal solution. Typically the influence is seen through the following aspects: number of iterations to achieve a solution with a desired tolerance, the ability to find a global solution as opposed to a local solution and computational effort. As discussed previously, the problem addressed in this paper is non-convex, hence convergence to the optimal solution is not guaranteed. As a simple computationally efficient alternative, we propose to choose the initial guess as $x_0 = \hat{x}_{LMMSE}$. In Section VI we analyze the sensitivity of the BEM algorithm to the choice of the initial guess. The BEM algorithm is presented in Algorithm 3.

B. Complexity Analysis

The most intensive part in solving (24) is the matrix inversion in (25). However, the matrix inversion can be avoided by rewriting (25) as

$$\left( \Phi_2(y, x_n) + \frac{\sigma_w^2}{\sigma_z^2} I \right) x_{n+1} = \Phi_1(y, x_n)^T y. \quad (28)$$

Therefore, at each iteration a solution of $K$ linear equations is required. Solving the set of equations using LU factorization [9], requires $\frac{2}{3} K^3 + 2 K^2$ [10]. Next we evaluate the complexity of forming the matrices $\Phi_1$ and $\Phi_2$. The number of flops for evaluating $\Phi_1$ is approximately $(4NK + 2K + N)$ [flops] and for $\Phi_2$ it is approximately $(2K^2N + 2K^2)$ [flops]. Therefore, the overall complexity of a single BEM iteration is approximately

$$c_{BEM} \approx \frac{2}{3} K^3 + 4K^2 + 2K^2 N + 4NK + 2K + N \text{ [flops]}. \quad (29)$$

As in the previous Section, if $N$ and $K$ are of the same order of magnitude, and ignoring all terms except the leading terms, we can simplify the overall complexity of the BEM to be

$$C_{BEM} \approx c_{BEM} \ ITR_{BEM} = 2 \frac{2}{3} N^3 ITR_{BEM}, \quad (30)$$

where $ITR_{BEM}$ is the number of iterations required till convergence. Unfortunately, $ITR_{BEM}$ is a random number which depends on the values of $\sigma_w^2$, $\sigma_g^2$ and $x_0$ and therefore, the complexity of the BEM is random and can be only evaluated via simulations. In Section VI we evaluate the number of iterations for different values of $\sigma_w^2$, $\sigma_g^2$ and $x_0$.

V. ESTIMATION WITH UNKNOWN NOISE VARIANCE

In this section we derive a novel MAP estimator for model (2) where inference is performed for both $x$ and $\sigma_w^2$, and we present a solution. To achieve this we need to solve the following

$$\left( \hat{x}, \hat{\sigma}_w^2 \right) = \arg \max_{x, \sigma_w^2} p(x, \sigma_w^2 \mid y). \quad (31)$$

Before presenting the solution, we note that the EM approach from Section IV would be difficult to perform efficiently, since now the expectation step involves a double integral. We propose an alternative approach which separates the problem into an iterative procedure consisting of the BEM algorithm as before, and a conditional optimization. An approach to solving (31) would be to introduce a change of variable $\zeta$, given by $\zeta = \sigma_g^2 |x|^2 + \sigma_w^2$. Our solution to this problem will be achieved utilizing the concept of annealed Gibbs sampling [11]. The idea is that since we are only after the maximum of the posterior, instead of working with the actual distribution
$p(x, \zeta | y)$, we instead work with a ‘heated’ version of it. The concept comes from statistical physics, where a metal in the molten state (at a very high temperature) is slowly cooled so that the system at any time is approximately in thermodynamic equilibrium. As cooling proceeds, the system becomes more ordered. When this concept is applied to find the mode of a distribution, one scheme is to consider the target posterior raised to a power, which is analogous to a temperature. As the temperature increases, the mass concentrates on the mode. Asymptotically, as the temperature goes to infinity, the result is a Dirac mass at the mode.

Here we adopt a Gibbs sampling approach. This allows us to separate the problem into iterations which involve the original BEM to optimize one of the full conditionals $p(x|y, \zeta)$ as before, and the optimization of $p(\zeta|y, x)$.

We define the prior density for $(\zeta|x)$ to be an Inverse Gamma distribution defined over the support $\zeta > 0$, given by

$$p(\zeta | x; \alpha, \beta) \sim IG(\zeta; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \zeta^{-\alpha - 1},$$

with shape parameter $\alpha$ and scale parameter $\beta$. This is a popular choice of prior, since it includes the option of an uninformative prior, and also caters for conjugacy under the Normal Inverse Gamma model [12]. We are free to specify the parameters of the prior as we wish. One way to do this when no prior information is available is to try to capture a diffuse prior. Here we take this notion and specify the parameters of the prior as follows: we have chosen the shape parameter $\alpha = 2$ to capture heavy tailed or diffuse prior. Next we center the prior in such a way that it includes some knowledge of the model, whilst remaining diffuse. Wet set the mean of the Inverse Gamma, given by

$$E(\zeta) = \frac{\beta}{\alpha - 1}.$$  

(34)

to be $|x|^2$, leading to $\beta = |x|^2$.

Having described the Bayesian model, we first write the conditional distributions

$$p(\zeta | x, y) \propto p(y | x, \zeta) p(\zeta | x) = \mathcal{N}(Hx, \zeta I) IG(\alpha, \beta)$$  

(35)

$$p(x | \zeta, y) \propto p(y | \zeta, x) p(\zeta | x) p(x)$$

$$= \mathcal{N}(Hx, \zeta I) IG(\alpha, \beta) \mathcal{N}(0, \sigma_w^2 I)$$  

(36)

Due to the conjugacy of the Normal Inverse Gamma model in (35) [12], the full posterior for $\zeta$ is given by

$$p(\zeta | x, y) \propto IG(\bar{\alpha}, \bar{\beta})$$  

(37)

where

$$\bar{\alpha} = \alpha + \frac{N}{2},$$

$$\bar{\beta} = \beta + \frac{\sum_{i=1}^{N} (y[i] - (Hx)[i])^2}{2},$$  

(38)

where $N$ is the number of elements in $y$. The method described in (32) is performed as follows:

Step 1: evaluate

$$\hat{\zeta}_n = \arg \max_{\zeta} p(\zeta | y, x) \bigg|_{\hat{x}_{n-1}},$$

$$\hat{x}_n = \arg \max_{x} p(x | y, \zeta) \bigg|_{\hat{\zeta}_n}.$$  

(39)

Step 2: simulate using BEM

$$\hat{x}_n = \arg \max_{x} p(x | y, \zeta) \bigg|_{\hat{\zeta}_n}.$$  

(40)

This procedure in (39-40) is repeated until a certain precision is reached.

VI. SIMULATION RESULTS

In this section numerical results illustrating the behavior of our new estimator in a MIMO system under a range of variance values for the random matrix $G$ are provided. For this simulation the parameters are $N = 16$, $K = 4$. The matrix $H$ was chosen as a concatenation of four $4 \times 4$ matrices with unit diagonal elements and 0.5 off-diagonal elements. We ran 5000 computer simulation for every realization. The results are provided. For this $\sigma^2_w$, we first compare the estimation mean squared error (MSE) of the BEM, the HC and the LMMSE solutions for various values of $\sigma^2_w$. This is depicted in Fig. 1. The results show comparable results of the BEM and the HC approaches and a smaller MSE than the LMMSE. Also note that the error floor in all the results is due to the uncertainty in the matrix $G$, and can’t be avoided, even in high SNR values. Next, a comparison of the performance of the BEM algorithm for different initial guess is performed in terms for MSE and the average number of iterations. While the MSE in both cases is comparable and converges to the same value for both cases, as discussed in Section the number of iterations is random. These results are depicted in Fig. 2. The number of iterations required by the initial guess $x_0 = 0$ is much larger than for $x_0 = x_{LIN}$. The number of iterations also depends on the values of $\sigma^2_w$ and $\sigma^2_w$, for both cases. Finally, we evaluate the performance of the estimator presented in Section V for which $\sigma^2_w$ is unknown and compare it with the case where it is given. These results are depicted in Fig. 3. The results show only a slight degradation in performance and demonstrate the robustness of the approach proposed in Section V.

VII. CONCLUSIONS

In this work, we discussed the MAP estimator of a random Gaussian vector $x$ in a linear model with random transformation matrix $G$. We proposed two low complexity algorithms
and provided an efficient solution for this case. We also extended the estimation problem to the case where the variance of the additive noise is unknown and provided an efficient solution for this case.

REFERENCES


Appendix

Proof of Lemma 2

The Matrix identity used in Algorithm 1, can be derived as follows:

Consider the L.H.S. expression \((H^TH + \eta I)^{-1}HT\) and substitute the SVD of \(H\) given by

\[
H = USV^T.
\]  

(41)

Now using the eigenvalue, eigenvector decomposition of \(H^TH = VS\Sigma V^T\) where \(V\) is a matrix of eigenvectors so we note that \(V\) is an orthogonal matrix with the property \(V^TV = VV^T = I\).

\[
(VS\Sigma V^T + \eta VV^T)^{-1} (USV^T)^T
\]  

(42)

Using the fact that the transpose of a diagonal matrix leaves it unchanged we obtain,

\[
(VS\Sigma V^T + \eta VV^T)^{-1} VSU^T.
\]  

(43)

We now perform some matrix algebra to obtain,

\[
(V (S\Sigma + \eta I) V^T)^{-1} VSU^T = V (S\Sigma + \eta I)^{-1} SU^T = V \Lambda \eta U^T
\]  

(44)

Q.E.D.