

# Distributed Wideband Spectrum Sensing

Thomas Kealy<sup>1</sup>, Oliver Johnson<sup>2</sup> and Robert Piechocki<sup>3</sup><sup>‡</sup>

February 18, 2015

## Abstract

We consider the problem of reconstructing wideband frequency spectra from distributed, compressive measurements. The measurements are made by a network of nodes, each independently mixing the ambient spectra with low frequency, random signals. The reconstruction takes place via local transmissions between nodes, each performing simple statistical operations such as ridge regression and shrinkage.

## 1 Introduction

There is an almost ubiquitous growing demand for mobile and wireless data, with consumers demanding faster speeds and better quality connections in more places. Consequently 4G is now being rolled out in the UK and US and with 5G being planned for 2020 and beyond [?].

However, there is constrained amount of frequencies over which to transmit this information; and demand for frequencies that provide sufficient bandwidth, good range and in-building penetration is high.

Not all spectrum is used in all places and at all times, and judicious spectrum management, by developing approaches to use white spaces where they occur, would likely be beneficial.

Broadly, access to spectrum is managed in two, complementary ways, namely through licensed and licence exempt access. Licensing authorises a particular user (or users) to access a specific frequency band. Licence exemption allows any user to access a band provided they meet certain technical requirements intended to limit the impact of interference on other spectrum users.

---

<sup>\*</sup>This work was supported by the Engineering and Physical Sciences Research Council [grant number EP/I028153/1]; Ofcom; and the University of Bristol. The authors would particularly like to thank Gary Clemo of Ofcom for useful discussions.

<sup>†1</sup>Thomas Kealy is with CDT in Communications, MVB, School of Engineering University of Bristol, UK [tk12098@bristol.ac.uk](mailto:tk12098@bristol.ac.uk)

<sup>‡2</sup>Oliver Johnson is with the Department of Mathematics, University Walk, Bristol, University of Bristol, UK. [O.Johnson@bristol.ac.uk](mailto:O.Johnson@bristol.ac.uk)

<sup>§3</sup>Robert Piechocki is with the CSN Group, MVB, School of Engineering, University of Bristol, UK. [r.j.piechocki@bristol.ac.uk](mailto:r.j.piechocki@bristol.ac.uk)

A licence exempt approach might be particularly suitable for managing access to white spaces. Devices seeking to access white spaces need a robust mechanism for learning of the frequencies that can be used at a particular time and location. One approach is to refer to a database, which maps the location of white spaces based on knowledge of existing spectrum users. An alternative approach is for devices to detect white spaces by monitoring spectrum use.

The advantages of spectrum monitoring [?] over persisting a database of space-frequency data are the ability of networks to make use of low-cost low-power devices, only capable of making local (as opposed to national) communications, keeping the cost of the network low and opportunistic channel usage for bursty traffic, reducing channel collisions in dense networks.

The realisation of any Cognitive Radio standard (such as IEEE 802.22 [?]), requires the co-existence of primary (TV users) and secondary (everybody else who wants to use TVWS spectrum) users of the frequency spectrum to ensure proper interference mitigation and appropriate network behaviour. We note, that whereas TVWS bands are an initial step towards dynamic spectrum access, the principles and approaches we describe are applicable to other frequency bands - in particular it makes ultra-wideband .

The challenges of this technology are that Cognitive Radios (CRs) must sense whether spectrum is available, and must be able to detect very weak primary user signals. Furthermore they must sense over a wide bandwidth (due to the amount of TVWS spectrum proposed), which challenges traditional Nyquist sampling techniques, because the sampling rates required are not technically feasible with current RF or Analogue-to-Digital conversion technology.

Due to the inherent sparsity of spectral utilisation, Compressive Sensing (CS) [?] is an appropriate formalism within which to tackle this problem. CS has recently emerged as a new sampling paradigm allowing images to be taken from a single pixel camera for example. Applying this to wireless communication, we are able to reconstruct sparse signals at sampling rates below what would be required by Nyquist theory.

However, even with CS, spectrum sensing from a single machine will be costly as the proposed TVWS band will be over a large frequency range (for instance in the UK the proposed TVWS band is from 470 MHz to 790 MHz, requiring traditional sampling rates of  $\sim 1600$  MHz). CS at a single sensor would still require high sampling rates. In this paper we propose a distributed model, which allows a sensing budget at each node far below what is required by centralised CS. The main contribution of this paper is that the model can be solved in a fully distributed manner - we do not require a central fusion centre as in [?]. Moreover, we are able to show that the set of updates at each nodes takes closed form.

The structure of the paper is as follows: in section (2) we introduce the sensing model, in section (3) we describe the distributed reconstruction algorithm [?], and finally in section (5) we show some results of the reconstruction quality of this model. [?] [?]

## 2 Model

We consider a radio environment with a single primary user (PU) and a network of  $J$  nodes collaboratively trying to sense and reconstruct the PU signal, either in a fully distributed manner (by local communication), or by transmitting measurements to a fusion centre which then solves the linear system.

We try to sense and reconstruct a wideband signal, divided into  $L$  channels. We have a (connected) network of  $J$  ( $= 50$ ) nodes placed uniformly at random within the square  $[0, 1] \times [0, 1]$ . This is the same model, as in [?]. The calculations which follow are taken from [?] as well.

The nodes individually take measurements (as in [?]) by mixing the incoming analogue signal  $x(t)$  with a mixing function  $p_i(t)$  aliasing the spectrum.  $x(t)$  is assumed to be bandlimited and composed of up to  $k$  uncorrelated transmissions over the  $L$  possible narrowband channels - i.e. the signal is  $k$ -sparse.

The mixing functions - which are independent for each node - are required to be periodic, with period  $T_p$ . Since  $p_i$  is periodic it has Fourier expansion:

$$p_i(t) = \sum_{l=-\infty}^{\infty} c_{il} \exp\left(jlt \frac{2\pi}{T_p}\right) \quad (2.0.1)$$

The  $c_{il}$  are the Fourier coefficients of the expansion and are defined in the standard manner. The result of the mixing procedure in channel  $i$  is therefore the product  $x p_i$ , with Fourier transform (we denote the Fourier Transform of  $x$  by  $X(\cdot)$ ):

$$\begin{aligned} X_i(f) &= \int_{-\infty}^{\infty} x(t) p_i(t) dt \\ &= \sum_{l=-\infty}^{\infty} c_{il} X(f - l f_p) \end{aligned} \quad (2.0.2)$$

(We insert the Fourier series for  $p_i$ , then exchange the sum and integral). The output of this mixing process then, is a linear combination of shifted copies of  $X(f)$ , with at most  $\lceil f_N Y Q / f_p \rceil$  terms since  $X(f)$  is zero outside its support (we have assumed this Nyquist frequency exists, even though we never sample at that rate).

This process is repeated in parallel at each node so that each band in  $x$  appears in baseband.

Once the mixing process has been completed the signal in each channel is low-pass filtered and sampled at a rate  $f_s \geq f_p$ . In the frequency domain this is a ideal rectangle function, so the output of a single channel is:

$$Y_i(e^{j2\pi f T_s}) = \sum_{l=-L_0}^{+L_0} c_{il} X(f - l f_p) \quad (2.0.3)$$

since frequencies outside of  $[-f_s/2, f_s/2]$  will filtered out.  $L_0$  is the smallest integer number of non-zero contributions in  $X(f)$  over  $[-f_s/2, f_s/2]$  - at most

$\lceil f_N YQ/f_p \rceil$  if we choose  $f_s = f_p$ . These relations can be written in matrix form as:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w} \quad (2.0.4)$$

where  $\mathbf{y}$  contains the output of the measurement process, and  $\mathbf{A}$  is a product matrix of the mixing functions, their Fourier coefficients, a partial Fourier Matrix, and a matrix of channel coefficients.  $\mathbf{x}$  is the vector of unknown samples of  $x(t)$ .

i.e.  $\mathbf{A}$  can be written:

$$\mathbf{A}^{m \times L} = \mathbf{S}^{m \times L} \mathbf{F}^{L \times L} \mathbf{D}^{L \times L} \mathbf{H}^{L \times L} \quad (2.0.5)$$

The measurements  $\mathbf{y}$  are transmitted to a Fusion Centre via a control channel. The system 2.0.4 can then be solved (in the sense of finding the sparse vector  $\mathbf{x}$  by convex optimisation via minimising the objective function:

$$\frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1 \quad (2.0.6)$$

where  $\lambda$  is a parameter chosen to promote sparsity. Larger  $\lambda$  means sparser  $\mathbf{x}$ .

### 3 ADMM

The alternating direction method of multipliers [?], (ADMM), algorithm solves problems of the form

$$\begin{aligned} \arg \min_x f(x) + g(z) \\ \text{s.t. } Ux + Vz = c \end{aligned} \quad (3.0.7)$$

where  $f$  and  $g$  are assumed to be convex function with range in  $\mathbb{R}$ ,  $U \in \mathbb{R}^{p \times n}$  and  $V \in \mathbb{R}^{p \times m}$  are matrices (not assumed to have full rank), and  $c \in \mathbb{R}^p$ .

ADMM consists of iteratively minimising the augmented Lagrangian

$$\begin{aligned} L_p(x, z, \eta) = f(x) + g(z) + \eta^T (Ux + Vz - c) + \\ \frac{\rho}{2} \|Ux + Vz - c\|_2^2 \end{aligned}$$

( $\eta$  is a Lagrange multiplier), and  $\rho$  is a parameter we can choose to make  $g(z)$  smooth [?], with the following iterations:

$$x^{k+1} := \arg \min_x L_\rho(x, z^k, \eta^k) \quad (3.0.8)$$

$$z^{k+1} := \arg \min_z L_\rho(x^{k+1}, z, \eta^k) \quad (3.0.9)$$

$$\eta^{k+1} := \eta^k + \rho (Ux^{k+1} + Vz^{k+1} - c) \quad (3.0.10)$$

The alternating minimisation works because of the decomposability of the objective function: the  $x$  minimisation step is independent of the  $z$  minimisation step and vice versa.

We illustrate an example, relevant to the type of problems encountered in signal processing.

### 3.1 Example: ADMM for Centralised LASSO

ADMM can be formulated as an iterative MAP estimation procedure for the problem (which is referred to as LASSO see [?]):

$$\frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1 \quad (3.1.11)$$

This can be cast in constrained form as:

$$\frac{1}{2}\|Ax - b\|_2^2 + \lambda\|z\|_1 \quad (3.1.12)$$

$$\text{s.t } z = x \quad (3.1.13)$$

i.e this is of the form (3.0.7) with  $f(x) = \|Ax - y\|_2^2$ ,  $g(z) = \lambda\|z\|_1$ ,  $U = A$ ,  $V = -I$ , and  $c = 0$ .

The associated Lagrangian is:

$$L_\rho = \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|z\|_1 + \eta(x - z) + \frac{\rho}{2}\|x - z\|^2 \quad (3.1.14)$$

Now, given a set of noisy measurements (say of radio spectra)  $\mathbf{y}$ , and a sensing matrix  $\mathbf{A}$  we can use ADMM to find the (sparse) radio spectra.

The ADMM iterations for LASSO, which can be found by alternately differentiating (3.1.14) with respect to  $x, z$  and  $\eta$ , are (in closed form):

$$x^{k+1} := (A^T A + \rho I)^{-1} (A^T b + \rho(z^k - y^k)) \quad (3.1.15)$$

$$z^{k+1} := S_{\lambda/\rho}(x^{k+1} + \eta^k/\rho) \quad (3.1.16)$$

$$\eta^{k+1} := \eta^k + \rho(x^{k+1} - z^{k+1}) \quad (3.1.17)$$

where  $S_{\lambda/\rho}(\circ)$  is the soft thresholding operator:  $S_\gamma(x)_i = \text{sign}(x_i)(|x_i| - \gamma)^+$ .

This algorithm has a nice statistical interpretation: it iteratively performs ridge regression, followed by shrinkage towards zero. This is the MAP estimate for  $x$  under a Laplace prior.

The soft-thresholding operator can be derived by considering the MAP estimate of the following model:

$$y = x + w \quad (3.1.18)$$

where  $x$  is some (sparse) signal, and  $w$  is additive white Gaussian noise. We seek

$$\hat{x} = \arg \max_x \mathbb{P}_{x|y}(x|y) \quad (3.1.19)$$

This can be recast in the following form by using Bayes rule, noting that the denominator is independent of  $x$  and taking logarithms:

$$\hat{x} = \arg \max_x [\log \mathbb{P}_w(y - x) + \log \mathbb{P}(x)] \quad (3.1.20)$$

The term  $\mathbb{P}_w(y - x)$  arises because we are considering  $x + w$  with  $w$  zero mean Gaussian, with variance  $\sigma_n^2$ . So, the conditional distribution of  $y$  (given  $x$ ) will be a Gaussian centred at  $x$ .

We will take  $\mathbb{P}(x)$  to be a Laplacian distribution:

$$\mathbb{P}(x) = \frac{1}{\sqrt{2}\sigma} \exp -\frac{\sqrt{2}}{\sigma} |x| \quad (3.1.21)$$

Note that  $f(x) = \log \mathbb{P}_x(x) = -\frac{\sqrt{2}}{\sigma} |x|$ , and so by differentiating  $f'(x) = -\frac{\sqrt{2}}{\sigma} \text{sign}(x)$

Taking the maximum of 3.1.20 we obtain:

$$\frac{y - \hat{x}}{\sigma_n^2} - \frac{\sqrt{2}}{\sigma} \text{sign}(x) = 0 \quad (3.1.22)$$

Which leads the soft thresholding operation defined earlier, with  $\gamma = \frac{\sqrt{2}\sigma_n^2}{\sigma}$  as (via rearrangement):

$$y = \hat{x} + \frac{\sqrt{2}\sigma_n^2}{\sigma} \text{sign}(x)$$

or

$$\hat{x}(y) = \text{sign}(y) \left( y - \frac{\sqrt{2}\sigma_n^2}{\sigma} \right)_+$$

i.e  $S_\gamma(y)$ .

## 4 Constrained Optimisation on Graphs

We model the network as an undirected graph  $G = (V, E)$ , where  $V = \{1 \dots J\}$  is the set of vertices, and  $E = V \times V$  is the set of edges. An edge between nodes  $i$  and  $j$  implies that the two nodes can communicate. The set of nodes node  $i$  can communicate with is written  $\mathcal{N}_i$  and the degree of node  $i$  is  $D_i = |\mathcal{N}_i|$ .

Individually nodes make the following measurements:

$$\mathbf{y}_p = \mathbf{A}_p \mathbf{x} + \mathbf{n}_p \quad (4.0.23)$$

where  $\mathbf{A}_p$  is the  $p^{th}$  row of the sensing matrix from (2.0.4), and the system (2.0.4) is formed by concatenating the individual nodes measurements together.

We assume that a proper (or approximate) colouring of the graph is available: that is each node is assigned a number from a set  $C = \{1 \dots c\}$ , and no node shares a colour with any neighbour.

To find the  $\mathbf{x}$  we are seeking, to each node we give a copy of  $\mathbf{x}, \mathbf{x}_p$  and we constrain the copies to be identical across all edges in the network. We can write the combined optimisation variable as  $\bar{x}$ , which collects together  $C$  copies of a  $n \times 1$  vector  $\mathbf{x}$ :

**Definition 1.**

$$\bar{x} = \sum_{c=1}^C w_c \otimes x_c = [x_1^T, \dots, x_C^T]^T \quad (4.0.24)$$

where  $w_{c(i)} = \mathbb{I}(c(i) = c)$ ,  $\mathbb{I}$  is the indicator function, and we have written  $c(i)$  for the colour of the  $i$ th node.

The problem then then have to solve:

$$\begin{aligned} \arg \min_{\bar{x}} \sum_{c=1}^C \sum_{j \in c} \|A_j x_j - y_j\|_2^2 + \frac{\lambda}{J} \|\bar{x}\|_1 \\ \text{and } x_i = x_j \text{ if } \{i, j\} \in E \\ \text{and } x_i = z_i \forall i \in \{1, \dots, C\} \end{aligned} \quad (4.0.25)$$

These constraints can be written more compactly by introducing the node-arc incidence matrix  $B$ : a  $V$  by  $E$  matrix where each column is associated with an edge  $(i, j) \in E$  and has 1 and  $-1$  in the  $i$ th and  $j$ th entry respectively. Figures (4.1) and (4.2) show examples of a network and it's associated incidence matrix.

The constraint  $x_i = x_j$  if  $\{i, j\} \in E$  can now be written

$$\sum_{c=1}^C (B_c^T \otimes I_n) \bar{x}_c = 0 \quad (4.0.26)$$

note that  $(B^T \otimes I_n) \in \mathbb{R}^{nE \times nJ}$ . Together (4.0.24) and (4.0.26), suggests that the problem (4.0.25) can be re-written as:

$$\arg \min_{\bar{x}} \sum_{c=1}^C \sum_{j \in C_c} \|A_j x_j - y_j\|_2^2 + \beta \|z_j\|_1 \quad (4.0.27)$$

$$\text{s.t. } \sum_{c=1}^C (B_c^T \otimes I_n) \bar{x}_c = 0 \quad (4.0.28)$$

$$\text{and } \bar{x}_c - \bar{z}_c = 0 \quad (4.0.29)$$

where  $\beta = \frac{\lambda}{J}$ .

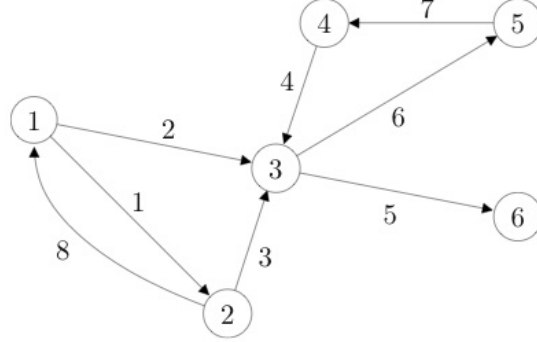


Figure 4.1: An example of a network

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & -1 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}.$$

Figure 4.2: The incidence matrix associated with Figure (4.1)

The Augmented Lagrangian for the problem (4.0.29) can be written down as:

$$L_\rho = \sum_{c=1}^C \sum_{j \in c} (\|A_j x_j - y_j\|_2^2 + \beta \|x\|_1) + \eta^T M_c \bar{x}_c + \frac{\rho}{2} \left\| \sum_{c=1}^C M_c \bar{x}_c \right\|^2 + \frac{\rho N_c}{2} \|x_c - z_c\|^2 \quad (4.0.30)$$

where we have defined  $M_c := (B_c^T \otimes I_n)$ , and where  $N_c$  is the number of nodes with colour  $c$ .

We can tidy up the second and third terms of (4.0.30). To this end, we introduce:



**Definition 2.**

$$\begin{aligned}
u &:= (B^T \otimes I_n) \bar{x} \\
&= (B^T \otimes I_n) \sum_{c=1}^C w_c \otimes x_c \\
&= \sum_{c=1}^C (B^T w_c) \otimes x_c
\end{aligned}$$

where we have used the definition (4.0.24) in the second line, and the property of Kronecker products  $(A \otimes C)(B \otimes D) = (AB \otimes CD)$  between the second and third lines.

The final term  $(\|\sum_{c=1}^C (B_c^T \otimes I_n) \bar{x}_c\|^2)$  of (4.0.30), can be decomposed, using the following lemma:

**Lemma 4.1.**

$$\left\| \sum_{c=1}^C (B_c^T \otimes I_n) \bar{x}_c \right\|^2 = \sum_{j \in C_1} \left( D_j \|x_j\| - \sum_{k \in N_j} x_j^T x^k \right) \quad (4.0.31)$$

*Proof.*

$$\begin{aligned}
u^T u &= \sum_{c_1=1}^C \sum_{c_2=1}^C ((w_{c_1}^T B) \otimes x_{c_1}^T) ((B^T w_{c_2}) \otimes x_{c_2}) \\
&= \sum_{c_1, c_2} w_{c_1}^T B B^T w_{c_2} \otimes x_{c_1}^T x_{c_2}
\end{aligned}$$

For  $c_1 = c_2$   $BB^T$  is a  $J \times J$  matrix, with the degree of the nodes on the main diagonal and  $-1$  in position  $(i, j)$  if nodes  $i$  and  $j$  are neighbours (i.e  $BB^T$  is the graph Laplacian). The trace of  $B_1 B_1^T$  is simply the sum of the degrees of nodes with colour 1.

For  $c_1 \neq c_2$ ,  $B_1 B_c^T$  corresponds to an off diagonal block of the graph Laplacian, and so counts how many neighbours each node with colour 1 has.  $\square$

Finally, we need to consider the second term from (4.0.30):

$$\eta^T (B_1^T \otimes I_n) \bar{x}_1 = \sum_{l \in C_1} \sum_{m \in N_l} \text{sign}(m - l) \eta_{ml}^T x_l \quad (4.0.32)$$

where  $\eta$  is decomposed edge-wise:  $\eta = (\dots, \eta_{ij}, \dots)$ , such that  $\eta_{i,j} = \eta_{j,i}$  and is associated with the constraint  $x_i = x_j$ .

Adding together this with the lemma, lets us write (4.0.30) as:

$$\begin{aligned}
L_\rho &= \sum_{c=1}^C \sum_{j \in C_c} \|A_c x_c - y_c\|_2^2 + \beta \|x_c\|_1 + \nu^T x_c \\
&\quad + \frac{\rho}{2} D_i \|x_c\|^2 + \frac{\rho N_c}{2} \|x_c - z_c\|^2
\end{aligned} \quad (4.0.33)$$

where we have defined:

$$\nu_i = \left( \sum_{k \in \mathcal{N}_i} \text{sign}(k - i) \eta_{\{i,k\}} - \rho x_k \right) \quad (4.0.34)$$

this is a rescaled version of the Lagrange multiplier,  $\eta$ , which respects the graph structure.

Then by differentiating (4.0.33) with respect to  $x_j$  and  $z_j$  we can find closed forms for the updates as:

$$x_j^{k+1} := (A_j^T A_j + (\rho D_J + N_c)I)^{-1} (A_j^T y_j + N_c z^k - \nu^{kT}) \quad (4.0.35)$$

$$z^{k+1} := S_{\beta/\rho}(x_j^{k+1}) \quad (4.0.36)$$

$$\nu^{k+1} := \nu^k + \rho(x^{k+1} - z^{k+1}) \quad (4.0.37)$$

## 5 Results

The model described in section (2), equation (2.0.4) was simulated, with a wideband signal of 201 channels and a network of 50 nodes (i.e. the signal will be sampled at a 1/4 of rate predicted by Nyquist theory). The mixing patterns were generated from iid Gaussian sources (i.e the matrix  $S$  had each entry drawn from an iid Gaussian source). Monte Carlo simulations were performed at SNR values ranging from 1 to 20, and the expected Mean Squared Error (MSE) of solutions of a centralised solver (spgl1) and a distributed solver (ADMM) were calculated over 10 simulations per SNR value. The results can be seen in fig (5.3).

These results indicate that for both centralised and distributed solvers, adding noise to the system results in a degrading of performance. Interestingly note, that the distributed solver seems to (slightly) outperform the centralised solver at all SNRs. This is counter-intuitive, as it would be expected that centralised solvers knowing *all* the available information would outperform distributed solutions. We conjecture that the updates described in section (4), take into account differences in noise across the network. The distributed averaging steps, which form the new prior for each node, then penalise updates from relatively more noisy observations. This corroborates observations from [?].

This observation is (partially) confirmed in figure (5.4), which plots the progress of the centralised and distributed solvers (as a function of iterations) towards the optimum solution. The SNR is 0.5 (i.e the signal is twice as strong as the noise). Note that after around 300 iterations, the MSE of the distributed solver is consistently below that of the centralised solver.

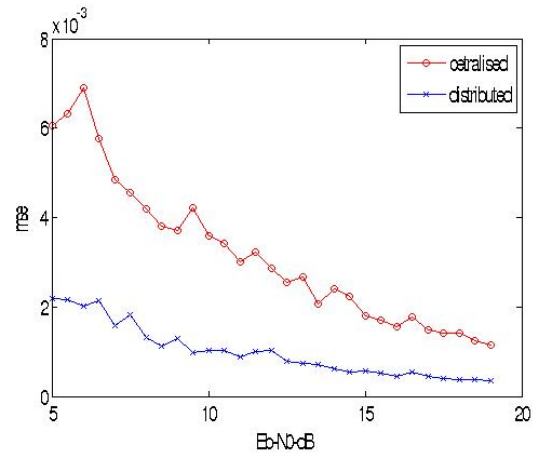


Figure 5.3: Mse vs SNR for the sensing model, showing the performance of distributed and centralised solvers

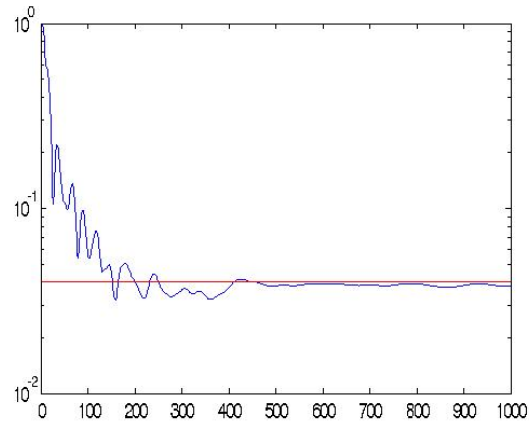


Figure 5.4: The progress of a distributed (red) and a centralised (blue) solver as a function of the number of iterations