End of Year Report

October 1, 2015

Contents

1 Summary

- Extending the DADMM algorithm to composite functions.
- Providing new, clearer, proofs of the edge decomposition of the Lagrangian
- Extending the Modulated Wideband Converter to model signals composed of rectangles.
- De-noising OFCOM spectrum data.

Table 1: Things Which Do/Don't Work

	Haar Wavelets	db10 Wavelets	Splines	Differences
Centralised	Y	Y	Y	Y
Distributed	Y	N	Y	Y
Noise	N(0,1)	N(0,1)	N(0,1)	N(0,1)
Regulariser	$\lambda \|\|x\|\|_1$	$\lambda \|\|x\|\|_1$	$\lambda \|\|x\|\|_1$	$\lambda \|\|x\|\|_1$
Lambda	$\sqrt{2\log(M)}$ / $\sqrt{40\log(M)}$	$\sqrt{2\log(M)}$	$\sqrt{2\log(M)}$ $2.5\sqrt{2\log(M)}$	$\sqrt{2\log(M)}$ / $10\sqrt{2}$
Step Size	0.5	$\frac{1}{max\|eig(A^TA)\|}$	$\frac{1}{max\|eig(A^TA)\|}$	$\frac{1}{max\|eig(A^TA)\|}$

2 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 [5].

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.

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 [1] over maintaining 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 [10]), requires the co-existence of primary (e.g. 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 spectrum sensing possible.

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) [4] 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, for example the works [7], and [11] detail how this sampling can be achieved.

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 ~600 MHz). CS at a single sensor would still require high sampling rates. In this report we propose a distributed model, which allows a sensing budget at each node far below what is required by centralised CS.

The contributions of this report are that we propose a distributed model and solver pair which obviates the need for a Fusion Centre (centralised node) as in [12]) to do any data processing. That is the solution is found in a distributed manner, by local computations and communications in with one-hop neighbours. This can be applied to other models which previously required central processing.

Moreover, our algorithm is simple to understand (it is an extension of the multi-block ADMM [8]) and can be applied to other composite optimisation problems.

We also give new proofs of ideas found in [8].

The structure of the report is as follows: in section 5 we introduce the sensing model, in section 4 we describe the distributed reconstruction algorithm [8], and finally in section 7 we show some results of the reconstruction quality of this model.

3 ADMM

Given a set of measurements of the form

$$y = Ax + n \tag{3.0.1}$$

where $x \in \mathbb{R}^n$ is an s-sparse vector we wish to recover, $y \in \mathbb{R}^m$ is a set of noisy measurements, $A \in \mathbb{R}^{m \times n}$ is a design or measurement matrix s.t. x is not in the null-space of A, and $z \in \mathbb{R}^m$ is AGWN. The signal x can be recovered by algorithms minimising the objective function:

$$L = \frac{1}{2}\left|\left|Ax - y\right|\right|_2^2 + \lambda\left|\left|x\right|\right|_1 \tag{3.0.2}$$

where λ is a parameter which trades off the reconstruction accuracy and sparsity of x: larger λ means sparser x.

One such algorithm is the alternating direction method of multipliers [3], (ADMM). This algorithm solves problems of the form

$$\begin{aligned} &\arg\min_x f(x) + g(z) \\ &\text{s.t } Ux + Vz = c \end{aligned} \tag{3.0.3}$$

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

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

(η is a Lagrange multiplier), and ρ is a parameter we can choose to make $g(z)$ smooth [9], with the following iterations:

$$x^{k+1} := \arg\min_x L_\rho\left(x, z^k, \eta^k\right) \tag{3.0.4}$$

$$z^{k+1} := \arg\min_z L_\rho\left(x^{k+1}, z, \eta^k\right) \tag{3.0.5}$$

$$\eta^{k+1} := \eta^k + \rho\left(Ux^{k+1} + Vz^{k+1} - c\right) \tag{3.0.6}$$

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.

ADMM can be formulated as an iterative MAP estimation procedure for the problem (3.0.2). We can write (3.0.2) in constrained form as:

$$\frac{1}{2}\|Ax - b\|_2^2 + \lambda\|z\|_1 \tag{3.0.7}$$

$$\text{s.t } z = x \tag{3.0.8}$$

i.e this is of the form (3.0.3) with $f(x) = \|Ax - y\|_2^2$, $g(z) = \lambda\|z\|_1$, $U = I$, $V = -I$, and $c = 0$.

The associated (augmented) Lagrangian is:

$$L_\rho = \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|z\|_1 + \eta(x - z) + \frac{\rho}{2}\|x - z\|^2 \tag{3.0.9}$$

The ADMM iterations for LASSO, which can be found by alternately differentiating (3.0.9) with respect to x,z and η, are (in closed form):

$$x^{k+1} := \left(A^T A + \rho I\right)^{-1} \left(A^T y + \rho \left(z^k - \eta^k/\rho\right)\right) \tag{3.0.10}$$

$$z^{k+1} := S_{\lambda/\rho}\left(x^{k+1} + \eta^k/\rho\right) \tag{3.0.11}$$

$$\eta^{k+1} := \eta^k + \rho\left(x^{k+1} - z^{k+1}\right) \tag{3.0.12}$$

where $S_{\lambda/\rho}(\circ)$ is the soft thresholding operator: $S_\gamma(x)_i = \text{sign}(x_i)\left(|x_i| - \gamma\right)^+$.
These can be found differentiating (3.0.9) with respect to x and z as follows:

$$\frac{\partial L}{\partial x} = -A^T\left(y - Ax\right) + \rho(x - z) + \eta$$

as

$$\frac{\partial}{\partial x}\left|\left|F(x)\right|\right|_2^2 = 2\left(\frac{\partial}{\partial x}F(x)\right)F(x) \tag{3.0.13}$$

by the chain rule, and $\partial/\partial x(Ax) = -A^T$ (see the Matrix Cookbook) as differentiation exchanges a linear operator with its adjoint.
Setting (3.0.13) to zero and collecting like terms:

$$\left(A^T A + \rho I\right)x = A^T y + \rho z - \eta \tag{3.0.14}$$

so we find the optimal x is:

$$x = \left(A^T A + \rho I\right)^{-1}\left(A^T y + \rho\left(z - \eta/\rho\right)\right) \tag{3.0.15}$$

note that this estimator is a weighted average of the ordinary least squares estimate ($A^T y$) and a Gaussian prior. This is to be expected, as the minimisation problem w.r.t x is an l_2-regularised MAP problem.
for $z > 0$

$$\frac{\partial L}{\partial z} = \lambda + \rho(x - z) - \eta \tag{3.0.16}$$

from which we obtain:

$$z = x + \frac{1}{\rho}(\eta - \lambda)$$

since $z > 0$ then $x + \frac{1}{\rho}(\eta - \lambda I) > 0$ when $x + \frac{\eta}{\rho} > \frac{\lambda}{\rho}$.
Similarly for $z < 0$:

$$\frac{\partial L}{\partial z} = -\lambda + \rho(x - z) \tag{3.0.17}$$

setting (3.0.17) to zero we obtain:

$$z = x + \frac{1}{\rho}(\eta + \lambda)$$

since $z < 0$ then $x + \frac{1}{\rho}(\eta + \lambda) < 0$ when $x + \frac{\eta}{\rho} < -\frac{\lambda}{\rho}$.
at $z = 0$ we find:

$$-\frac{\lambda}{\rho} \leq x + \frac{\eta}{\rho} \leq \frac{\lambda}{\rho}$$

i.e.

$$\mid x + \frac{\eta}{\rho} \mid \leq \frac{\lambda}{\rho} \tag{3.0.18}$$

combining (3.0.17), (3.0.16), (3.0.18) together we find the optimal z is:

$$z = \text{sign}(x + \frac{\eta}{\rho}) \max\left(\mid x + \frac{\eta}{\rho} \mid -\frac{\lambda}{\rho}, 0\right) \tag{3.0.19}$$

Together (3.0.15), (3.0.19) and the third step of (3.0.12) constitute the steps of the ADMM algorithm.

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:

$$y = x + w \tag{3.0.20}$$

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

$$\hat{x} = \arg\max_x \mathbb{P}_{x|y}(x|y) \tag{3.0.21}$$

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

$$\hat{x} = \arg\max_x [\log \mathbb{P}_w(y - x) + \log \mathbb{P}(x)] \tag{3.0.22}$$

The term $\mathbb{P}_n(y - x)$ arises because we are considering $x + w$ with w zero mean Gaussian, with variance σ_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:

$$\mathbb{P}(x) = \frac{1}{\sqrt{2}\sigma} \exp -\frac{\sqrt{2}}{\sigma}|x| \tag{3.0.23}$$

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 3.0.22 we obtain:

$$\frac{y-\hat{x}}{\sigma_n^2} - \frac{\sqrt{2}}{\sigma} sign(x) = 0 \tag{3.0.24}$$

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 $S_\gamma(y)$.

3.1 The Proximity Operator

The Proximity Operator for a closed, convex, and proper function f (the set of all such functions will be denoted Γ in a Hilbert space $\mathcal{H}$ is defined as:

Definition 3.1 (Proximity Operator)**.**

$$\text{Prox}_f\,(y) := \underset{y \in \mathcal{H}}{\arg\min}\, f\,(y) + \frac{1}{2}\,||y-x||^2 \tag{3.1.25}$$

Intuitively the Proximity Operator approximates a point x by another point y, that is close in the mean-square sense under the penalty f.

The $\text{Prox}\,(\circ)$ operator exists for closed and convex f as $(y) + \frac{1}{2}\,||y-x||^2$ is closed with compact level sets and is unique as $(y) + \frac{1}{2}\,||y-x||^2$ is strictly convex.

The corresponding Moreau envelope is defined as

Definition 3.2 (Moreau Envelope)**.**

$$\text{M}_f\,(y) := \min_{y \in \mathcal{H}} f\,(y) + \frac{1}{2}\,||y-x||^2 \tag{3.1.26}$$

The Moreau envelope is a strict generalisation of the squared distance function. M_f is real valued - even when f takes the value ∞, whilst Prox_f is $\mathcal{H}$-valued.

3.1.1 Properties

Theorem 3.3 (Moreau '65)**.** *Let $f \in \Gamma$ and f^* be its Fenchel conjugate. Then the following are equivalent:*

- $z = x + y, y \in \partial f\,(x)$

- $x = \text{Prox}_f(z), y = \text{Prox}_{f^*}(z)$

Theorem 3.4 (Moreau '65)**.** *Let $f \in \Gamma$. Then for all $z \in \mathcal{H}$*

- $\text{Prox}_f(z) + \text{Prox}_{f^*}(z) = z$
- $\text{M}_f(z) + \text{M}_{f^*}(z) = \frac{1}{2}||z||^2$

Theorem 3.5 (Moreau '65)**.** *The Moreau envelope is (Frechet) differentiable, with*

$$\nabla \text{M}_f = Id - \text{Prox}_f = \text{Prox}_{f^*} \tag{3.1.27}$$

Theorem 3.6 (Moreau '65)**.** $\text{Prox}_f : (\mathcal{H}, ||\circ||) \leftarrow (\mathcal{H}, ||\circ||)$ *is 1-Lipchitz continuous.*

3.1.2 Motivation

We are solving problems of the following form:

$$\min_{x \in \mathcal{H}} f(x) + g(z) \tag{3.1.28}$$

$$\text{s.t } x - z = 0 \tag{3.1.29}$$

with $f, g \in \Gamma$. To solve this problem we form the augmented Lagrangian:

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

and then performing the following iterative minimisation:

$$x^{k+1} := \arg\min_x L_\rho\left(x, z^k, \eta^k\right) \tag{3.1.30}$$

$$z^{k+1} := \arg\min_z L_\rho\left(x^{k+1}, z, \eta^k\right) \tag{3.1.31}$$

$$\eta^{k+1} := \eta^k + \rho\left(x^{k+1} + z^{k+1}\right) \tag{3.1.32}$$

i.e.

$$x^{k+1} := \arg\min_x \left(f(x) + \eta^{kT} x + \frac{\rho}{2}\left|\left|x - z^k\right|\right|^2\right) \tag{3.1.33}$$

$$z^{k+1} := \arg\min_z \left(g(z) - \eta^{kT} z + \frac{\rho}{2}\left|\left|x^{k+1} - z\right|\right|^2\right) \tag{3.1.34}$$

$$\eta^{k+1} := \eta^k + \rho\left(x^{k+1} + z^{k+1}\right) \tag{3.1.35}$$

pulling the linear terms into the quadratic ones we get:

$$x^{k+1} := \arg\min_x \left(f(x) + \frac{\rho}{2} \left|\left| x - z^k + (1/\rho)\,\eta^k \right|\right|^2 \right) \tag{3.1.36}$$

$$z^{k+1} := \arg\min_z \left(g(z) + \frac{\rho}{2} \left|\left| x^{k+1} - z - (1/\rho)\,\eta^k \right|\right|^2 \right) \tag{3.1.37}$$

$$\eta^{k+1} := \eta^k + \rho\left(x^{k+1} + z^{k+1}\right) \tag{3.1.38}$$

i.e.

$$x^{k+1} := \mathrm{Prox}_f\left(z^k - u^k\right) \tag{3.1.39}$$

$$z^{k+1} := \mathrm{Prox}_f\left(x^{k+1} + u^k\right) \tag{3.1.40}$$

$$u^{k+1} := u^k + \left(x^{k+1} + z^{k+1}\right) \tag{3.1.41}$$

with $u^k = (1/\rho)\,\eta^k$.

The motivation for the Proximal operator should now be clear: to perform the minimisation we simply calculate the proximal operator of each of the functions at each step. For many functions found in Statistics (e.g. the l_p norms, this can be found in closed form, and so ADMM presents a particularly attractive method for finding MAP solutions to regularised statistical problems.

3.1.3 Examples

Example 3.7 (Indicator)**.** *From the definition*

$$\mathrm{Prox}_I(x) := \arg\min_y I_C(y) + \frac{1}{2}||y - x||^2 \tag{3.1.42}$$

$$= \arg\min_{y \in C} \frac{1}{2}||y - x||^2 \tag{3.1.43}$$

$$= P_C(x) \tag{3.1.44}$$

where $I_C(y)$ is the indicator of some set C and P_C is the projection operator onto that set.

Example 3.8 (l_2 norm)**.** *For $f(y) = \frac{\mu}{2}||y||^2$ the* Prox *operator is:*

$$\mathrm{Prox}_f(x) := \arg\min_y \frac{\mu}{2}||y||^2 + \frac{1}{2}||y - x||^2 \tag{3.1.45}$$

$$= \frac{1}{1+\mu}x \tag{3.1.46}$$

Example 3.9 (l_1 norm)**.** $f = \|x\|_1$

$$\mathrm{Prox}_f(x) := \mathrm{sign}(x_i)\left(|x_i| - \gamma\right)^+ = S_\gamma(x)_i \tag{3.1.47}$$

Example 3.10 (Elastic Net). *Consider*

$$f(x) = \lambda \|x\|_1 + \mu \, ||x|| \tag{3.1.48}$$

$$\text{Prox}_f \, (x) := \frac{\lambda}{1+\mu} S_\gamma \, (x)_i \tag{3.1.49}$$

Example 3.11 (Fused Lasso). *Consider*

$$f(x) = \|x\|_1 + \sum_{i=1}^{d-1} (x_i - x_{i-1}) \tag{3.1.50}$$

i.e the sum of the l_1 and TV norms

$$\text{Prox}_f \, (x) := \text{Prox}_{l_1} \circ \text{Prox}_{TV} = S_\gamma \, (\text{Prox}_{TV})_i \tag{3.1.51}$$

Example 3.12 (Consensus). *Suppose we want to solve a problem such as:*

$$\underset{x}{\textit{minimize}} \quad \sum_i f_i \, (x)$$

this could arise in statistical computing where f_i would be the loss function for the i^th block of training data. We can write the problem for distributed optimisation as:

$$\begin{aligned} &\underset{x}{\textit{minimize}} && \sum_i f_i \, (x_i) \\ &\textit{subject to} && x_i - z = 0 \end{aligned}$$

where x_i are local variables (for example local to each node in a spectrum sensing) and $x_i - z = 0$ are the consensus constraints. Consensus and regularisation can be achieved by adding a regularisation term $g \, (z)$ - for example $g \, (z) = \lambda ||x||_1$ corresponds to the LASSO, and the f_i would be $f_i = ||A_i x_i - b||_2^2$.

As per the previous sections, we form the Augmented Lagrangian:

$$L_\rho \, (x, y) = \sum_i^n \left(f_i \, (x_i) + y_i^T \, (x_i - z) + \frac{\rho}{2} \|x_i - z\|_2^2 \right) \tag{3.1.52}$$

The ADMM iterations for this Lagrangian are:

$$x_i^{k+1} := \arg\min x_i \left(f_i \, (x_i) + y_i^{kT} \, (x_i - z) + \frac{\rho}{2} \|x_i - z\|_2^2 \right) \tag{3.1.53}$$

$$z^{k+1} := \frac{1}{n} \sum_i^n \left(x_i^{k+1} + (1\rho) \, y_i^k \right) \tag{3.1.54}$$

$$y_i^{k+1} := y_i^k + \rho \left(x_i^{k+1} - z^{k+1} \right) \tag{3.1.55}$$

The z^{k+1} iteration is analytic as we're minimising the squared norm of $x_i - z$ - so we average. With $\|x\|_1$ regularisation we perform soft-thresholding after the z update.

At each iteration the sum of the dual variables y_i is zero, so the algorithm can be simplified to:

$$x_i^{k+1} := \arg\min x_i \left(f_i\left(x_i\right) + y_i^{kT}\left(x_i - \bar{x}^k\right) + \frac{\rho}{2}\|x_i - \bar{x}^k\|_2^2 \right) \tag{3.1.56}$$

$$y_i^{k+1} := y_i^k + \rho\left(x_i^{k+1} - z^{k+1}\right) \tag{3.1.57}$$

where

$$\bar{x}^k = \frac{1}{n}\sum_i^n x_i^k \tag{3.1.58}$$

This algorithm can be summarised as follows: in each iteration

- *gather x^k and average to get $\bar{x}^k$*
- *scatter the average to nodes*
- *update y_i^k locally*
- *update x_i locally*

Each agent is minimising it's own function, plus a quadratic term (the squared norm) which penalises the agent from moving too far from the previous average.

Note that the 'gather' stage doesn't require a central processor - this can be done in a distributed manner also.

3.2 Statistical Interpretation

At each step k of the algorithm each agent is minimising it's own loss function, plus a quadratic. This has a simple interpretation: we're doing MAP estimation under the prior $\mathcal{N}\left(\bar{x}^k + (1\rho)\, y_i^k, \rho I\right)$. I.e. the prior mean is the previous iteration's consensus shifted by node i disagreeing with the previous consensus.

3.3 Acceleration

4 Constrained Optimisation on Graphs

We model the network of sensors as an undirected graph $G = (V, E)$, where $V = \{1 \ldots J\}$ 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 sensors can communicate. The set of nodes that 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 (as discussed in section 5):

$$\mathbf{y}_p = \mathbf{A}_p \mathbf{x} + \mathbf{n}_p \tag{4.0.59}$$

where $\mathbf{A}_p$ is the p^{th} row of the sensing matrix from (5.1.78), and the system (5.1.78) is formed by concatenating the individual nodes' measurements together.

We assume that a proper 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. This is so that nodes may communicate in colour order, as opposed to communicating individually thus reducing the total number of communication rounds required.

To find the $\mathbf{x}$ we are seeking (the solution to the linear system, 5.1.78), 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. Each node, thus has a separate optimisation to solve, subject to the constraint that it is consistent with its neighbours.

The problem then is to solve:

$$\begin{aligned} \arg\min_{\bar{x}} \sum_{c=1}^{C} \sum_{j \in c} f(x_j) + \frac{\lambda}{J} g(x_j) \\ \text{and } x_i = x_j \text{ if } \{i,j\} \in E \\ \text{and } x_i = z_i \ \forall i \in \{1, \dots, C\} \end{aligned} \tag{4.0.60}$$

with a particular special case being:

$$\begin{aligned} \arg\min_{\bar{x}} \sum_{c=1}^{C} \sum_{j \in c} \|A_j x_j - y_j\|_2^2 + \frac{\lambda}{J} \|z\|_1 \\ \text{and } x_i = x_j \text{ if } \{i,j\} \in E \\ \text{and } x_i = z_i \ \forall i \in \{1, \dots, C\} \end{aligned} \tag{4.0.61}$$

i.e. $f = ||x||_2^2$ and $g = ||x||_1$.

That is, at each node we minimise a Lasso functional constrained to be consistent across edges but that is separable in the l_2 and l_1 norms.

We can write the global optimisation variable as $\bar{x}$, which collects together C copies of a $n \times 1$ vector $\mathbf{x}$:

Definition 1. *We define vectors x_c, where $c = 1, \dots, C$ and write the vector of length nJ:*

$$\bar{x} = \sum_{c=1}^{C} w_c \otimes x_c = \left[x_{c(1)}^T, \dots, x_{c(J)}^T \right]^T \tag{4.0.62}$$

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 ith node.

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 ith and jth 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

$$\sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c = 0 \tag{4.0.63}$$

note that $\left(B^T \otimes I_n\right) \in \mathbb{R}^{nE \times nJ}$. Together (4.0.62) and (4.0.63), suggests that the problem (4.0.61) can be re-written as:

$$\begin{aligned} \arg\min_{\bar{x}} & \sum_{c=1}^{C} \sum_{j \in C_c} f\left(x_j\right) + \frac{\lambda}{J} g\left(z_j\right) \\ \text{s.t.} & \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c = 0 \\ & \text{and } \bar{x}_c - \bar{z}_c = 0 \end{aligned} \tag{4.0.64}$$

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

The global Augmented Lagrangian [3] for the problem (4.0.64) can be written down as:

$$\begin{aligned} L_\rho = \sum_{c=1}^{C} (\sum_{j \in c} & f\left(x_j\right) + \frac{\lambda}{J} g\left(z_j\right) + \\ & + \theta^T \left(\bar{x}_j - \bar{z}_j\right) + \frac{\rho}{2} \left|\left|\bar{x}_j - \bar{z}_j\right|\right|_2^2) + \\ & + \eta^T \left(B_c^T \otimes I_n\right) \bar{x}_c + \frac{\rho}{2} \left|\left| \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c \right|\right|_2^2 \end{aligned} \tag{4.0.65}$$

This is, superficially, similar to the Augmented Lagrangian for the Lasso problem [3][Section 6.4]. That is, the terms indexed by j are a straightforward Lasso problem, constrained by edge-wise variables (indexed by c) forcing consistency across the network. However, the problem (as currently written) is not separable across the edges of the network as the final and penultimate term represent the constraint that the nodes agree on their estimates across edges.

To make it possible that 4.0.65 can be posed as a constrained optimisation problem at each node, we introduce the following variable (so that the the final term of 4.0.65 is separable across edges of the graph):

Definition 2.

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

where we have used the definition (4.0.62) *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, and we write* $B_c = w_c^T B$.

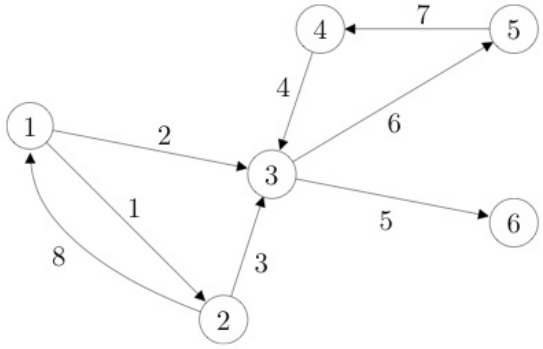

Figure 4.1: An example of a network

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

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

The terms $\| \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c \|^2$ and $\eta^T \left(B_c^T \otimes I_n\right) \bar{x}_c$ of (4.0.65), can be decomposed across edges, using the following lemma:

Lemma 4.1 (Edge Decomposition)**.**

$$\left|\left| \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_c \right|\right|^2 = \sum_{j \in C_1} \left(D_j \left|\left|x_j\right|\right|_2^2 - \sum_{k \in N_j} x_j^T x^k \right) \tag{4.0.66}$$

and

$$\eta^T \sum_{c=1}^{C} \left(B_c^T \otimes I_n\right) \bar{x}_1 = \sum_{l \in C_c} \sum_{m \in N_l} sign\left(m-l\right) \eta_{ml}^T x_l \tag{4.0.67}$$

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

Proof.

$$\begin{aligned} u^T u &= \sum_{c_1=1}^{C} \sum_{c_2=1}^{C} \left(B_{c_1} \otimes x_{c_1}^T\right) \left(B_{c_2}^T \otimes x_c\right) \\ &= \sum_{c_1, c_2} B_{c_1} B_{c_2}^T \otimes x_{c_1}^T x_{c_2} \end{aligned}$$

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). Hence, since we can write $B_{c_1} B_{c_2}^T = w_{c_1}^T B B^T w_{c_2}$, the trace of $B_{c_1} B_{c_1}^T$ is simply the sum of the degrees of nodes with colour 1.

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

Finally, note that $\eta \in \mathbb{R}^{nE}$ and can be written:

$$\eta = \sum_{c=1}^{C} w_c \otimes \eta_c \tag{4.0.68}$$

where η_c is the vector of Lagrange multipliers associated across edges from colour c. Now

$$\eta^T u = \sum_{c_1=1}^{C} \sum_{c_2=1}^{C} w_{c_1} B w_{c_2} \otimes \eta_{c_1}^T x_c$$

by the properties of Kronecker products, and the definition of B_c. For $c_1 = c_2$, $\eta^T u$ is zero, as there are no edges between nodes of the same colour b definition. For $c_1 \neq c_2$, $\eta^T u$ counts the edges from c_1 to c_2, with the consideration that the edges from c_2 to c_1 are counted with opposite parity. □

Adding together this with the lemma, lets us write (4.0.65) as:

$$\begin{aligned} L_\rho = \sum_{c=1}^{C} \sum_{j \in C_c} &\left(f\left(x_j\right) + \beta g\left(z_j\right)\right) + \nu^T x_j \\ &\theta\left(x_j - z_j\right) + \frac{\rho}{2} D_i \left|\left|x_j\right|\right|^2 + \frac{\rho}{2} \|x_j - z_j\|^2 \end{aligned} \tag{4.0.69}$$

where we have defined:

$$\nu_i = \left(\sum_{k \in \mathcal{N}_i} sign\left(k-i\right) \eta_{\{i,k\}} - \rho x_k \right) \tag{4.0.70}$$

this is a rescaled version of the Lagrange multiplier, η, which respects the graph structure.

Then by differentiating (4.0.69) with respect to x_j and z_j we can find closed forms for the updates as:

Theorem 1.

$$x_j^{k+1} := \left(A_j^T A_j + (\rho D_J + 1) I\right)^{-1} \left(A_j^T y_j + z^k - \nu^{kT}\right) \tag{4.0.71}$$

$$z_j^{k+1} := S_{\beta/\rho}\left(x_j^{k+1}\right) \tag{4.0.72}$$

$$\theta_j^{k+1} := \theta_j^k + \rho\left(x^{k+1} - z^{k+1}\right) \tag{4.0.73}$$

$$\eta_j^{k+1} := \eta_j^k + \rho \left(\sum_{m \in N_j} z_m^k - z_j^k \right) \tag{4.0.74}$$

This algorithm can be thought of as follows: each node performs an iteration of (non multi-block) ADMM - i.e. each node solves an approximate Gaussian least-squares problem and then soft-thresholds - and then exchanges the result of this computation with its one-hop neighbours. This explains the inclusion of an extra Lagrange multiplier: the multiplier θ controls how far each node moves from its previous estimate in each iteration, whilst the multiplier η enforces consistency between nodes. Note that there is no communication of data between the nodes - only the result the computation in each round.

5 Compressive Sensing Architechtures

5.1 Modulated Wideband Converter

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 [12]. The calculations which follow are taken from [12] as well.

The nodes individually take measurements (as in [7]) by mixing the incoming analogue signal $x\left(t\right)$ with a mixing function $p_i\left(t\right)$ aliasing the spectrum. $x\left(t\right)$ 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:

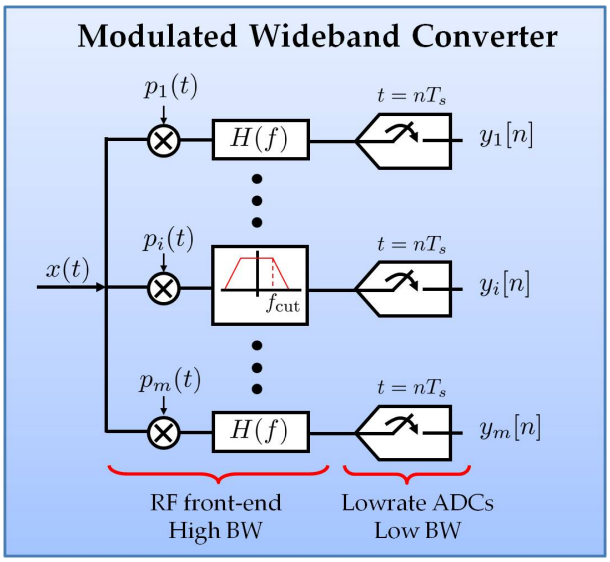

Figure 5.3: Mse vs SNR for the sensing model, with AWGN only, showing the performance of distributed and centralised solvers

$$p_i(t) = \sum_{l=-\infty}^{\infty} c_{il} \exp\left(jlt\frac{2\pi}{T_p}\right) \tag{5.1.75}$$

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 xp_i, with Fourier transform (we denote the Fourier Transform of x by $X(\cdot)$):

$$\begin{aligned} X_i(f) = & \int_{-\infty}^{\infty} x(t)\, p_i(t)\, dt \\ = & \sum_{l=-\infty}^{\infty} c_{il} X(f - lf_p) \end{aligned} \tag{5.1.76}$$

(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 $\lceil f_N YQ/f_p \rceil$ 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:

$$Y_i\left(e^{j2\pi fT_s}\right) = \sum_{l=-L_0}^{+L_0} c_{il} X\left(f - lf_p\right) \tag{5.1.77}$$

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:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w} \tag{5.1.78}$$

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:

$$\mathbf{A}^{m\times L} = \mathbf{S}^{m\times L}\mathbf{F}^{L\times L}\mathbf{D}^{L\times L}\mathbf{H}^{L\times L} \tag{5.1.79}$$

The system 5.1.78 can then be solved (in the sense of finding the sparse vector $\mathbf{x}$ by convex optimisation via minimising the objective function:

$$\frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_1 \tag{5.1.80}$$

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

5.2 Random Demodulator

We assume that the analogue signal $x(t)$ is comprised of a finite number of components from some arbitrary dictionary $\psi_n(t)$:

$$x(t) = \sum_{n=1}^{N} \alpha_n \psi_n(t) \tag{5.2.81}$$

The signal is said to be sparse when there are only a few non-zero α_n. The dictionary elements ψ_n may have a relatively high bandwidth, but the signal itself will have only a few degrees of freedom.

The signal acquisition method proposed consists of three stages (all analogue processing): demodulation, filtering and uniform sampling.

Initially, the signal is modulated by a pseudo-random sequence $p_c(t)$, which alternates at frequencies at (or above) the Nyquist frequency of $x(t)$. The signal is then filtered, through a filter with impulse response $h(t)$, before being sampled at rate $\mathcal{M}$ with a traditional ADC.

The output of this system, $y[m]$, can be related to the input $x(t)$ via a linear transformation of the coefficient vector α_n.

To find the transformation A, first consider the output of $y[m]$, which is the result of convolution and demodulation followed by sampling at rate $\mathcal{M}$:

$$y[m] = \int_{-\infty}^{\infty} x(\tau)\, p_c(\tau)\, h(t-\tau)\, |_{t=m\mathcal{M}}\, d\tau \tag{5.2.82}$$

and by expanding $x(t) = \sum_{n=1}^{N} \alpha_n \psi_n(t)$:

$$y[m] = \sum_{n=1}^{N} \alpha_n \int_{-\infty}^{\infty} \psi_n(t)\, p_c(\tau)\, h(m\mathcal{M}-\tau)\, d\tau \tag{5.2.83}$$

we see that the output can be written as:

$$y = Ax \tag{5.2.84}$$

with

$$A_{m,n} = \int_{-\infty}^{\infty} \psi_n(t)\, p_c(\tau)\, h(m\mathcal{M}-\tau)\, d\tau \tag{5.2.85}$$

6 Joint Space-Frequency Model

We write the power spectral density (psd) of the sth transmitter as:

$$\phi_s = \sum_b \beta_{bs} \psi_b(f) \tag{6.0.86}$$

This model expresses in psd of the transmitter in a suitable basis - for example $\psi_b(f)$ could be zero everywhere except for the set of frequencies where $f = b$ i.e. ψ is a rectangular function with height β_{bs} and support f. Other candidates for ψ include splines (e.g. raised cosines), and complex exponentials.

Given this, the psd at the rth receiver is:

$$\phi_r = \sum_s g_{sr}\phi_s = \sum_s g_{sr} \sum_b \beta_{bs}\psi_b(f) \tag{6.0.87}$$

where

$$g_{sr} = \exp(-||x_r - x_s||_2^{\alpha}) \tag{6.0.88}$$

is the channel response between the sth transmitter and the rth reciver.

This model can be summarised using Kronecker products as follows:

Let $\tilde{G} = g_s r^T$, e_r, e_b be unit vectors i.e. they are 1 for the i^{th} receiver or frequency band respectively.

The received power at a receiver (when only a single transmitter is transmitting) can be written:

$$y_r = \left(e_r^T \bigotimes I_{n_b}\right) y \tag{6.0.89}$$

with,

$$y = \left(\tilde{G} \bigotimes I_{n_b}\right) \phi \tag{6.0.90}$$

Now, we have

$$\phi = e_s \bigotimes \phi_s \tag{6.0.91}$$

so,

$$y = \left(\tilde{G} \bigotimes I_{n_b}\right) \left(e_s \bigotimes \phi_s\right) \tag{6.0.92}$$

finally we have,

$$y_r = \left(e_r^T \bigotimes I_{n_b}\right) \left[\left(\tilde{G} \bigotimes I_{n_b}\right) \left(e_s \bigotimes \phi_s\right)\right] \tag{6.0.93}$$

$\beta_{bs} \in \mathbb{R}^{1 \times n_b}$, $g_{sr} \in \mathbb{R}^{n_r \times n_s}$ and $\psi_{kb} \in 1 \times n_k n_b$ where n_k is the number of frequency bands (in this example $n_k = n_b$.

In the absence of knowledge of the location of the transmitters we introduce a grid of *candidate* locations, to make the above model linear. s now runs over the set of these candidate locations.

The problem of estimating the coefficients, β, from noisy observations $y = \phi_r + N(0, 1)$ is now one that can be tackled by linear regression/convex optimisation.

7 Results

The model described in section (5), equation (5.1.78) 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 5 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 (7.5).

The MSE was calculated as follows:

$$\frac{\left|\left|Z^k - Z*\right|\right|}{||Z*||} \tag{7.0.94}$$

where Z^k is the result of the algorithm at iteration k, and Z^* is the optimal solution.

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 [2].

This observation is (partially) confirmed in figure (**??**), 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.

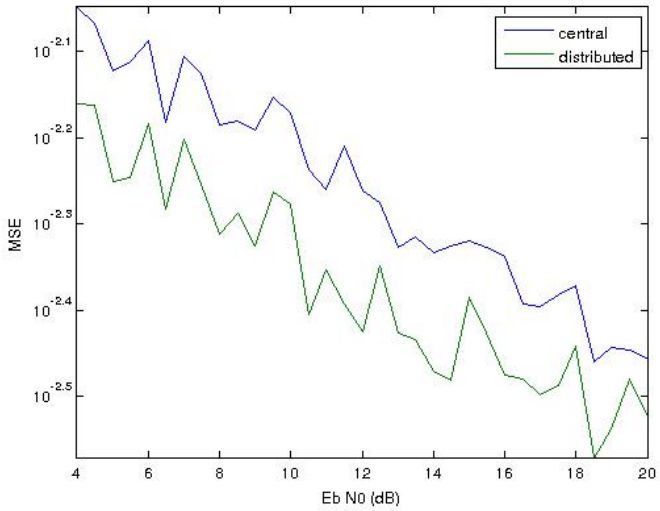

Figure 7.4: Mse vs SNR for the sensing model, with AWGN only, showing the performance of distributed and centralised solvers

8 Conclusions

We have demonstrated an alternating direction algorithm for distributed optimisation with closed forms for the computation at each step, and discussed the statistical properties of the estimation.

We have simulated the performance of this distributed algorithm for the distributed estimation of frequency spectra, in the presence of additive (white, Gaussian) and multiplicative (frequency flat) noise. We have shown that the algorithm is robust to a variety of SNRs and converges to the same solution as an equivalent centralised algorithm (in relative mean-squared-error).

We plan to work on larger, more detailed, models for the frequency spectra and to accelerate the convergence via Nesterov type methods to smooth the

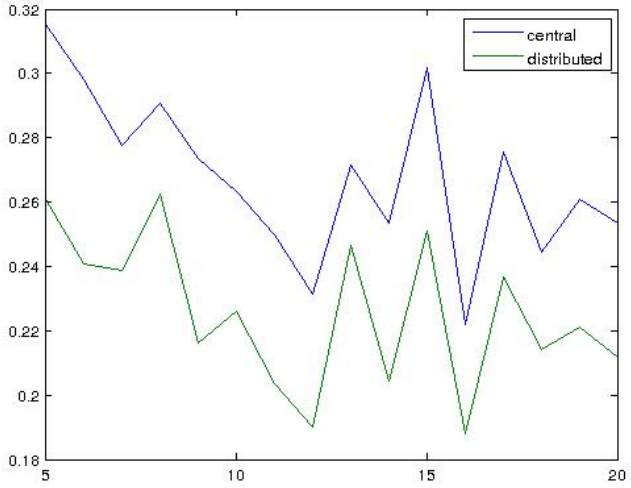

Figure 7.5: Mse vs SNR for the sensing model, showing the performance of distributed and centralised solvers

convergence of the distributed algorithm [6]. Specifically, we seek to dampen the ringing seen in Figure 7.12

References

[1] Ozgur B Akan, O Karli, and Ozgur Ergul. Cognitive radio sensor networks. *Network, IEEE*, 23(4):34–40, 2009.

[2] G B Bazerque, J.A Giannakis. Distributed spectrum sensing for cognitive radios by exploiting sparsity. *Proc. of 42nd Asilomar Conf. on Signals, Systems, and Computers*, 2008.

[3] Stephen Boyd. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends® in Machine Learning*, (1):1–122.

[4] Emmanuel J Candes, Justin Romberg, and Terence Tao. Robust Uncertainty Principles : Exact Signal Frequency Information. 52(2):489–509, 2006.

[5] Eirk Dahlman. 5G wireless acces: Requirements and realization. *IEEE Communications Magazine*, (December):42–47, 2014.

[6] Tom Goldstein, Brendan O'Donoghue, Simon Setzer, and Richard Baraniuk. Fast alternating direction optimization methods. *SIAM Journal on Imaging Sciences*, 7(3):1588–1623, 2014.

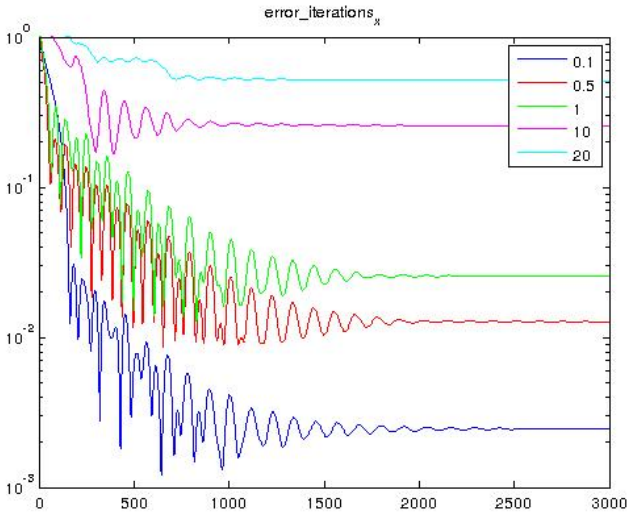

Figure 7.6: The progress of the distributed solver as a function of the number of iterations, with different values of the regression parameter λ

[7] Moshe Mishali and Yonina C Eldar. From theory to practice: Sub-nyquist sampling of sparse wideband analog signals. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):375–391, 2010.

[8] João FC Mota, João MF Xavier, Pedro MQ Aguiar, and Markus Puschel. D-admm: A communication-efficient distributed algorithm for separable optimization. *Signal Processing, IEEE Transactions on*, 61(10):2718–2723.

[9] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[10] Carl R Stevenson, Gerald Chouinard, Zhongding Lei, Wendong Hu, Stephen J Shellhammer, and Winston Caldwell. IEEE 802.22: The first cognitive radio wireless regional area network standard. *IEEE Communications Magazine*, 47(1):130–138, 2009.

[11] Joel A Tropp, Jason N Laska, Marco F Duarte, Justin K Romberg, and Richard G Baraniuk. Beyond nyquist: Efficient sampling of sparse bandlimited signals. *Information Theory, IEEE Transactions on*, 56(1):520–544, 2010.

[12] Huazi Zhang, Zhaoyang Zhang, and Yuen Chau. Distributed compressed wideband sensing in Cognitive Radio Sensor Networks. In *2011 IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2011*, pages 13–17, 2011.

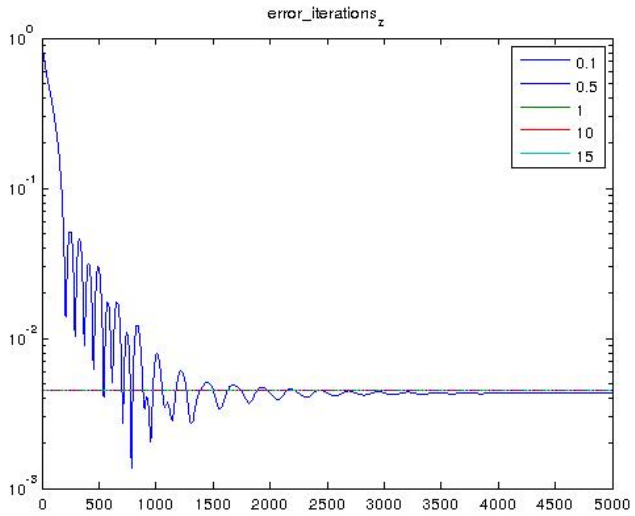

Figure 7.7: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

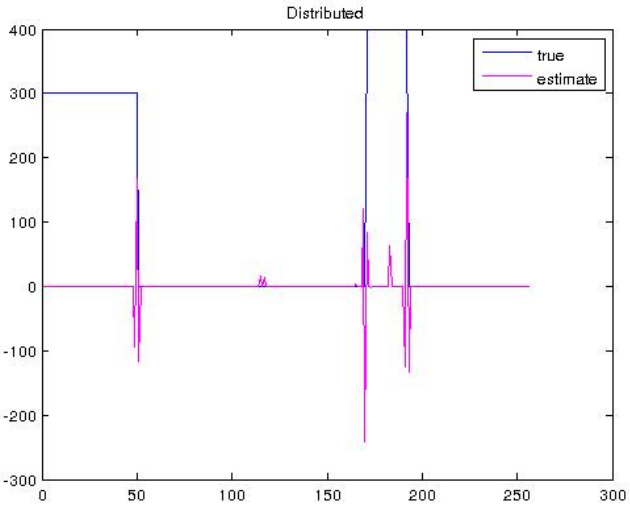

Figure 7.8: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

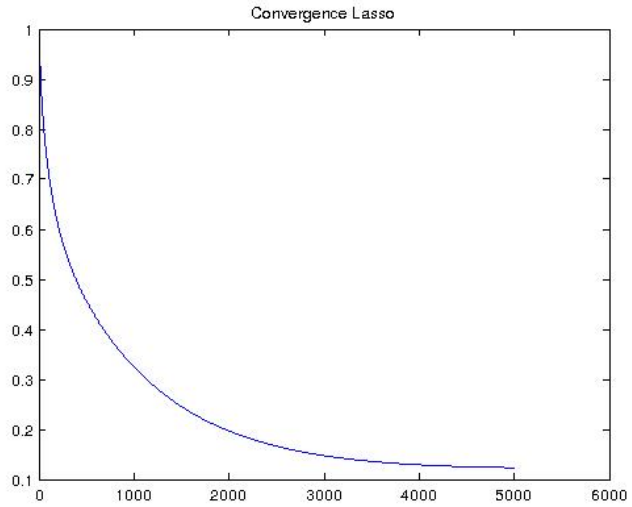

Figure 7.9: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

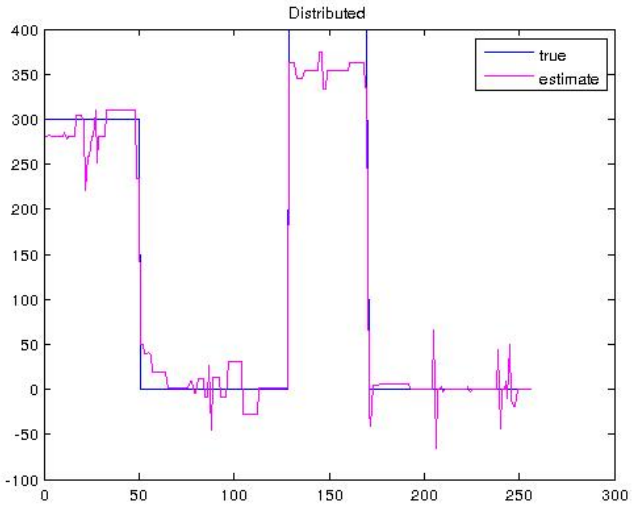

Figure 7.10: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

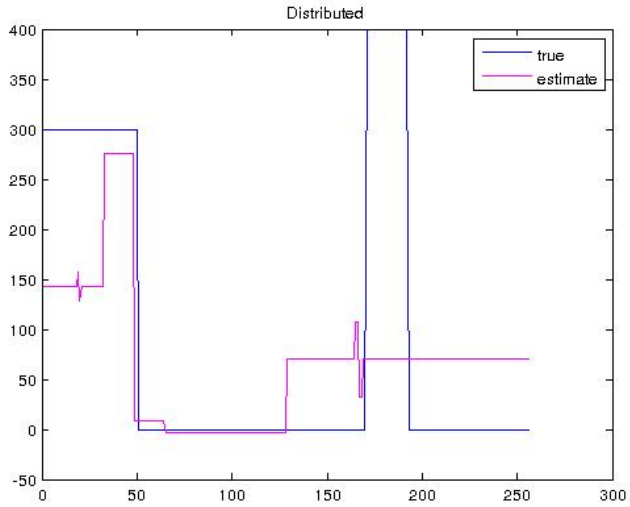

Figure 7.11: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

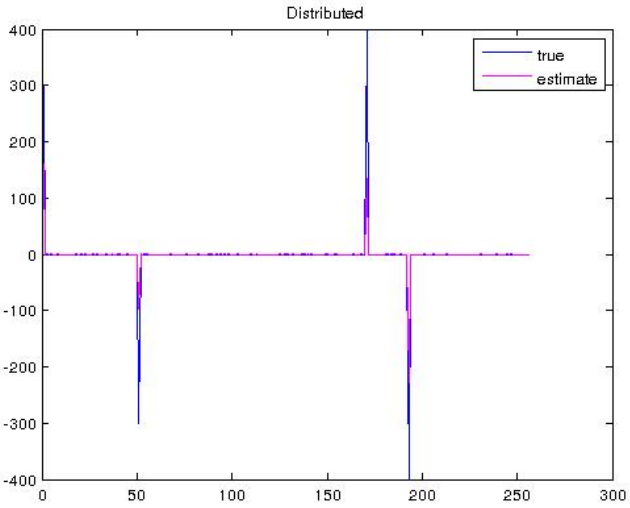

Figure 7.12: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

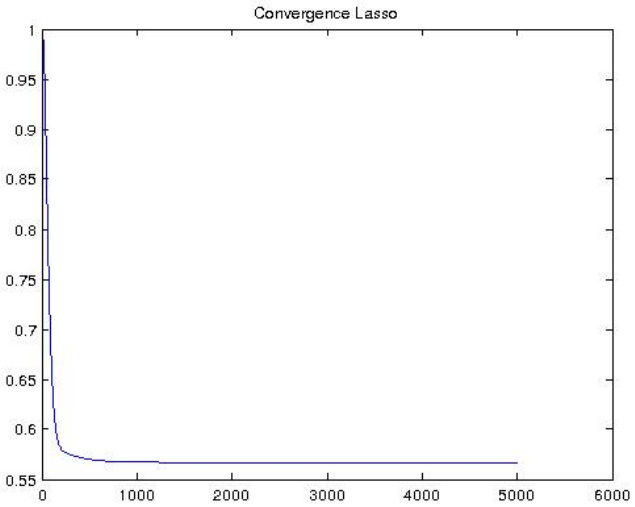

Figure 7.13: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

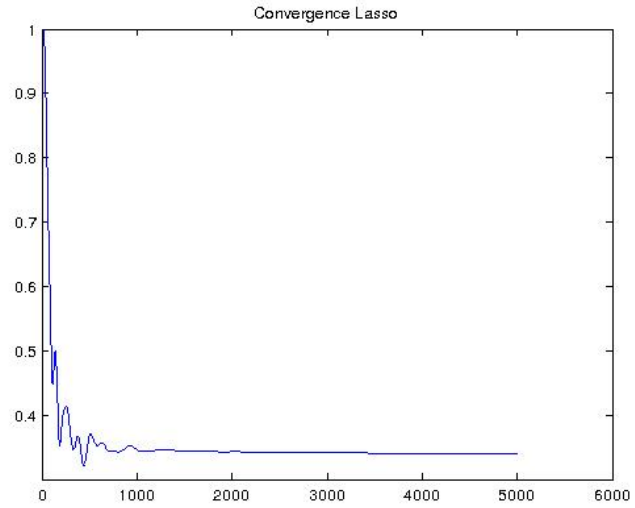

Figure 7.14: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$

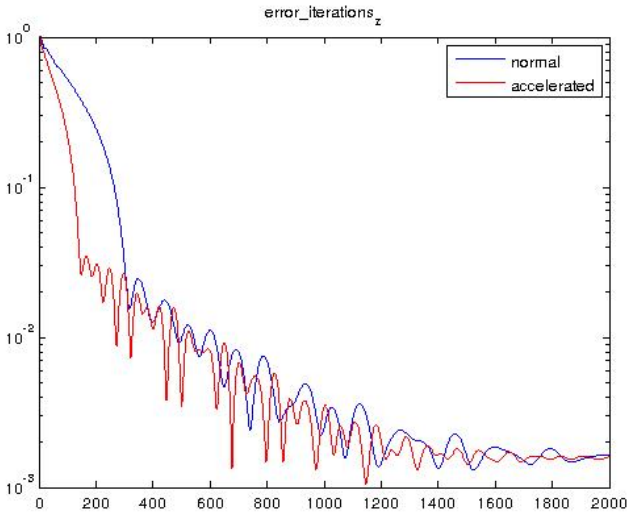

Figure 7.15: The progress of a distributed (blue) and a centralised (green) solver as a function of the number of iterations. The value of $\lambda = 0.1$