Fourier Bayesian networks; A novel approach for network structure inference with application to brain connectivity studies from magnetoencephalographic recordings

Luis Ramón Peraza Rodríguez

PhD

University of York

Department of Electronics

August 2012

Abstract

This thesis proposes a novel approach for connectivity studies in Electrophysiology and Neuroimaging based on Bayesian Network (BN) analysis in the Fourier domain that is named Fourier Bayesian Networks (FBNs). FBNs use the complex information available in time series to make inferences about an unknown network structure. Using the Fourier transform, the frequency power and frequency phase information are estimated; then probabilistic models using power and phase are built and used in the network structure searching algorithm.

FBNs are able to deal with massive datasets with long time series and large numbers of sources. This property is inherited by the Fourier transform from which the Fourier coefficients instead of raw time series are used during network searching.

The analysis of the phase using the Fourier transform makes FBNs non-parametric, meaning that these networks do not rely on a model to make inferences. This is an important property for causality inference since several network unfoldings, as in the case of Dynamic BNs, are not needed. This makes FBNs robust to the underlying model.

The proposed method is tested using multivariate autoregressive (MVAR) and non-linear (NL) systems with the variable model order d. Networks are estimated from the MVAR(d) and NL(d) systems directly and also from a magnetoencephalographic (MEG)-simulated environment where beamforming is implemented for source inference. For all experiments $d = 1$ and $d = 2$ are used. The optimization method for network structure searching is simulated annealing.

Simulations show that FBNs are robust to the model order change, and that this method is able to correctly estimate network structures and functional brain connectivity in MEG studies.

Contents

List of Figures

List of Tables

Acknowledgements

This thesis represents almost four years of work, readings, and lots of coffee and tea cups. For the author it represents an achievement that would not have been possible without the help, support and advices of many people to whom I want to thank in these paragraphs.

First of all I want to thank my parents Faustino Peraza and Teodora Rodriguez for all their support, advice, and education. Thanks for always tell me to pursue my dreams, even if these meant to travel to other continents.

Of course I want to thank my supervisor Dr. David Halliday for all the time he dedicated to my project. Thanks for all our weekly meetings and discussions during these four years and for your comments and advices during the development of this project.

I want also to thank all the Intelligent Systems Group at the Department of Electronics, mainly to my PhD student colleagues, with whom I have had great coffee chats, pub evenings, and UK trips. I will not cite names here because there are many of them, some recently arrived to the group others graduated already and are living new experiences in their new jobs but they know who they are and I want to thank you.

Thanks to Dr. Aziz Asghar from the Hull York Medical School for his help in this project and for the MTBI datasets. Our collaboration will continue for the MTBI project in the following months. There is plenty of research to do yet with these datasets, some of this work is mentioned in the future work chapter in this thesis. Thanks to Prof. Gary Green from the York Neuroimaging Centre, his comments were also very valuable.

And finally I want to thank the National Council on Science and Technology of Mexico (CONACyT) for financing my PhD studies and my stay in the United Kingdom. I want to congratulate CONACyT for the support they also give to hundreds of other Mexican PhD students in the UK and thousands worldwide.

Author's declaration

Some parts of this thesis have been disclosed in the following publications.

Journals and conferences:

- Peraza L. R., Asghar A., Green G., Halliday D.M. , *Volume conduction effects in brain network inference from electroencephalographic recordings using phase lag index*, Journal of Neuroscience Methods, 2012.
  DOI: 10.1016/j.jneumeth.2012.04.007 .

- Peraza L. R., Asghar A., Townend W., Midgley S., Green G., Halliday D. M., *Brain functional connectivity inference in Magnetoencephalography using Fourier Bayesian Networks*, FENS Forum- Barcelona, Spain, 2012. Abstr. A-471-0238-00795.

- Peraza L. R. and Halliday D. M., *Fourier Bayesian information criterion for network structure and causality estimation*, IEEE International Conference on Signals and Electronic Systems, Gliwice - Poland, September 7-10, 2010.

- Peraza L. R. and Halliday D. M., *A fast dynamic Bayesian network algorithm for structure learning using a time-lagged partial correlation matrix*, IEEE International Conference on Signals and Electronic Systems, Gliwice - Poland, September 7-10, 2010.

Introduction

The brain has drawn the attention of researchers for centuries, since the Greek Philosophers such as Hippocrates and Aristotle who developed the first theories about the brain, not exactly about how it works, but a more simple question: What is the purpose of this organ inside our heads? Today much is known about our brain but the way it works is still an unsolved subject that needs the collaboration of researchers from all kind of fields such as Mathematicians, Biologists, Surgeons, Engineers, Psychologists, Philosophers, etc. The understanding of the brain will lead to cures or at least to better treatments for brain conditions and diseases such as Alzheimer's, Dementia, Schizophrenia, or Epilepsy. Also the Engineering field is receiving improvements from the brain. In the computer science field, there is no a single computer today that can do what the brain does. How does the brain store images, memories, songs, or languages? What does the brain do to recognize a face even if the image is distorted? These kind of tasks can not be solved even by the state of the art algorithms in Machine Learning and Artificial Intelligence, but our brain does these and more complex activities every day. To understand how the brain solves its tasks will help to create brain-inspired technologies to perform brain-like processing.

In recent years, brain networks have been an important field of study; functional, structural, and effective brain networks. The connectome project[1] which aims to describe all neural paths in the human brain is possibly the most important effort today in structural brain connectivity. In order to infer functional networks in the brain, different technologies exist to today such as Electroencephalography (EEG), functional magnetic resonance imaging (fMRI), and magnetoencephalogra-

[1] http://humanconnectome.org/

phy (MEG). Each one of them has their advantages and disadvantages ranging from device cost to spatial and time resolution for the acquisition of brain related activity. In this thesis, the technology of interest is MEG and the goal is the designing of a novel method for functional/effective brain connectivity. The proposed method is named Fourier Bayesian Networks or FBNs, and are based in the same theory of Bayesian networks with the difference that the relation parents-children is performed in the Fourier domain by taking advantage of the power and phase properties of the Fourier transform. In this document, it will be shown and proved that FBNs are a potential tool for network inference and structure searching algorithms and also for functional brain connectivity in MEG.

The chapters in this thesis are arranged as follows: Chapter 1 gives a review on Neuroscience and Neuroimaging covering from the beginnings of Neuroscience to the state of the art technologies in brain imaging. In Chapter 2 attention is focused on MEG which is the main study of this document. Chapter 3 gives a brief review on network theory and network metrics. In Chapter 4 some of the most common and applied source relation inference method are reviewed. Chapter 5 introduces Bayesian networks and network structure searching algorithms. Chapter 6 covers all simulations for this thesis showing results about the performance of FBNs for network structure inference. Real MEG datasets are analysed in Chapter 7 using FBNs and finally Chapter 8 discusses about FBN properties and delineates future work needed to improve the current state of FBNs.

Chapter 1

Introduction to Neuroscience and Neuroimaging

The brain is possibly the most exciting organ in the human body. It has drawn the attention of physicians, philosophers and researchers from all fields for centuries and still our current accumulated knowledge give us only a glimpse of all its secrets. The brain is in charge of analysing an astonishing amount of information coming from our five senses, and from this information it is able to recall memories to perform future and almost instant actions. Our voluntary and many involuntary movements are controlled by the brain; walking, blinking, talking and even typing a PhD thesis are controlled by this organ.

The brain stores our memories and experiences which also shape our personality. Everything we are is defined in our brain and understanding how it works is not an easy task. It requires the aid of a wide range of disciplines such as psychology, biology, engineering, mathematics, philosophy and more.

Neuroscience, as its name suggests, is the science that study the nervous system including the brain. Neuroscience comprehends such an extremely large field of knowledge that it is preferable to subdivide it according to levels of complexity, from the chemical phenomena to the study of the mind (Bear et al. , 2007). These levels are summarized as follows:

- Molecular Neuroscience; It studies the chemistry of the brain, mainly neuro-

transmitters, hormones, and drugs and how these substances influence either our behaviour or our health.

- Cellular Neuroscience; Mainly it is focused on the study of the different kind of neurons in the nervous system, how they interact with other neurons and how the nervous system develops from fetal stage to old age.

- System Neuroscience; It studies the neural circuitry and how it works to allow the performing of different tasks, like reading or walking. It is mainly focused on the study of the steps performed by the neural circuitry when executing physical or cognitive actions.

- Behavioural Neuroscience; It is interested in understanding the biological processes that leads to the behaviour of individuals, it is also known as Psychobiology.

- Cognitive Neuroscience; It studies the same field of Behavioural Neuroscience, but Cognitive Neuroscience avoids attaching the mind to biological processes only, by complementing its understanding with several other fields such as Computer Science, Information Theory, Philosophy and Mathematics. It is also commonly called 'Biology of the mind" or "Science of the mind" (Gazzaniga , 1998).

Cognitive Neuroscience is normally referred as a new school of thinking which recognizes that the understanding of an extremely complex system such as the brain, requires the expertise of multiple fields. Among these we can also find Computational Neuroscience and Neuroimaging. Computational Neuroscience aims to find mathematical models that explain the biological and behavioural processes in the brain. It studies neural membrane phenomena, such as conductive channels, membrane potentials, axonal interactions and dendrite models. It also studies the behaviour of neural networks, aiming to understand how big groups of networks communicate with each other. On the other hand, Neuroimaging groups a set of techniques and technologies designed to acquire either invasively or not, functional or structural information from the brain that helps to understand the processes that occur in it. To

achieve this, there are technologies such as electroencephalography (EEG), magnetoencephalography (MEG), magnetic resonance imaging (MRI), and its functional complement (fMRI).

The study of the brain has become in recent years a multidisciplinary field. The brain and in general the nervous system still represents a challenge for the scientific community. Its understanding will lead to better treatments for brain related diseases, and faster and more accurate clinical diagnoses.

1.1 Beginnings of Neuroscience

The brain has fascinated philosophers and scientists for centuries. Even the first hominids were aware that the brain was crucial to sustain life. The archaeological evidence dating back a million years has found hominid skulls presenting fatal damage produced by direct attack. The brain resides at the centre of the head, protected by the skull and close to all our sensorial receptors. Our ears, eyes, and nose are in the head and their importance was obvious for the first hominids.

Human skulls as old as 7000 years has been found showing signs of being bored in a process called trepanation, showing recovery after the procedure, clearly indicating that death was not desired to the subject but some kind of relief instead. The purpose of trepanation at that time is not really known, but it is speculated that trepanation may have been used to treat headaches or mental disorders by which the hole in the skull represented an escaping route for the evil spirit (Finger , 1994).

Egyptian manuscripts dating back 5000 years describe that this culture was aware of some of the symptoms of brain damage (Finger , 1994). Possibly by their observations on head concussions and its behavioural consequences. Nevertheless, for Egyptians it was the heart and not the brain where all memories were placed, and for this reason the heart was carefully preserved during mummification while the brain was extracted through the nostrils and discarded.

The view of the heart as the centre of intelligence and memories was not challenged until the times of Hippocrates (470-379 B.C), when the Greek philosophers started wondered about the function of the brain. Hippocrates stated that the brain

was not only the centre of all sensations but also the seat of intelligence (Bear et al. , 2007). This statement was contradicted by Aristotle (384-322 B.C.) who kept the current view that the heart was the centre of intelligence and the brain was a cooling device for blood. He explained that the rational temperament of some individuals was due to the large cooling capacity of the brain. Hence, in some sense, intelligence, patience, and wisdom were indeed related to the brain size and its cooling capacity.

Hippocrates also aimed to explain brain related diseases such as epilepsy. He proposed that an interruption of the blood flow into the brain was the cause of this disease, leading to the inability of the brain of taking the vital principle from the blood and causing convulsions and loss of consciousness (Bear et al. , 2007).

An important milestone in Neuroscience was achieved by Galen (130-200 A.D.), considered the most important figure of Roman medicine. Galen shared the Hippocratic thinking of the brain function and most of his studies were influenced by anatomical observations. As the physician of gladiators, Galen had the opportunity of studying the symptoms caused by spinal and brain injuries, and complemented his studies using sheep as subjects. In his research work, Galen tried to infer brain functionality. He first noticed that the brain can be divided in two areas, being these the cerebrum and the cerebellum. By touching a fresh brain, Galen found that the cerebrum was soft while the cerebellum was harder. Using his observations and his experience as physician, Galen correctly inferred that the cerebellum was in charge of muscle control while the cerebrum manages memories (Bear et al. , 2007). His hypothesis was very simple, since memories always change and must be stored in some way in the brain, the storing must occur in the more moldable cerebrum, while the locomotion knowledge which is a more fixed skill, had to be placed in the harder part of the brain, the cerebellum.

Galen's studies remained prominent for almost 1500 years until new theories were proposed during the Renaissance. French inventions like hydraulically controlled machines, brought new ideas into the mind of the French philosopher Descartes (1596-1650). He theorized that the brain ventricles with their enclosed liquid (cerebrospinal fluid or CSF) worked as a hydraulic machine to control the muscles of the human body, a principle called "the automaton". The automaton, shown in

Figure 1.1, was the description of the human body as a machine and this model explained most of the movement abilities in mammals, but it did not explain the main ability that makes humans different, the mind and voluntary behaviour.

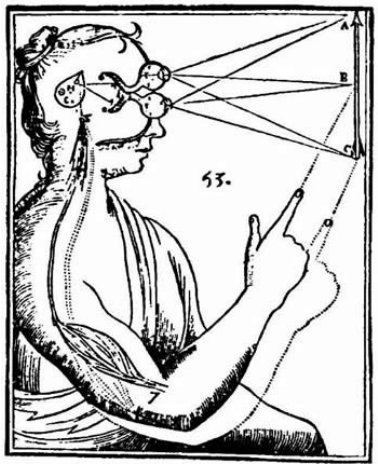

Figure 1.1: Descartes' model for the nervous system. Descartes thought that the nervous system worked as pumping pipes, a hydraulic automaton.

To complete his model, Descartes proposed that the soul or spirit was placed inside the ventricles and that the voluntary behaviour occurred thanks to interactions between the soul and the automaton, managed by the pineal gland (Finger , 1994; Bear et al. , 2007). Descartes chose the pineal gland for two reasons, first because this gland is unitary in the brain anatomy (most of the regions of the brain come in pairs, left and right) and second and most important because this gland is surrounded by CSF. Galen also made these observations about the pineal gland, he thought that the gland regulated the flow of spirits through the body. Nevertheless he concluded that this idea was ridiculous (Finger , 1994; Bear et al. , 2007).

1.2 Brain cortex

The first neuroscientists studied the brain by identifying its structures, which were catalogued and named according to the way they looked or the purpose they thought they had. The brain surface is called "cortex" or brain cortex. It is where most of the brain neural cells are allocated. The cortex is mainly a sheet of neuronal tissue folded several times in an attempt to maximize the available space inside the

skull. This cortical folding produces bumps and grooves in the brain cortex which are common in all humans and are commonly taken as markers for anatomical localization. The bumps are called gyri (singular gyrus) and the grooves are called either sulci (singular sulcus) or fissures if they are bigger. Two thirds of the brain cortex is found in the sulci.

Fissures are common in all humans, and as mentioned previously these are used to divide anatomically the brain. The right and left hemisphere are divided by the longitudinal fissure that runs from the frontal lobe to the occipital lobe. Furthermore, each hemisphere is subdivided in lobules; the lateral lobule (left and right), frontal lobule, parietal lobule and the occipital lobule. see Figure 1.2.

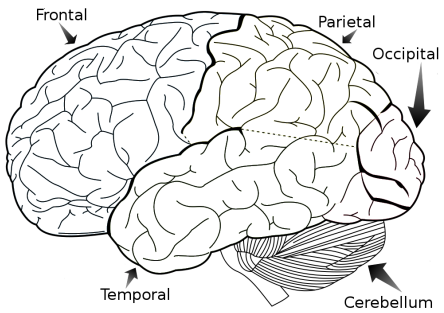

Figure 1.2: Lateral view of the brain, left hemisphere. The brain is divided mainly in four paired lobes: frontal, parietal, occipital, and temporal. The cerebellum shown at the bottom, is a structure in charge of control and learning of movement.

A highly important internal structure is the Corpus Callosum. This structure is a massive bundle of neuronal connections that connects the left and right hemispheres. Hence, the Corpus Callosum is the main communication channel between both brain hemispheres.

There are plenty more substructures in the brain, all of them equally important. Unfortunately in this thesis it is not possible to mention all of them, but a complete review on brain's anatomy can be found in Bear et al. (2007).

1.2.1 Functional localization

At the beginning of the nineteenth century there was the idea that the nervous system could be divided by functions. Franz Joseph Gall (1758-1828) was a pioneer

researcher in localization of mental functions and his vision certainly changed the study of the human brain. Gall was a very talented physician and anatomist, and his work led him to formulate the idea that the development of specific cortical regions were a result or at least an indicator of special abilities in individuals. That meant that people's abilities were correlated with the development of certain brain areas.

Gall started his research studying skulls and he hypothesized that bumps on the skull surface explained special characteristics on individuals like intelligence, creativity, memory or even child caring. For example, a prominent front was an indicative of intelligence while the contrary the lack of it. Also, a flat nape according to Gall's theory was an indicative of laziness, and this was produced by putting hands on the nape while resting. Following his ideas, Gall collected skulls and casts of skulls from different individuals with opposing backgrounds, some of them were writers, poets, and statesmen and others were lunatics and criminals (Finger , 1994). With these Gall attempted to correlate their skull shapes with the lives they had.

Figure 1.3: American Phrenological Journal front. At the beginning of the 19th century, Gall's Phrenology became an influential school of thought.

Gall started giving public lectures about his theory amazing the public and at that moment a new science was born. It was called Phrenology, a science specialized in cortical localization. Suddenly, Phrenology expanded through United States and

Europe and Gall gathered more than 300 skulls and 120 casts for his studies (Finger , 1994). There were also Phrenology societies in the United States and England spreading the word of the new science. But in France some researchers resisted the enchantment of Phrenology. One of them was Marie-Jean-Pierre Flourens, a researcher with an extremely good reputation in France, where he published many papers on Comparative Anatomy, Anaesthesiology, Embryology and Physiology (Finger , 1994). Flourens claimed that it was impossible to delimit cortical areas by measuring the skull and correlating its shape with human capacities. Flourens became the most outspoken critic of Gall's theory.

Flourens was a true believer of laboratory work in research, and developed protocols for the use of animals in ablation and stimulation of cortical areas. Ablation in Neuroscience is a technique that destroys tissue of a specific area and sees how the specimen's behaviour is affected. From his experiments, Flourens theorized that it was impossible to define cortical localization and inferred that the cerebral cortex worked as a whole and not in specialized areas, a theory that he called "Equipotentiality". Nowadays, historians consider that Flourens was so blinded by his hate of Phrenology that he did not realize that some of his work confirmed functional cortical division (Finger , 1994). In many of his ablation experiments, Flourens found a recovering of the specimen after some time, and for him this confirmed his theory about equipotentiality of the cerebral cortex.

Nevertheless, Gall and Flourens together are considered today as the fathers of cortical localization. Gall was a visionary with a great theory but the wrong method, and Flourens was a great experimentalist with the correct method but the wrong answer.

1.2.2 On the search of language, Broca's area

By the middle of the nineteenth century, the strongest accepted theory for functionality of the cerebral cortex was the "equipotential cortex" mainly advocated by Flourens in France, although the ghost of Phrenology still wandered in the corridors of academic societies. In 1861, the French physician Alexandre Ernest Aubertin, a convinced follower of the localization theory (although not by Phrenologist's meth-

ods), presented a case of a patient who attempted suicide by shooting the front of his head. The results of this was a cracked frontal bone which exposed his frontal lobes. During the medical intervention, the patient was interrogated while placing the blade of a spatula on his frontal lobes. By means of light pressure on the frontal lobes, Aubertin was able to induce Aphasia. The patient lost his ability to speak and this returned when the pressure was not delivered (Finger , 1994; Bear et al. , 2007). Other scientists argued that the aphasia phenomenon occurred because Aubertin also transmitted the pressure to other areas of the brain, but Aubertin argued back saying that the spatula was located in such a way that just the frontal lobes were affected. After his finding, Aubertin studied several patients with frontal damage and claimed that the ability of speech was in some area of the frontal lobe. He also hypothesized that the cases where this did not occur were because the area of articulated speech was not established and also in the cases of unilateral damage, the speech ability can be compensated by the other side of the brain (Finger , 1994), a property known today as "brain plasticity".

Aubertin also stated that he would renounce his idea of language localization, if he found contrary evidence in an aphasic patient that he had studied for a long time and was close to death. One of the scientists who attended Aubertin's lectures was Paul Broca.

In the same year 1861, a patient named Monsieur Leborgne, was referred to Broca's surgical service. Leborgne was known by other patients as "tan" or "tan-tan", because this word was the only thing he could say, besides some swearing. Broca invited Aubertin to analyse his patient "tan" and after an inspection Aubertin concluded that the aphasia had its source in the frontal lobe (Finger , 1994).

Six days later Leborgne passed away and Broca presented his case and his brain to the Society of Anthropology in France the next day. Four months later Broca submitted a full report to the Society of Anatomy in France and presented a strong support that the damaged area was involved in articulated language. In his report Broca congratulated Aubertin for gathering a considerable amount of evidence about the role of the frontal lobe in speech and from that moment the case of the patient "tan" revolutionized the ideas of cortical localization. Finally, the area that Broca

found in the frontal lobe of his patient is known today as Broca's area.

1.2.3 Wernicke's area

In 1874 a German neurologist Karl Wernicke reported finding an area on the left temporal hemisphere that also produced aphasia, but this area was different from Broca's area. The aphasia produced by damage in Broca's area is manifested in disruption of speech but comprehension or coherence of what is said is complete. The aphasia that Wernicke reported was contrary of the one reported by Broca. The patients were very talkative, their speech was fluent but there was not coherence in their conversations (Gazzaniga , 1998).

1.2.4 The Wernicke-Geschwind model

Just after the discovery of Wernicke's area, Wernicke proposed a model for language processing in the brain. This model was extended later by Norman Geschwind and it is known today as the Wernicke-Geschwind model. Wernicke's model was the first attempt to propose a neural communication path among different regions of the brain, extending the concept of cortical functional localization by suggesting interaction among the brain areas.

The elements that integrate Wernicke-Geschwind model are Broca's area, Wernicke's area, the arcuate fasciculus (a group of axons that connect both areas), the angular gyrus, the motor cortex, and the auditory cortex as shown in Figure 1.4.

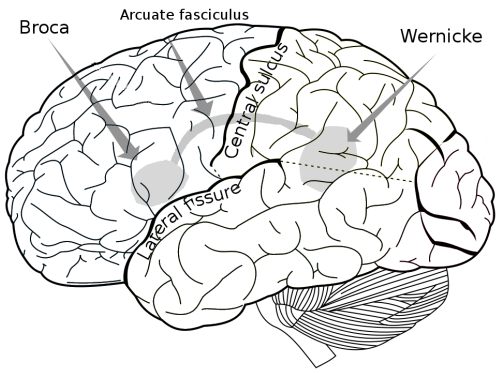

Figure 1.4: The Broca's and Wernicke's areas are involved in the communication process.

The model explains the communication process for two cases; the repetition of a spoken word and the reading of a written word. In the first case the word must be listened to by the auditory cortex and it does not have any meaning until it arrives at Wernicke's area where the word is analysed. Then, the information travels through the arcuate fasciculus to Broca's area which does the necessary coding of information. This code is sent to the motor cortex which will move the necessary muscles, tongue, larynx, and so on for the expression of the word.

The second case, the reading of a written word, is very similar but it starts in the visual cortex. The visual patterns are processed in the visual cortex and the information is sent to the angular gyrus, it is thought that the angular gyrus transforms the read word into a code similar to that of a heard word, and then this code is sent to Wernicke's area to continue the language processing path (Gazzaniga , 1998).

By the ending of the 19th century, the theory of brain functionality where cortical regions were assumed to be in charge of one or maybe more tasks was well accepted. The introduction of new microscope techniques in Neuroscience fuelled the discovery of the brain's construction and the cells that compose it.

1.3 Cytoarchitecture of the cerebral cortex

The brain of an adult consists of approximately 100 billion neurons and a larger number of glial cells, the supporting cells of the brain (Abeles , 1991). Brain neurons are arranged in an extremely intricate manner that may seem to be a chaotic universe of cells and neuronal connections. But there is no chaos in the brain, every single connection or synapse (which can be numbered in hundreds of trillions) has a purpose of existence, such as the neuron where it comes from. In this section we review neuron cells, their functions, and some of their properties.

1.3.1 The neuron

Although the microscope was invented long before Broca's time, it was not possible to see brain cells through it. The reason was because in order to see cells on the

microscope, very thin slices of tissue are needed, at least as thin as the size of the cell. The brain is a very soft tissue more or less having the consistency of gelatin dessert. Hence, it was impossible for neuroscientist to obtain such thin slices without disturbing the tissue. It was not until neuroscientists learnt how to harden the brain by immersing the tissue on formaldehyde and developed a special device to obtain thin slices called microtome, which works similar to meat-slicers used in butcher shops, that the neural world inside the brain was discovered.

Anyway, even when neuroscientists could harden the brain, problems remained. The reason was that fresh hard brain tissue is completely monochrome. It has monotone pink colour from which no cell can be seen. It was not until researchers were able to discover stains that have the property to attach their pigments to specific parts of the tissue but not all, that neurons appeared in the microscope. One of these stains was Nissl stain which has the property of staining only the soma of some neurons. The soma is in some sense the body of the neuron. This was the first time where the brain cells could be physically seen and also this allowed neuroscientists to catalog brain tissues by their cellular construction, also called cytoarchitecture.

In 1873, the Italian histologist Camillo Golgi (1843-1926) discovered that by immersing brain tissue in silver chromate solution, neurons were pigmented almost entirely, allowing to see in the microscope two principal parts of the neuron; the soma which is the body of the neuron and a series of thin tubes that radiates from the soma called neuritas, which are divided in two types, axons and dendrites. Neurologists of that time were able to infer that axons work like wires from which the messages are sent to other neurons or tissues. Also, they inferred that dendrites, which are shorter reaching only the vicinity of the soma, work as antennae that receive messages from other neurons.

By his work, Golgi inferred that neuritas were fused together to form a continuous reticulum or a network, in a similar way to the veins and arteries in the circulatory system. Hence, Golgi claimed that the neural network in the brain was an exception of the cell theory, which states that cells are the minimal unit of all biological tissues. A contemporary histologist was Santiago Ramon y Cajal (1852-1934) who learned

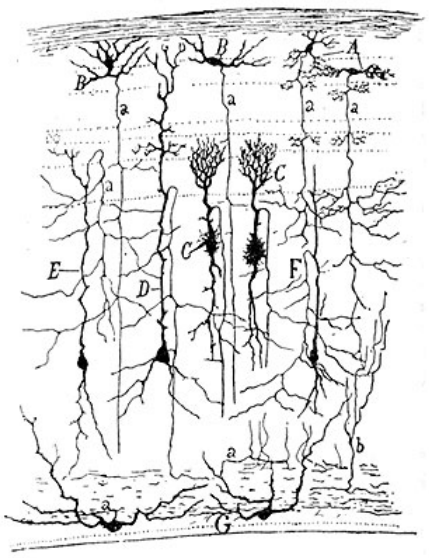

Figure 1.5: One of original drawings of Ramon y Cajal. A section of the optic tectum of a sparrow, Madrid 1905.

Golgi's method around 1888. Surprinsingly, Cajal arrived at a completely different conclusion. Cajal claimed that neuritas of different neurons were not fused but joined by contact. Cajal's findings led to the neuron doctrine, which places the neuron as an individual processing unit which connects with other neurons forming the neural network.

The brain is formed mainly of two kinds of cells, neurons and glia (or glial cells). It is thought that the glial cells contribute to the nourishing, support, and isolation of the neurons and that these cells do not perform any information processing. The glial cells represent 90% of the brain mass while the neurons represent the remaining 10%. Hence, the research community focuses its attention on a minimal portion of the brain. Nevertheless, some researchers are working on glial cells in order to discover if these important cells also contribute to the cognitive processes in the brain (Bear et al. , 2007).

It is assumed that neurons perform all the signal processing in the nervous system. There are different kind of neurons depending on their specific function. Although their functions are not fully understood, some neurons are dedicated to sense our environment by our ears, eyes, nose, and our skin. Others are dedicated to feel pain, if something is hot or cold or even spicy thanks to neuronal terminals in our

tongue. The muscles are also controlled by specialized neurons that send signals projecting their axons into muscle fibers, also known as innervation.

Neurons, as all eukariotic cells, encloses in its interior organelles that contribute to the cell functions; one nucleus, Golgi apparatus, endoplasmic reticulum (ER, rough and smooth), mitochondrias and ribosomes. From the inside there is little difference between a neuron and any other cell in our body. The main characteristics that make neurons unique are their cellular membrane, their peculiar shape, the proteins embedded in their membrane, and an important number of mitochondrias which are the fuel store of all cells, indicating that neurons consume important amounts of energy. Is not a surprise then, that the brain consumes 20% of the calories we intake but it represents only 5% of our body mass (Lauglin , 2001).

Figure 1.6 shows a typical neuron, its body is divided in three sections: the soma, dendrites, and the axon. The soma is the body of the cell, it is where all the organelles are placed and where the cell machinery works to allow the neuron to perform all its functions. The dendrites are considered as a kind of antenna where signals coming from other neurons are caught. Finally, the axon has the function of sending electrical signals processed by the neuron to other neurons or muscles. For this reason the axon can be as long as a meter projecting from the backbone to the finger tips.

The most common way to catalog neurons is by their number of neurites (axons and dendrites). If the neuron has only one neurite (which should be an axon) it is called unipolar, two neurites is called bipolar, and if there are three or more neurites the neuron is called multipolar (Bear et al. , 2007). From the multipolar category there are two types of multipolar neurons in the cerebral cortex, these are stellate cells (star shaped) and pyramidal cells (pyramid shaped body). The stellate neuron has a more local radius of influence in the cerebral cortex, it projects its axon to local neurons and can receive signals from either neighbour neurons or neurons placed far away in the nervous system. Pyramidal neurons also receive signals from local and far away neurons, but their main characteristic are their long axons.

The axon starts in the axon hillock and finishes in the axon's terminals which synapse onto other neurons. The axon is in charge of transporting the signal to

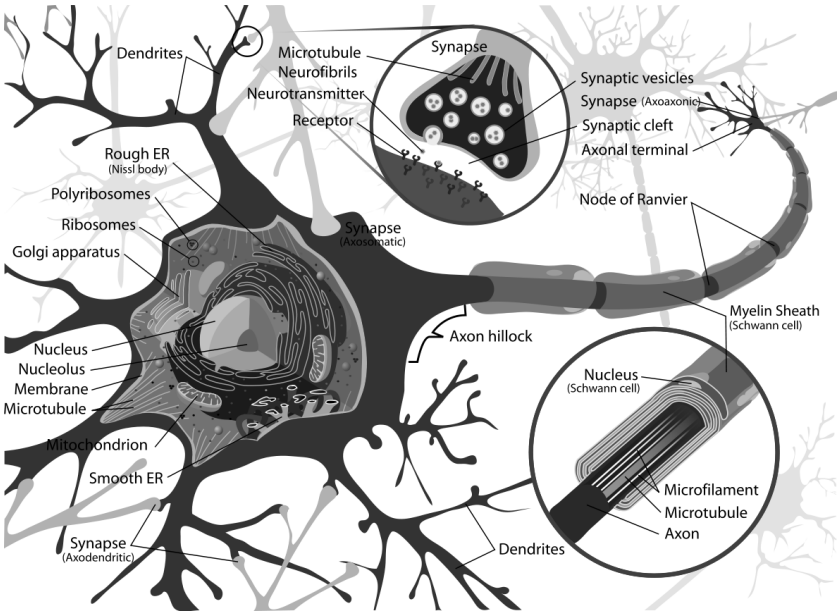

Figure 1.6: The neuron. The neuron is not very different to any other cell in the human body. It has a nucleous, a nucleolus and all the organelles. What makes the neuron unique are its membrane and its irregular shape. The figure shows a pyramidal neuron and the neuritas: axons and dendrites which connect to other neurons (synapses). The myelin sheaths that surround the axon work as insulators that improve the transmission of action potentials. Image from http://en.wikipedia.org/wiki/Neuron

other neurons thanks to unique characteristics of the neural membrane. The neural membrane as other cellular membranes is made of phospholipids (fatty material) (Bear et al. , 2007), a molecule formed by a polar "head" containing phosphate and a nonpolar "tail" containing hydrocarbon. An important characteristic that makes the cellular membrane so efficient is the phosphate head of the molecule that is hydrophilic, that means that it does not repel water and the hydrocarbon "tail" is hydrophobic, meaning that it repels water. The cellular membrane is formed of two layers of phospholipids (phospholipid bilayer), with the hydrophobic tail in the inner section and the hydrophilic heads in the outer section, creating an effective layer that isolates the cytosol of the cell from the outside environment.

Embedded in the neural membrane are specialized amino acid proteins that work as channels for the crossing of particles and ions through the neural membrane. The ion channels are carefully folded proteins arranged as if they were a barrel with a hole in the centre where the ion particles cross. It is thought that the folding of these proteins is performed inside the Golgi apparatus just after the transcription

of the mRNA into protein by the rough ER, then the new ion channel travels to the membrane transported by a vesicle (Bear et al. , 2007).

All the electrical features of neurons are in the ion channels and the cellular membrane. The electric current that neurons produce can be compared to the one generated by a battery. Ions with different electrical charge are present inside and outside the neural membrane. The ionic concentration difference produces a differential potential between both sides of the cellular membrane. The main ions involved in the neural battery are Sodium (Na^+), Potassium (K^+), Calcium (Ca^{2+}), and Chloride (Cl^-), the first three have positive charge and Chloride is negative as it is indicated. However, the potential is due to the difference of the ionic concentration, for example K^+ is 20 times more concentrated in the inside than in the outside of the cell providing a difference potential of -80 mV according to Nernst equation (Bear et al. , 2007)

$$E = \frac{RT}{zF} \ln \frac{concentration\ of\ ions\ outside\ cell}{concentration\ of\ ions\ inside\ cell} , \tag{1.1}$$

where R is the universal gas constant, T is the absolute temperature, z is the number of electrons, and F is the Faraday constant. With different concentrations of ions in the neural environment, the cell membrane presents a differential potential of approximately -65mV which is commonly called the "resting potential". The resting potential is the voltage that can be measured at the membrane in its resting state, which means that the neuron is neither receiving signals from other neurons nor sending information through its axon. The information received or sent among neurons is manifested in a voltage change, also called action potential (AP). This AP is a change in the resting potential from -65mV up to 40mV due to changes in the neural membrane permeability to ions. In the cerebral cortex, the changes in the membrane permeability are mainly induced by the arrival of APs from other neurons. The ionic principle of the membrane potential gives the AP the property of travelling through the membrane. The AP travels on the neuron's membrane thanks to changes in local ion channels. These changes produce a chain reaction in neighbouring channels and the AP is then transmitted to the neighbouring areas of the membrane.

When the AP arrives at the axon hillock it is accelerated thanks to the oligodendroglial cells and the nodes of Ranvier. The oligodendroglian cells are cells with an unusual sheath shape. They literally wrap the axon with the myelinated sheath in several layers and work as an isolator for the axon "cable", and the nodes of Ranvier are membrane gaps left by the myelin sheath. Both can be seen in Figure 1.6. The myelin wrapping avoids current leakages across the wrapped section and the spike is transmitted to the next node of Ranvier electrically by the differential potential, accelerating the spike transmission. The axon terminal might divide itself to synapse onto several neurons at the same time. Commonly, the axon terminals synapse a spine in a dendrite of another neuron.

Now that we have seen just a part of the process of neural transmission, we can imagine the complexity of a network of 100 billion of neurons each one receiving an average of 10,000 synapses approximately in the adult brain. Although complex, the arrangement of neurons in the cerebral cortex obeys some detectable patterns in delimited cortical areas. According to the theory of functional localization in the cortex, the differences among cortical regions are because these regions are specialized to perform specific processing tasks.

1.3.2 Brodmann's map

In 1909 Korbinian Brodmann identified 52 different regions in the human cerebral cortex. Brodmann categorized these areas by analysing the neural cytoarchitecture, this means the variations in the proportion of cell types within the layers of the brain cortex. The map he discovered is known today as Brodmann's areas or Brodmann's map. Brodmann guessed (but could not prove) that different neural cytoarchitectures have a specific functionality attached to them (Bear et al. , 2007; Gazzaniga , 1998).

Today research work has proved that indeed this is true. For instance, area 17 is called the visual cortex because this area receives axons from the thalamus which is connected to the retina. Area 4 is called the motor cortex because neurons in this area project axons to the spinal cord which in turn manages the contracting of the body muscles. Figure 1.7 shows the regions found by Brodmann.

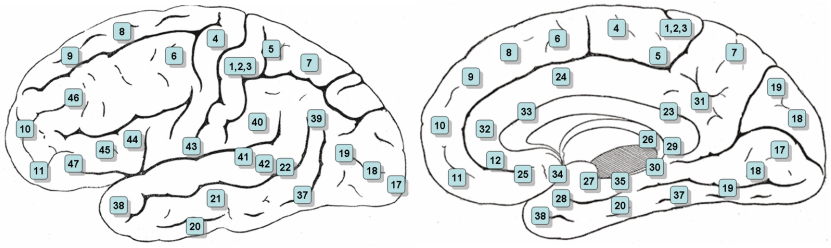

Figure 1.7: Brodmann's map or areas were defined by Korbinian Brodmann in 1909. By using Nissl stain Brodmann was able to identify different neural cytoarchitectures in the cortex which he labelled in 52 regions. Gray's Anatomy, 1918.

One of the questions that has intrigued researchers since the findings of Brodmann is how the human brain evolved such a complex architecture. Brodmann suggested that the human cortex evolved by adding areas to perform abilities needed by the individual. This was partially confirmed by the analysis of cortical cytoarchitectures in other mammals like the cat or the rat whose brains have fewer areas. Furthermore, there is little difference between the thickness of the human cortex compared with other mammals, suggesting that evolution of the human brain was by increasing the size of the cortical sheath and the number of areas (Bear et al. , 2007).

1.4 Cortical maps

Due to the complexity of the cerebral cortex and the nervous system, the study of the functional anatomy of the brain results in a great challenge for researchers. Discovering the secrets of the neural network has been the work of thousands of researchers around the world. How are the networks formed? How do neurons synchronize? How and where is memory stored?, These are just some of the questions that still remain unanswered.

To study the brain and its functional areas, different research paths have been followed. Because the brain structure of mammals is approximately similar, some researchers use cats, dogs, monkeys or rats in their research experiments and extrapolate their results to the human brain. The brains of lower mammals are simpler

than the human so its study is also easier. Furthermore many experiments carried out using drugs and ablation procedures can be performed in animals but not in humans. Experiments in mammals provide insights about how the cortical areas of the brain interact in different controlled scenarios, but the study of cortical areas do not provide knowledge of how the neurons work at a cellular level. In order to tackle this question some researchers prefer to work using invertebrates such as the squid or the sea slug *Aplysia Californica* (Bear et al. , 2007). The reason why using such a different animal like a slug to infer processes in the human nervous system is because all living things in this planet have a common ancestor. That means that humans share DNA with any other animal. For example, the yeast cell *Saccharomyces Cervisae* shares 20% of its DNA with humans and this organism is widely used to study cellular processes also present in human cells. The same analogy is applied to the sea slug and its nervous system.

There are also studies in humans which are noninvasive using neuroimaging technologies such as electroencephalography (EEG), magnetoencephalography (MEG), and functional magnetic resonance imaging (fMRI). These kind of technologies acquire information which is correlated to brain activity and also have the capacity to localize its position. This property makes neuroimaging techniques extremely valuable in brain studies. On the contrary, trascranial magnetic stimulation or TMS induces a current in the brain by producing a magnetic field focussed on a specific cortical region, which is also valuable for functional localization studies. A similar approach but using direct electrical stimulation led to the discovery of the sensory-motor map in the cortex.

The sensory-motor cortex

Long before Broadmann's map, it was already known that the nervous system communicates using electrical currents. In 1791 the Italian physician Luigi Galvani discovered that frog's legs twitched when struck by an electrical spark, starting a new field known today as Bioelectricity, which studies the electrical properties of the nervous system and muscles. In 1809 an Italian anatomist, Luigi Rolando, used Galvanic currents to stimulate the brain cortex, and in 1937 John Zachary Young

suggested that the squid giant axon could be used to study the electrical properties of nerve cells, a suggestion that finally ended in the ground breaking work of Alan Lloyd Hodgkin and Andrew Huxley who in 1952 created a mathematical model that describes the ionic mechanisms for the axon potential propagation in the giant squid.

In the field of cortical functionality, pioneering work was achieved by Wilder Penfield in the 1940s. Penfield was carrying out operations to extract cortical tissue from the brain of epileptic patients. When the epileptic source can be localized in the cerebral cortex, its extraction commonly leads to the cure of the disease, and the stopping of epileptic seizures. Taking advantage of the exposed patient's brain during the medical procedure, Penfield explored the effects of small levels of electrical stimulation on several areas of the cerebral cortex. Since the brain lacks pain receptors, the patient could not feel any discomfort allowing Penfield to record the patient's experiences (Finger , 1994).

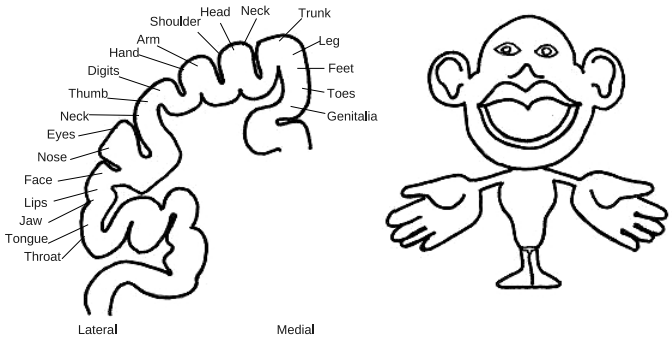

Figure 1.8: The sensory-motor cortex (left) shows the correspondence between cortical areas and the body areas. The homunculus (right) represents broadly how our brain sees our body. The size of the brain regions is proportional to the size of the body parts. Image adapted from Purves D. et al. editors (2004).

For many years Penfield and colleagues experimented with the effects of cortical stimulation in their patients and most of their work was focused on the area of the central sulcus. By stimulating the precentral gyrus, they noticed that body movement was produced and that the type of movement was directly related to the stimulated area. The same phenomenon happened in the postcentral gyrus but instead of movement, the stimulation activated different sensations, like light touch, or itch. This correspondence between the cortical areas of the sensory-motor cerebral cortex with the body areas can be represented as a homunculus as shown

in Figure 1.8.

Language of the brain

Brain imaging techniques have been used to test hypotheses about the brain and its mysteries. For instance, language is a characteristic apparently unique in humans, which have a very complex communication system compared to other mammals. The language neural path in the human brain might be discovered or proposed with the Wernicke-Geschwind model in Section 1.2.4. But this model does not explain many of the phenomena involved in the process of language. The model can explain the main types of aphasia; Broca's aphasia, Wernicke's aphasia and conduction aphasia, which occurs when the arcuate fasciculus (the path between Wernicke's and Broca's area) is damaged. But it can not explain many other kind of specific symptoms and cases. How do we choose words? How do we make relations among them? Where are words stored? and many other questions that remain unanswered.

Figure 1.9 shows a theoretical model for language processing (Gazzaniga , 1998). At the centre of this model we can find the lexicon. The lexicon is the pool of words which are stored somewhere in our brain. It is theorized that the brain creates networks among all our stored words and these networks are always changing due to our continuous learning and forgetting of words during our life. For example, the word "car" can be related to "truck" but not to "lemon", and "car" and "truck" might be in a subnetwork called "vehicles". Also there might be a link between the word "firetruck" in the network of vehicles and the word "red" in the network of colours because both words are related (Gazzaniga , 1998).

Motor Learning

Motor control is maybe the most studied area in Neuroscience. The mechanisms of movement in our backbone and limbs are better understood than the ones in our brain. This is mainly because for many years researchers have performed experiments in animals to discover the circuits for movement. Today, we know that special neural circuits in our spinal cord communicate the commands of the brain to the limbs. Special neurons called *alpha motor neurons* send their axons to all the

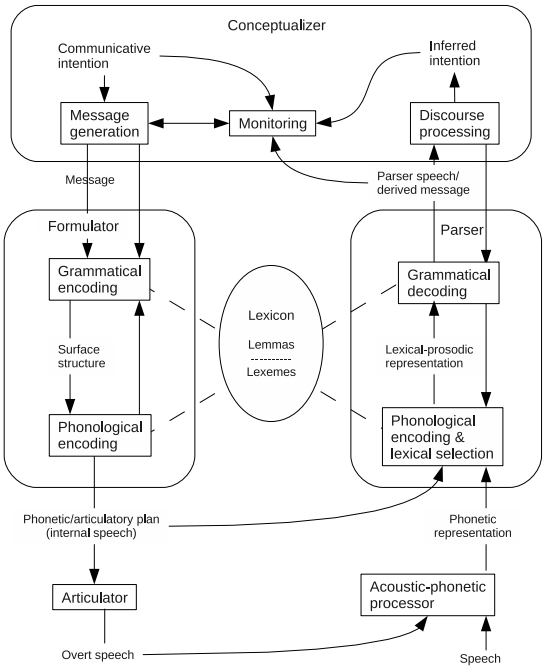

Figure 1.9: Theoretical model for language processing and mechanism of language. This map represents the model for language creation and use. Image adapted from Gazzaniga (1998).

muscles of our body leading to their contraction. Also, the motor neurons placed in the spinal cord can perform neural processing to activate movement without a direct command from the brain, these movements are called reflexes (Gazzaniga , 1998).

The areas of the brain involved in motor control are known, but an important issue is how these areas interact to launch movement. Fortunately the human being is not a machine, the discovery of the areas involved in motor control is not enough to understand the phenomenon. Besides movements, humans also make decisions, plans and even imagine a movement before it starts. Only reflexes are done unconsciously (that is why reflexes are better understood) and thus planned movement requires more research. Figure 1.10 shows the functional hypothesis of how the different motor areas of the brain interact, but this is just a consensus. Future work in brain mapping for control of movement might change or add more areas to the network in Figure 1.10.

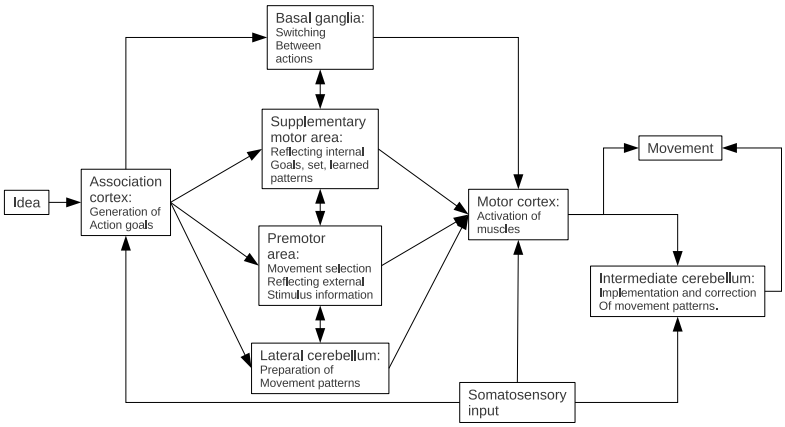

Figure 1.10: Functional hypothesis for movement decisions. The diagram shows the brain structures theoretically linked to perform skilled movement. Image adapted from Gazzaniga (1998).

Memory and Learning

Memory is the holy grail of Neuroscience. The place of memory in the human brain is still unsolved, but some of the memory phenomena are understood today. Currently, it is known that the temporal lobes are an important structure for memory and this fact was discovered almost by accident (as it often happens in science) in a patient who had serious epileptic seizures. His name was Henry Gustav Molaison (1926-2008), better known in the scientific community as H.M. In order to cure his epilepsy he had an operation where 8 cm of both temporal lobes were excised, including the amygdala and two thirds of the hippocampus. The surgeon succeeded in alleviating H.M's seizures (Bear et al. , 2007).

Everything looked fine in H.M., but the removal of so much temporal tissue left H.M. with retrograde amnesia of many years before the operation, that means that he could not remember anything in a period of years before the surgery. But that was just a minor problem in H.M.. The surgery also left him with a serious anterograde amnesia, meaning that he was unable to remember new experiences. H.M. could not even remember someone he met minutes ago. An interesting phenomenon also happened in H.M, he was able to learn new motor tasks. For instance, he was asked to draw by looking at his hand through a mirror, a task that needs practice. The odd thing is that after many trials to gain experience he could accomplish the task

successfully but he could not remember any of his previous trials. For H.M. a new trial was the first time.

At the cellular level much of the insight in learning has been provided by experiments in the sea slug Aplysia Californica. In a series of experiments in the 1960s, Eric Kandel worked on how memory was formed in this invertebrate animal. When a jet of water is squirted on the siphon (a fleshy exposed section) of the aplysia, its gill retracts. This reflex is called gill-withdrawal reflex. Kandel experimented habituation of aplysia to the sensation of the water jet until the gill stopped contracting. By measuring the electrical impulses in the neurons of aplysia, it was found that the presynaptic neuron that senses the skin continues sending the signal with the same intensity but the postsynaptic neuron that innervate the gill muscle fired with lower intensity. The reason of this was that the axon of the presynaptic neuron changed by releasing less neurotransmitters. Hence, the habituation learning for the gill-withdrawal reflex was explained by a presynaptic modification (Bear et al. , 2007). Eric Kandel was awarded the Nobel prize in 2004 for his research work.

Certainly the mapping of the brain and its correlation with functional and cognitive abilities in humans will continue revealing the secrets of our nervous system. Nowadays the acceleration of technologies provides Neuroscience with better tools to study the human nervous system, and has provided new ways to infer the networks that form it.

1.5 Networks in the brain

As mentioned in Section 1.3.1, it is estimated that the human brain has 100 billion neurons, each one receiving synapses from 10,000 other neurons on average. This creates a massive neural network that defines all what we are. Definitely a single cell such as a neuron is not capable of thinking, saving memories, recognizing patterns, or reading this thesis. But when billions of them communicate they create a much more complex and dynamic system which performs these and more cognitive tasks.

It is known that brain neurons create networks or neural patterns and clusters to perform different tasks. In the brain cortex there are regions designed to perform

information processing, and none of these regions works alone, each one of them interacts with several others.

It is the desire of researchers to understand the interaction of the different brain regions either at the cellular level (micro scale) and at the regional level (macro scale). These two views of network neuroscience, try to understand the cellular network including the interaction of large areas of the brain cortex (Sporns et al. , 2004). Both views, the micro and macro scales, are equally important and in fact both are necessary to understand the brain and to decrease the current gap that exists between the biological brain and the human mind.

Researchers agree that the understanding of the neuronal processes can also be achieved by studying the network patterns in the brain. Network theory focuses on the shape of networks to define the properties of the systems. In the case of a network the interest is focused on its shape and the interaction among its elements.

1.5.1 Small-world network

A network is formed by agents called "nodes" or "vertices" and connections normally called "edges", which are the lines that connect the nodes. Networks are part of our daily life, for instance the Internet, the electrical power grids, or the city subway. Also our friends create a network of human relations. The social networks are possibly the most cited examples in network theory today from where the concept of "six degrees of separation" started. The six degrees of separation concept states that in the world-wide network of friends and acquaintances, every human is separated an average of six people from any other person in the world. This "friend of a friend" network was called by the first researchers in the field as small-world network.

Small-world networks were first quantitatively described by Watts and Strogatz (1998). In their paper, Watts and Strogatz took a regular graph in which each node was connected to its k nearest neighbours, then the network edges were reconnected randomly with a low probability p. Regular networks have as common characteristic a high clustering coefficient C, meaning that the nodes in this network are mostly connected to their nearest neighbours. Furthermore, these networks present a high path length L or in other words, several steps are needed through the network edges

to arrive from one node to any other. On the other hand, random networks which are created by connecting nodes randomly, present low clustering C and low path length L. Small-world networks are somewhere in between, having high C but low L.

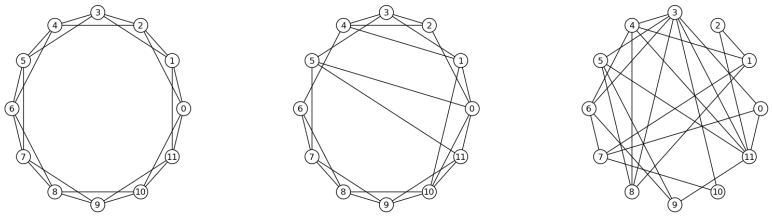

Figure 1.11: Randomness in a network. The figure shows a lattice (left), a small-world (centre), and a random (right) networks. A small-world network can be created by reconnecting pseudorandomly edges from a lattice network. Image generated using Python module: networkx.

Figure 1.11 shows an example of small-world network generation. Here, the network at the left is a regular circular network of 10 nodes where each node is connected to its 4 nearest neighbours. The network in the middle has the same arrangement as the regular network but a second step of rewiring was implemented. Every edge was rewired randomly with a probability $p = 0.15$ to create a small-world topology. Finally, the network at the right is a random network. In order to create this network, all edges from the regular network were rewired randomly with probability $p = 1$.

Interestingly, the small-world phenomenon has been found in all kinds of systems, mainly biological. The first and only nervous system which has been completely described is the Caenorhabditis elegans' system, a nematode of approximately 1mm long which has been used for decades as a model organism in biology, mainly because of its simplicity. C. elegans' nervous system is exactly described by 282 neurons having 2462 synaptic connections (Basset and Bullmore , 2006) and it was proved to be a small-world network.

The small-world topology presents interesting properties. For instance it is said to be high synchronizable (Uhlhaas et al. , 2009), meaning that nodes in small-world systems tend to be synchronized. These kind of networks have been also noted as the optimal trade off between high communication among network nodes and wiring

cost, which is a natural behaviour of a system with high need of connectivity having limited resources, such as the nervous system (Bullmore and Sporns , 2009).

1.5.2 A small world in the brain

The search path for the brain network has basically two main branches; the structural network and the functional network (Sporns and Honey , 2006). The structural network represents all physical connections among neurons in the brain. Hence the structural network is mostly defined by the axons that connect brain neurons. The functional network, as we have previously said, represents the interaction among several areas of the brain when performing a cognitive task or no task at all (this is called resting state network, see (Raichle et al. , 2001)). Furthermore, there is a third kind of network called causal or effective network, which commonly has the same structure than the functional network but arrows are added instead of undirected edges to emphasize cause and effect during information flow.

Several researches have pointed out the importance of the brain network study (structural, functional, and effective) see for instance Bullmore and Sporns (2009); Rubinov and Sporns (2009). Sporns et al. (2007) studied several network metrics on the macaque brain cortex identifying that some areas of the cortex act as bridges between neuronal clusters. These kind of regions are called hubs, see Figure 1.12. Hubs in the cerebral cortex have acquired great importance in brain network studies. This is because hubs represent communication bridges among several other regions and if affected by a disease, the performance of the brain network will decrease considerably. Furthermore, it is hypothesized that hubs are also in charge of synchronizing neurons in regions they interconnect (Uhlhaas et al. , 2009).

Latter work has focused on the structural network in the human brain using diffusion spectrum imaging (DSI), a technique capable of imaging the axonal tracts in the brain white matter. Hagmann et al. (2008) used DSI to infer the structural network of the cerebral cortex, see Figure 1.13. In his publication he divided the cortex in 998 regions of interest (ROIs) and inferred connections using DSI axonal paths. It was found in Hagmann et al. (2008) that cortex is composed mostly of short range connections. 54% of them were within their ROI, 42% among ROIs

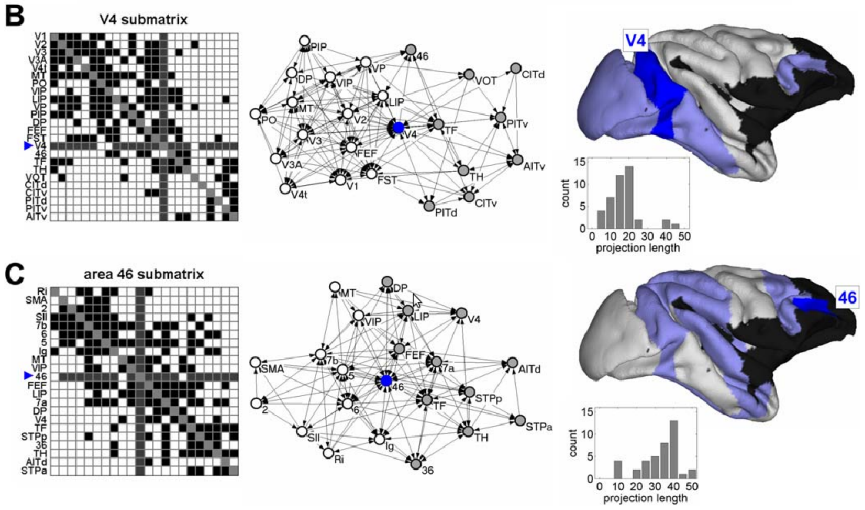

Figure 1.12: Macaque cortex. Regions V4 and 46 are hubs that communicate two important clusters of nodes. In the left column the connectivity matrices are shown. The middle column shows the network where hubs 46 and V4 are highlighted in blue and the two regions they communicate in white and gray respectively. The third column shows the cortex surfaces with regions V4 and 46 shaded in blue and their direct neighbours shaded in light blue. Image from Sporns et al. (2007). Colour image is shown in Figure B.1.

of different areas in the same hemisphere and 4% were interhemispheric connections. These results will help to relate the functional network with the structural counterpart.

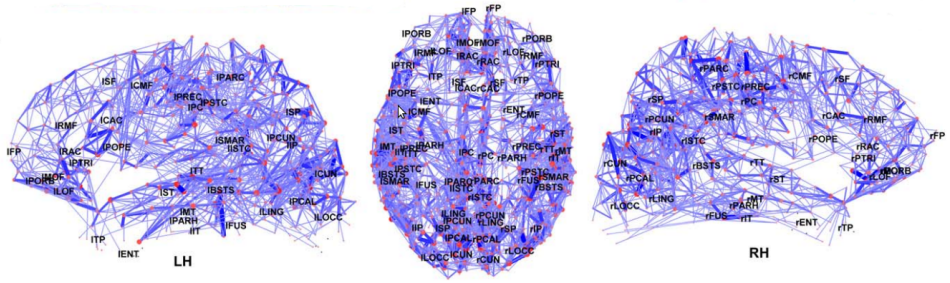

Figure 1.13: Dorsal and lateral views of the structural brain network backbone. The width of the edges is proportional of the connection weight. Image from Hagmann et al. (2008).

A complex system such as the brain network can also be studied from the point of view of efficiency. Efficiency measures the capacity of information flow within the network and this is directly related to network connectivity and wiring cost. It is mathematically defined as the inverse of the average path length, see Section 5.4. Achard and Bullmore (2007) studied efficiency of the small-world brain network and

how this is affected by ageing and pharmacological blockage of dopamine transmission. In Achard and Bullmore (2007) it was found that brain efficiency decrements with ageing and in the same manner when using sulpiride drug (a dopamine antagonist), see Figure 1.14.

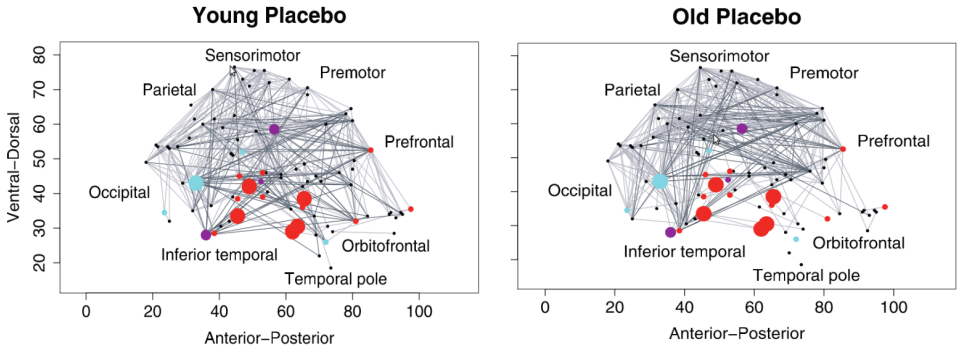

Figure 1.14: Brain functional networks of a young person and old person following placebo. The node efficiency is colour coded: The red nodes have a reduced efficiency by age, blue nodes by sulpiride (a dopamine antagonist), and purple nodes by both, age and sulpiride. Image from Achard and Bullmore (2007). Colour image can be seen in Figure B.2 (Appendix).

Fair et al. (2009) studied the brain network evolution following its development, by analysing the brain network of subjects at different ages. What the researchers found was that the brain network evolves from local connectivity to distributed connectivity. Distributed connectivity is fully related to the long range connections in the small-world network and these edges are at the same time fully correlated with the efficiency, synchronizablity and parallel information processing of the brain, indicating that the brain improves itself during our learning and maturation. This improvement helps us to solve problems faster. See Figure 1.15.

Kaiser and Hilgetag (2006) also studied wiring cost in the nervous system by using the available datasets of the C. elegans and the macaque cortex. Here a simulated annealing algorithm was implemented to rearrange the position of known nodes from their original position to a new position, having as objective function the minimization of the wiring cost. What they found is that positions of brain regions are not optimal. The wiring cost can be decreased dramatically by interchanging the physical position, but the average path length increases. This means that the number of edges between any pair of nodes increases. What this research suggests

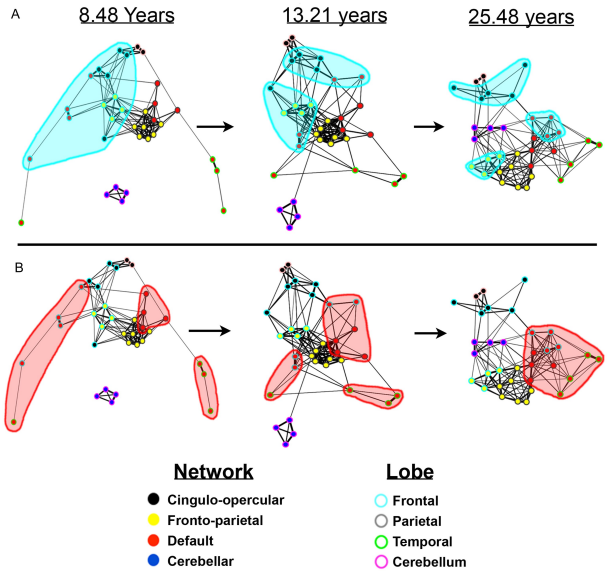

Figure 1.15: Default network maturation. The graph shows two individual screen shots A and B of times series correlation among brain areas. The data was divided in three groups according to their age and the nodes are colour coded. It can be seen that in children nodes are more clustered according to their location while in adults there are more connections among nodes of different regions. Image from Fair et al. (2009).

is that wiring cost is not the only objective of the small-world topology in the brain. The long paths in the brain network are placed with the intention of decreasing the path length and they are maintained in order to keep the optimal information flow by avoiding the use of several paths. See Figure 1.16.

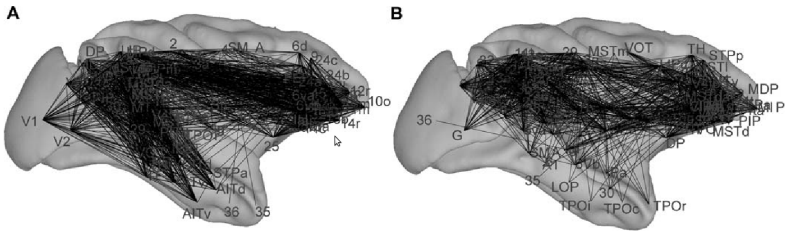

Figure 1.16: Macaque cortex network, original (left) and rearranged (right). The rearranged network was created by reallocating the physical position of the original nodes and keeping their original edges intact. By this arrangement it is proved that the wiring cost of the original network could be minimized, indicating that position of brain regions in the macaque cortex is not optimal and they obey other evolutionary rules than wiring cost only. Image from Kaiser and Hilgetag (2006).

The brain network can also be studied at different range of frequencies which the system might use for information flow purposes. For instance the long edges

in the functional brain network are stronger at low frequencies. Salvador (2005) showed this using fMRI studies on several patients in resting state. The importance of the work of Salvador (2005) is that long paths tend to communicate using low frequencies and this functional synchronization occurs at low bands below 0.3Hz (Salvador , 2005).

The resting state network has become an important tool for network analysis. Although it was taken with scepticism when it was proposed in Raichle and Snyder (2007), today is widely used and accepted in functional connectivity experiments. Raichle et al. (2001) were the first to propose a default mode network in the brain, a network which is always present but it is detectable when the subject is not performing a task. During decades the study of functional localization was focused on the activation of one or more areas when performing a specific cognitive task using experimental paradigms carefully designed to solve the psychological hypothesis of interest. Raichle noticed that when the brain performs a specific cognitive task the consumption of oxygen increases only 5% compared with the resting state. Furthermore, it was noticed that when a cognitive task was performed (and the oxygen consumption increased in a specific region) the oxygen consumption of other regions decreased. Raichle and others concluded that these decreasing areas were activity that was ongoing before the task performance. This activity was called default function (Raichle et al. , 2001; Raichle and Snyder , 2007).

The default mode network or resting state network is a common accepted concept, and practically all research work today in brain connectivity includes resting state experiments. The resting state network has been pointed to be very useful for brain studies, and mainly brain related diseases (Buckner et al. , 2008). For instance de Vico Fallani (2009) used resting state and EEG to study brain network metrics in stroke patients. Also, He et al. (2010) and Stam (2004) applied resting state and brain network analysis to study brain connectivity in Alzheimer's disease.

1.6 Neuroimaging

Current findings on brain networks would not have been possible without the aid of brain imaging technologies. Brain imaging has its beginnings when Hans Berger (1873-1941) recorded the first human electroencephalogram using a Galvanometer of 130 μV/cm and photographic paper. Since Berger's work, the recording of brain signals has followed a continuous evolution, with the introduction of computers that led to new algorithms for quantitative analysis and better visualization.

According to the functional localization theory each brain area has a specific cytoarchitecture because these areas specialize in information processing linked to some task, either cognitive or motor. There are sections in the brain involved in memory like the hippocampus, visual processing like the visual cortex, or movement like the motor cortex. When a subject performs a task, neurons that are in charge of carrying on the task will fire more intensely. It is supposed that there is a cause-effect relation between the functional or cognitive activity performed by the subject and the neuronal firing.

When a large number of neurons fire it is possible to record the generated electrical signal without the need of single neuron electrodes that pierce and later kill the neural tissue. This is possible because neighbouring neurons tend to synchronize and fire at the same time, adding up their potential to measurable voltages. There are several techniques to measure brain activity, the most common are electrocorticography, electroencephalography, magnetoencephalography, and functional magnetic resonance imaging. The latter does not measure electrical activity but metabolic activity of neurons.

1.6.1 Electroencephalography

The first studies on human electroencephalography (EEG) came from the German neuropsychiatrist Hans Berger using rudimentary Galvanometers and photographic paper. Berger published a series of fourteen reports of his research work mainly related to the alpha band, the first of these published in 1929 (Niedermeyer and Lopes , 1993). In his reports Berger noticed that different signal waveforms are

present in EEG and also gave sufficient evidence to suggest that these signals might be correlated with functional activity (Niedermeyer and Lopes , 1993).

The different brain waves of approximately 50 μV amplitude (also called brain rhythms) that can be observed in an EEG exam are the gamma rhythms which are in the band of 30-80 Hz, the beta rhythms (13-30Hz), alpha rhythms (8-13Hz), theta rhythms (4-8Hz), and delta rhythms (0.5-4Hz). The alpha rhythms are associated with quiet, conscious states. This rhythm appears when the subject closes his eyes and stays conscious. Theta rhythm is highly present during sleep stages, and delta rhythm is the hallmark of deep sleep, see Figure 1.18.

EEG is a common technique worldwide for medical diagnosis, thus the placement of the electrodes on the head is standardized to allow the reproducibility of the test among different laboratories. This standard is the 10-20 system where the "10" and "20" are because the electrodes are placed on the scalp letting a space among them of 10% or 20% of the subject's scalp distance from the Nasion to the Inion. The Nasion is the zone of the face just above the nose between the eyes and the Inion is a bone bump at the back of the head between the scalp and the neck. Furthermore, the 10-20 system labels all the electrodes with a letter and a number. The letter indicates the lobe where the electrode is located, F for frontal, O (occipital), P (parietal), and T (temporal). There is also a C label for the electrodes placed just above the central sulcus. The number next to each label indicates the electrode order. The electrodes nearer to the centre receive lower label number, and the electrodes just over the line that crosses the scalp from the Nasion to the Inion instead of a number receive a z label. Figure 1.17 shows the arrangement of the electrodes for the international 10-20 system.

EEG is widely used because of its low cost and noninvasiveness. EEG causes no harm to the subject and can be repeated any number of times. Its main application is in medical diagnosis of brain related diseases like epilepsy, sleep disorders, head trauma or dementia. There are also applications to detect promptly Parkinson's and Alzheimer's diseases based on mapping of functional activity and evoked potentials (EP) (Niedermeyer and Lopes , 1993). An EP is a wave of very low magnitude that appears just after a stimulus (visual, auditory or somatosensory). Several trials of

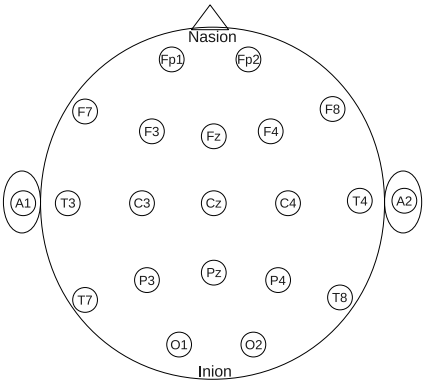

Figure 1.17: International 10-20 system for electrode placement in EEG. Electrodes A1 and A2 are used as voltage references.

the stimulus must be recorded in order to see an EP, otherwise the evoked wave would remain masked by the high amplitude of the background EEG. The main goal of this kind of research is to study the behaviour of the brain during different kind of stimuli. For example, it is possible to study the visual system by recording EEG signals from the visual cortex and see how the visual cortex activates using different kinds of images such as faces (Gazzaniga , 1998; Niedermeyer and Lopes , 1993).

EEG is the first tool used to diagnose epilepsy and can be used to approximately locate the source of the epileptic focus in the cerebral cortex. If the epileptic seizures are so intense and untreatable with conventional drugs, the epileptic tissue must be excised by opening the skull (craniotomy) in an attempt to eliminate the seizures. In this case, EEG gives the first clue of where the epileptic focus is in order to proceed with the craniotomy procedure.

EEG signals can also be used to implement brain computer interfaces (BCI). BCI research tries to help paraplegic patients by allowing them to interact with their environment using a computer interface. EEG electrodes are attached to the patient's scalp. Then a computer reads the brain signals to control an electronic device such as a mouse pointer on a computer screen. In this manner patients are able to move a mouse pointer and click on several options.

Because EEG signals are taken from the scalp and not from the cerebral cortex, there is a diffusion of the neural activity from the cortex to the scalp and also a

decrement of these potentials due to the skull resistance. This means that a signal acquired by an electrode placed on the scalp comes from a wider area of the cerebral cortex and also its potential is decremented due to the insulating property of the skull which behaves as a barrier for electric currents.

There are some issues with the acquisition of signals from the scalp, and one of these is the volume conduction problem. Volume conduction is manifested as a high correlation among all the EEG channels. This is because the cortex flesh, the CSF, and scalp skin are able to transmit electricity as a normal conductor. Thus a current that is present at the scalp on the left lobe side will also be present at the right lobe. This means that all channels in EEG will show a baseline correlation.

Another problem in EEG is the presence of *artifacts*, which are spurious signals that do not belong to the brain electrical activity but contribute to the EEG recording. This occurs because the brain is not the only organ that produces electricity in our body. The heart and all the muscles produce different kind of signals. The most common sources of artifacts in EEG are the heart (electrocardiogram or ECG), the eyes (electrooculogram or EOG), the muscles (electromyogram or EMG) and the tongue (glossokinetic artifact). Furthermore, artifacts might be several times larger than the brain signals, making them difficult to eliminate.

1.6.2 Electrocorticography

Even when EEG is successful on reading brain rhythms and localising epileptic sources, higher resolution is still needed for brain surgery. In this case electrocorticography (ECoG) is the next step after EEG.

ECoG is an invasive technique to measure electrical activity directly from the cerebral cortex. The reason why this technique is invasive is because a section of the skull must be removed in a surgery called craniotomy in order to expose the cerebral cortex. Then a grid of electrodes is placed over the cortical surface to record the electrical signals, as shown in Figure 1.19. The signals that are acquired using the grid of electrodes come mainly from the vertical pyramidal neurons that are placed in the gyri of the cortex, see Figure 2.1

Although very invasive, ECoG has good spatial resolution which depends on the

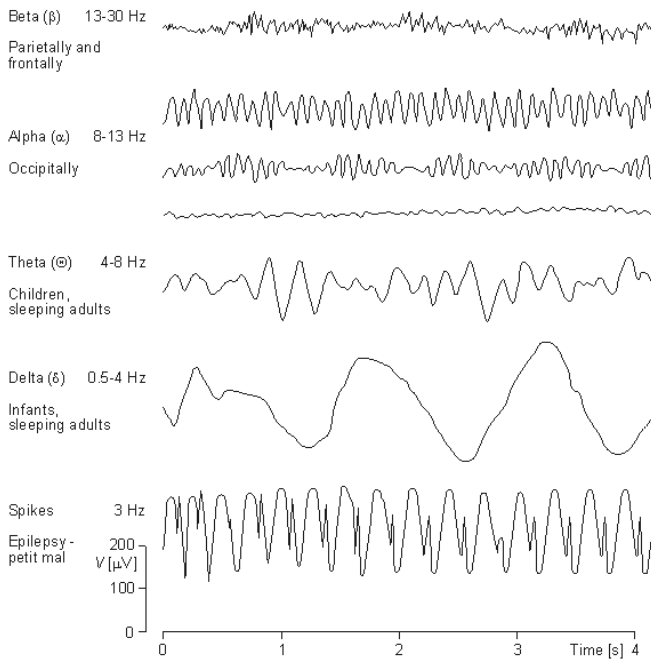

Figure 1.18: Typical EEG waveforms and their frequency bands. Image from Malmivuo and Plonsey (1995).

electrode grid used. Using new technologies like micro electrode arrays it is possible to obtain spatial resolutions of millimetres or even less. If the micro electrode arrays are combined with sampling frequencies above 10 KHz, it is possible to isolate the action potentials of individual neurons (Quiroga and Panzeri , 2009). Therefore, time resolution is also another of its advantages.

As in EEG another application of ECoG that is becoming important is BCIs. It is possible to implant a grid of electrodes on the cerebral cortex and seal the skull again letting enough space for the electrode's wires. Using algorithms of pattern recognition to process the brain signals, the subject can learn how to use the electrodes to move devices. An advantage of ECoG for BCIs is the automatic elimination of artifacts. Because the signals are recorded directly from the cerebral cortex, the muscle artifacts are eliminated automatically. The only artifact that might be present is the ECG artifact, but this can be avoided by not placing the cortical electrodes above cerebral vessels.

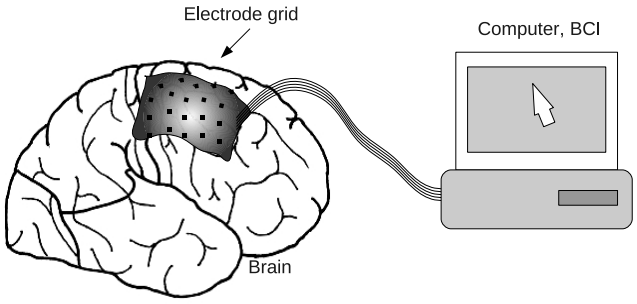

Figure 1.19: ECoG grid placed on the scalp. The ECoG can be also used for brain computer interfaces, which allow a paraplegic subject to interact with his/her environment through a computer device (for instance a computer mouse) and click on several options.

1.6.3 Functional magnetic resonance imaging

Magnetic resonance imaging (MRI) is a technique based on the principle that when atoms are excited by an electromagnetic pulse, they spin at specific frequencies which depends on the properties of the excited material. In 1920 the Austrian physicist Wolfgang Pauli noted anomalies in the electromagnetic spectra emitted by excited atoms. Pauli stated that atoms have two properties; a spin and a magnetic moment. He also stated that atomic frequencies were a discrete quantity, this means that different chemical elements spin at different frequencies. Pauli's conjectures would not be proved for more than a decade until Rabi's experiments (Huettel et al. , 2009).

In 1930 Isidor Rabi, an American physicist, ran an experiment using a modified version of Stern-Gerlach's gaseous beam (Huettel et al. , 2009). This gaseous beam uses a single element and allows to send a beam of different particles of the same element. Then the beam is passed through a magnetic field to induce a splitting of the beam into more beams. What Rabi did was to add an oscillating magnetic field and found that at a specific frequency and power the lithium atoms in the beam were deflected, indicating according to the experiment setup that these atoms absorbed energy. The phenomenon was called *magnetic resonance* (MR) and Pauli was awarded the Nobel Prize in Physics in 1944 for his discovery (Huettel et al. , 2009).

During the following years, Magnetic resonance's developing path was awarded four Nobel prizes in Physics and Medicine. The first two were won in 1952 by Felix Bloch and Edward Purcell for their work using MR in bulk matter. By 1945 the current MR experiments were only run using beams of particles, Bloch and Purcell proved that MR was also present in common materials. The second two Nobel Prizes were awarded to Paul Lauterbur and Peter Mansfield in 2003 for their work on improving the MR image acquisition. In 1972 Lauterbur had the idea of varying the magnetic field over the space, in this way the resonance frequency of different atoms would also vary. By measuring the resonance frequency and energy emitted by the excited atoms, Lauterbur was able to infer how many atoms of an element were present. This technique is called *induction of spatial gradients* (Huettel et al. , 2009). Lauterbur also noticed that spatial gradients only gave information in one dimension, thus he applied spatial gradients at different orientations in order to infer shape, Figure 1.20. In 1976 Mansfield imagined a more efficient recording approach. His technique used an electromagnetic pulse just before the introduction of magnetic gradients while recording the MR signals. Then the image could be reproduced by using Fourier analysis techniques (Huettel et al. , 2009). Mansfield's technique was later called echo-planar imaging (EPI) and still continues being the standard for MR Imaging (MRI) today.

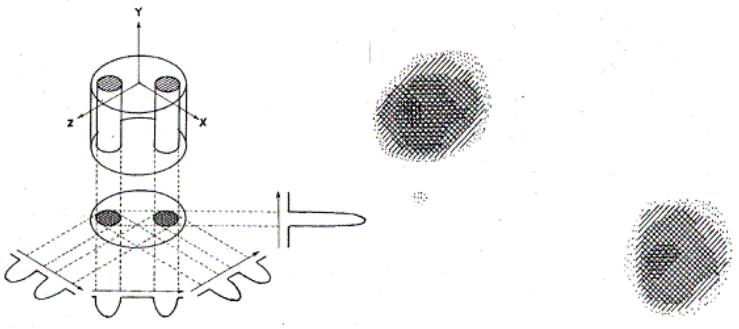

Figure 1.20: First MR Image obtained by Lauterbur. Lauterbur applied electromagnetic gradients at different orientations in order to infer to object's shape. In this case the shape of two bars.

The standard use of MRI as its name says is just for imaging. It can be used to obtain images of knees, feet, elbows, brain, and any other section of the body or all of it. MRI does not use any kind of ionizing radiation like Computerized Tomography

(CT), which uses X-rays. Thus it can be run several times on the same patient. Furthermore, MRI is a noninvasive technique, it does not require the injection of radioactive agents as occurs in positron emission tomography (PET) (Huettel et al. , 2009).

MRI obtains the adjective of "functional" when it is used to observe how the brain works when different cognitive tasks are performed and is commonly called fMRI for this case. In order to detect cognitive (or functional) activity in the brain, it is necessary to sense the activity of the cells in charge, the neurons. But fMRI cannot sense the electrical activity of neurons as EEG and MEG do. What fMRI does is to sense the metabolism of the neural tissue, the oxygen consumption which is highly correlated with neural activity.

BOLD fMRI

Blood-oxygenation-level dependent (BOLD) fMRI is a technique used to acquire images of brain functional activity. BOLD fMRI is based on the phenomenon that when a haemoglobin molecule is carrying oxygen the molecule is completely *diamagnetic*, this means that it has paired electrons and no magnetic moment. When haemoglobin releases its oxygen it becomes *paramagnetic*, meaning that now the molecule has a magnetic moment. Because paramagnetic materials distort magnetic fields, the protons that surround this material will be exposed to different field strengths. This phenomenon of paramagnetic haemoglobin is used in fMRI.

When neurons show high activity, they send several excitatory postsynaptic potentials (EPSPs) to other neurons. In fact this is the activity acquired by MEG. But when neurons send EPSPs (and also inhibitory or IPSP), this action represents a cost of energy for the neural cells because in order to maintain the ionic concentration inside and outside the cellular membrane, neurons have to activate their Sodium-Calcium pumps (membrane channels) to transport ions against the concentration gradient and restore the membrane potential (Bear et al. , 2007). When the Sodium-Calcium pumps are active, they consume *adenosine triphosphate* (ATP) molecules available in the cytoplasm. This molecule is commonly known as the energy currency for all cells in our body. The breaking of ATP molecules

produces enough energy to make the Sodium-Calcium channels work.

ATP molecules are created by mitochondria inside the neurons. It is thought that mitochondrial cells were bacteria that parasited the cells during evolution and they were found to be useful for energy production. Inside the mitochondria occurs a series of chemical reactions known as *Glycolysis* and the *Krebs cycle*. During Glycolysis the mitochondrion uses a 6-carbon molecule of Glucose and breaks it in two 3-carbon molecules of Pyruvate and two ATP molecules, by the investment of two ATP molecules. Therefore the total ATP gain at this point is zero. Then, in the presence of oxygen the pyruvate molecules oxidates and by a series of reactions using an enzyme called *ATP synthase* during the Krebs cycle, the mitochondrion generates 34 ATP molecules. Hence, in the presence of oxygen 34 ATP molecules per molecule of Glucose are generated. The oxygen needed by the mitochondria to create ATP is the reason why we need to breath and get oxygen from the air. The oxygen that enters the lungs is absorbed by the *red blood cells* where haemoglobin *oxidates*, and this oxygen is delivered to the brain and other tissues by the circulatory system.

Hence what fMRI really measures is the delivery of oxygen to neurons in the brain tissues or in other words, it measures *deoxygenation of haemoglobin*. The delivery of oxygen occurs in the capillary vessels. These vessels are thin enough to allow the crossing of just one red blood cell. During their crossing, the oxygen is delivered and the CO_2 generated as waste during the Krebs cycle is extracted (Huettel et al. , 2009). The increase of functional activity activates the flow of more blood through the capillaries because more oxygen is needed. Hence the increase of the blood flow also increases the number of deoxygenated haemoglobin molecules. Furthermore, it is assumed that the increase in the rate of haemoglobin deoxygenation occurs in the cerebral area needed to perform functional activity.

The use of fMRI and its correlation with functional activity has opened new research horizons. Now it is possible to find with high spatial resolution the exact point where the neural activity is happening. But even a high-tech device like fMRI has some minor issues. First of all, the signal that is measured is not neural activity but a collateral phenomenon (deoxygenation of haemoglobin) which is assumed to be correlated with functional activity. Also, BOLD signals are not “real time”.

This is because when EPSPs and IPSPs are sent, neurons will use first their stock of ATP and their current oxygen and just a couple of seconds later an haemoglobin response will be detected and this will last for several seconds. Therefore, although fMRI shows a high spatial resolution it has low time resolution, not only for the time that it takes to get the image but also by the phenomenon that fMRI measures.

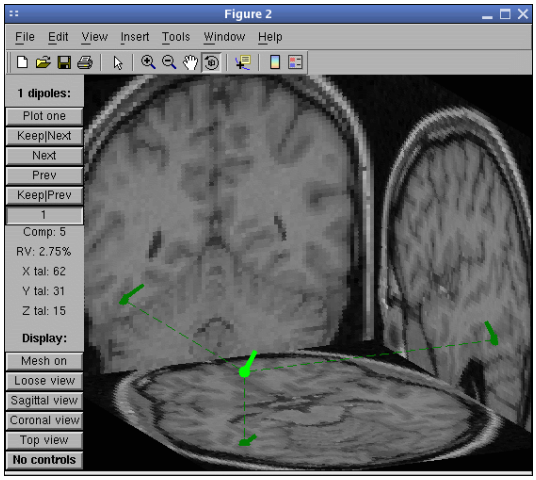

Figure 1.21: Coregistration of fMRI and MEG for dipole localization. Image obtained using EEGLab: http://sccn.ucsd.edu/eeglab/ .

An interesting use of fMRI is when it is accompanied by a MEG or EEG test. Even though fMRI has a low temporal resolution when detecting functional activity, MRI is the safest technique to obtain brain images. Thus, it is possible to obtain images from the same subject and perform *coregistration* with MEG in an attempt to get the best of both techniques, the spatial resolution of MRI and the temporal resolution of MEG, see Figure 1.21 for an example using EEGLab software. Having images of the subjects' brain allows to build more realistic models of the head and improves the accuracy of MEG in brain source localization. Furthermore, if MEG and fMRI are performed using similar experimental protocols, it is possible to use coregistration as an extra validation of the data if both techniques show similar results, like neural activity in the same area.

1.6.4 Magnetoencephalography

In magnetoencephalography (MEG) the brain signals are not acquired using electrodes. Instead of electrodes the signals are acquired using coils called *magnetometers*. The neuronal axons can be modelled as conductors carrying electrons and when an electrical current travels through a conductor it also produces a magnetic field. Therefore each neuronal axon is capable of generating a magnetic field.

The advantage of MEG over EEG is that the recorded signals come directly from the cerebral cortex instead from the scalp. This is because the skull does not offer any resistance to the magnetic fields as it does with the cortical electric currents. Furthermore, MEG does not need reference signals. Hence there are no ground electrodes that can be contaminated with artifacts or external noise.

MEG also has some weaknesses. The estimation of the neuronal sources is still not accurate enough to make it a fully reliable technique. There are many publications in the literature which work on source localization for MEG and the research on this field still continues. Currently, MEG techniques are able to theoretically find sources not only in the cerebral cortex but also in deeper regions of the brain. This is done by applying complex brain geometries that take into account the different brain tissues and also the skull.

Another problem in MEG is that the number of sources surpasses the number of sensors. The brain has thousands of neuronal groups firing at the same time and all of them contribute to the magnetic signals emitted by the brain. This number of superposed signals outnumbers the number of available magnetometers which makes the problem of finding the sources *ill-posed*. This means that the solution is not unique and many sources at different places can be found as probable solutions given the provided data. The problem of finding the position of magnetic sources is commonly called *the inverse problem* (Hämäläinen et al. , 1993; Baillet et al. , 2001).

Several techniques has been developed to tackle the inverse problem, like equivalent current dipole (ECD) (Sarvas , 1987) and beamforming (Van Veen and Buckley , 1988; Van Veen et al. , 1997). Beamforming is considered the most successful technique for source localization and was designed first in communications engineering

for target localization in radars and sonars. This topic will be studied more deeply in Chapter 2.

1.7 The volume conduction problem

Electrophysiological recordings present a common and difficult problem, volume conduction. Volume conduction is produced by the conductivity of the living tissues, which spreads the generated currents or voltages through the tissue volume. In the case of the brain the main tissues that produce volume conduction are the brain itself, the cerebrospinal fluid (CSF), the skull, and the scalp skin. Volume conduction causes that a cortical voltage generated by brain activity diffuses before arriving to the head scalp. Hence the signals recorded by EEG electrodes for instance are not necessarily generated at regions below them but by surrounding brain areas.

Volume conduction represents a challenge for connectivity studies and mainly for EEG. As we discussed in this chapter EEG acquires brain activity by placing electrodes on the head scalp, but due to volume conduction these signals are highly correlated to their neighbouring electrodes.

For electrical fields, volume conduction can be modelled analytically using models of concentric spheres, every sphere represents a conductive tissue. A widely applied model is the four sphere model, shown in Figure 1.22 where the four spheres are depicted representing each one from the inner to the outer; the brain, CSF, the skull bone, and the scalp. The spheres' radius are $r_1 = 8cm$, $r_2 = 8.1cm$, $r_3 = 8.6cm$, and $r = 9.2cm$ as shown in Figure 1.22. The conductivity ratios among spheres are commonly defined with the values $\sigma_1/\sigma_2 = 0.2$, $\sigma_1/\sigma_3 = 40.0$, and $\sigma_1/\sigma_2 = 1.0$ where the indices indicate the model's spheres.

The four sphere model and its analytical solution is well explained in Nunez and Srinivasan (2006). The equations for the voltages at any radius of the sphere were solved for a radial source (whose direction intersects the spheres' centre) and also a tangential source (tangential to the cortical surface). Neuronal sources at any direction can be modelled by using the linear properties of electromagnetism using three orthogonal dipoles. Sources at any position are also obtained by rotation of the

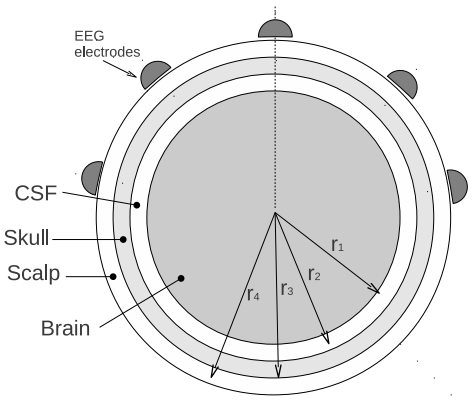

Figure 1.22: The four sphere model for volume conduction in EEG.

sphere model and again using the linear property of electromagnetism to simulate the activity of several sources distributed in the brain's sphere. Figure 1.23 shows a simulation of a tangential source (top row) and a radial source (bottom row) using the four sphere model. The sources were located at a radius of 7cm from the centre. Notice that the radial dipole shows a larger potential indicated by red colour (see Figure B.3). This shows that EEG is more sensitive to radial dipoles which are the main source of recorded EEG potentials. Radial dipoles are commonly present in the cortical gyri.

For the EEG case the research community has worked on several techniques to diminish the influence of the volume conduction problem. One of these approaches is to redefine digitally the EEG mounting, such as the averaged electrode reference, which consists on averaging all bipolar channels and use the result as the new reference. There are also source localization techniques for EEG which aim to reverse the volume conduction effect by unmixing the original sources using a volume conduction model. This method requires a deep knowledge of the head shape including its internal tissues. Although this information can be obtained by current imaging techniques like MRI, it is still a computationally intensive approach. Another method is independent component analysis (ICA) which is an improved version of the widely known principal component analysis (PCA). ICA aims to unmix the sources that generated the recorded signal assuming source independence and not

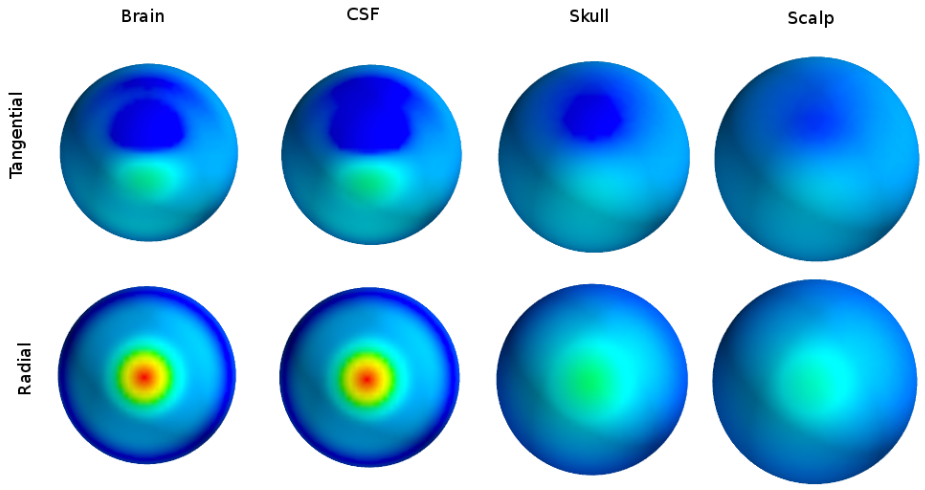

Figure 1.23: The four sphere model for volume conduction. The top row shows the four spheres representing the cortex, the CSF, skull, and the scalp when a tangential dipole is active. The bottom row shows the same but for a orthogonal dipole. Notice the surface voltage is more concentrated for orthogonal (also called normal) dipoles. Colour image can be seen in Figure B.3. Image created using Python-Mayavi2.

only orthogonality (or decorrelation) as PCA does, but it has two limitants. The first is the number of estimated sources, which is $N - 1$, where N is the number of electrodes, and the second is that the generating sources need to be independent.

MEG also suffers from the volume conduction problem, but the reasons of this are different to the ones in EEG. Volume conduction for MEG will be studied more deeply in Chapter 2.

Chapter summary

This introductory chapter gave a brief review of Neuroscience, its history, current developments and imaging technologies. It was emphasized that the brain is composed of billions of neural cells (100 billion approximately) and these neurons create a massively connected network with astonishing parallel (and also unparalleled for the best current computers) processing capabilities. Research work has found that the brain network follows a small-world topology. These kind of networks are associated with the optimum threshold between minimum wiring cost and maximum connectivity. It seems that the nervous system in humans and practically all species

evolved in order to minimize the number of steps among the network nodes by using long distance paths to connect high clustered subnetworks.

Some technologies used to acquire brain signals were also described in this chapter. These are EEG, ECoG, MEG, and fMRI. EEG is the oldest technique to acquire noninvasively brain signals. EEG acquires electrical activity generated mainly by pyramidal neurons located in the gyri of the cerebral cortex. However, because EEG electrodes are placed on the scalp and not directly on the cerebral cortex, the recorded signals suffer the volume conduction problem. A technique that overcomes this issue is ECoG. Here the electrodes are placed directly on the surface of the cerebral cortex, thus problems like volume conduction and skull resistance are avoided. ECoG is highly invasive because a section of the skull must be removed in order to get access to the cortical tissue and place the electrodes.

fMRI is another technology for functional and cognitive studies. MRI by itself can obtain images of the brain with high spatial resolution which are used by physicians to study brain tissues and detect tumours or any other malformation. Furthermore MRI can be adapted to sense functional activity (fMRI) by detecting the increase of deoxygenated haemoglobin in brain tissues.

MEG is a technique that technically overcomes the issues of EEG and also ECoG. MEG is able to acquire brain activity that comes directly from the brain tissue. What MEG detects are the magnetic fields generated by neuronal groups whose axons fire synchronously. It has been found that in the cerebral cortex, pyramidal neurons located in the sulci contribute most of the electromagnetic fields emitted by the brain. For this reason MEG is referred as a complemental technique for EEG, which acquires signals generated mainly in the gyri. However MEG also has one weakness regarded to the localization of the real current dipoles, but current research has proved that MEG is indeed a robust technique. This is explained in the following chapter.

Chapter 2

Magnetoencephalography

Magnetoencephalography or MEG is a brain imaging technology that senses the magnetic fields generated by large neuronal groups in the brain. Previously in Section 1.6.4 a brief introduction to MEG was given. In this chapter the mathematical concepts behind MEG as a brain imaging technique are covered.

2.1 Brain fields

When a neural group of approximately 10,000 cells fire synchronously in a small area of the brain, they create enough electrical current and in consequence enough magnetic field to be sensed by the MEG device. These magnetic fields are in the order of hundreds of femto-Teslas (10^{-15} Teslas) (Preissl , 2005). As in EEG and ECoG, these fields come from pyramidal neurons but instead of neurons located in the gyri, most of the recorded fields in MEG come from neurons located in the cortical sulci, see Figure 2.1.

Neural currents are commonly modelled as a current dipole $\mathbf{Q}$, which is nothing else than a small conductor segment that transports electrons. In Electromagnetism a current dipole can be created by placing two particles of opposing polarity and equal magnitude charges at a very close distance in a conductive environment. This is possible because the magnetic and electric fields of one charged particle can be computed easily since their analytical solution are well known. By placing two particles with opposite charges, the linear property of Electromagnetism allows to

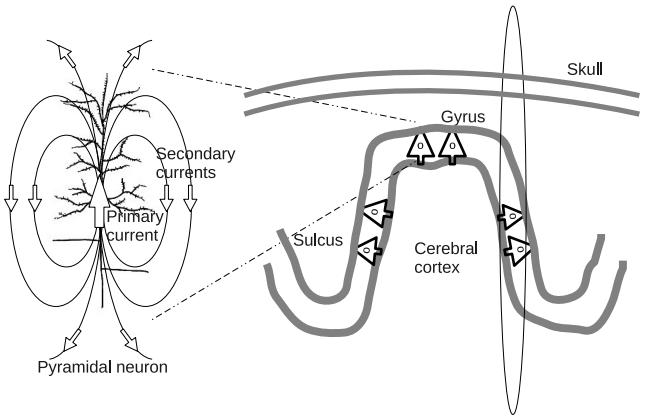

Figure 2.1: Pyramidal neurons in the gyri and sulci are the main sources for EEG and MEG signals respectively. It is needed the synchronous activity of approximately 10,000 pyramidal neurons to generate a magnetic field that can be sensed by the MEG device.

add up their fields thus creating the dipole.

In many MEG studies the frequencies of interest are below 100 Hz (Preissl , 2005; Hämäläinen et al. , 1993). At these frequencies the wavelengths of the brain fields are much more longer than the separation of the MEG sensors. Hence the quasistatic approximation of Maxwell's equations is used in brain dipole modelling (Hämäläinen et al. , 1993; Baillet et al. , 2001; Sarvas , 1987). These are

$$\begin{aligned} \mathbf{E} &= -\nabla V \ , \\ \nabla \times \mathbf{B} &= \mu_0 \mathbf{J} \ , \\ \nabla \cdot \mathbf{B} &= 0 \ , \\ \mathbf{J} &= \mathbf{J}^p + \sigma \mathbf{E} \ , \end{aligned} \tag{2.1}$$

where V is the electric potential, $\mathbf{J}$ is the total current density, $\mathbf{J}^p$ is the primary current, $\mathbf{B}$ is the magnetic field, and $\sigma\mathbf{E}$ is the Ohmic current. Maxwell's equations in Eq. 2.1 imply that the radiated signals do not suffer any delay due to spatial propagation.

The firing of several axons in a limited region is commonly modelled in the literature as a current dipole which generates a magnetic field $\mathbf{B}(\mathbf{r})$ defined as

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \int_v \mathbf{J}(\mathbf{r}_d) \times \frac{\mathbf{r} - \mathbf{r}_d}{|\mathbf{r} - \mathbf{r}_d|^3} dv \ . \tag{2.2}$$

Eq. 2.2 is the Biot-Savart law, which is used to compute magnetic fields generated

by $\mathbf{J}(\mathbf{r}_d)$ defined as

$$\mathbf{J} = \mathbf{J}^p - \sigma\nabla V \ , \tag{2.3}$$

where $\sigma\nabla V$ is the volume current that closes the circuit. Here σ is the macroscopic conductivity and the potential V is obtained by solving Poisson's equation $\nabla V = \nabla \cdot \mathbf{J}^p/\sigma$ (Hämäläinen et al. , 1993). $\mathbf{J}^p$ is the "battery" of the current dipole $\mathbf{Q}$ which is concentrated at the space point $\mathbf{r}_Q$ (Hämäläinen et al. , 1993). This concentration is defined using the δ function as

$$\mathbf{J}^p(\mathbf{r}) = \mathbf{Q}\delta(\mathbf{r} - \mathbf{r}_Q). \tag{2.4}$$

It is commonly assumed that only $\mathbf{J}^p$ contributes to the generation of the magnetic field to simplify the computation of the dipole position (Hämäläinen et al. , 1993), where the dipole current is confined in an infinitesimal conductive segment $\mathbf{dl}$. Using these assumptions we can rewrite Eq. 2.2 as

$$\mathbf{B}(\mathbf{r}) = \int \frac{\mu_0}{4\pi} \frac{J\mathbf{dl} \times (\mathbf{r} - \mathbf{r}_d)}{|\mathbf{r} - \mathbf{r}_d|^3} \ , \tag{2.5}$$

where J represents the current magnitude and $\mathbf{dl}$ the infinitesimal conductor segment. Computing the integral in Eq. 2.5 we have that

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \frac{q(\mathbf{r}_p - \mathbf{r}_n) \times (\mathbf{r} - \mathbf{r}_d)}{|\mathbf{r} - \mathbf{r}_d|^3} \ . \tag{2.6}$$

Here q represents the magnitude of the positive and negative charges, whose positions are r_p and r_n respectively. Figure 2.2 shows this dipole arrangement represented by Eq. 2.6.

By averaging the vector distances $\mathbf{r}_n$ and $\mathbf{r}_p$ to $\mathbf{r}_d$, Eq. 2.6 can also be represented as

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \frac{\mathbf{q}(\mathbf{r}_d, t) \times (\mathbf{r} - \mathbf{r}_d)}{|\mathbf{r} - \mathbf{r}_d|^3} \ . \tag{2.7}$$

where $\mathbf{q}(\mathbf{r}_d, t)$ is the vector whose magnitude and direction equals the modelled dipole and $\mathbf{r}_d$ is the dipole's centre. Notice that the variable t in the dipole representation was added to indicate that the dipole's current changes over time. Now assuming that in the brain there are N number of dipoles or sources. Using the linear property of Electromagnetism, it is possible to represent the magnetic field

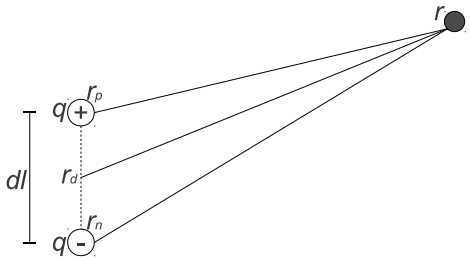

Figure 2.2: Dipole represented by charges of equal magnitude but different polarity separated by a distance *dl*. **r** represents the point of magnetic field measurement.

$\mathbf{B}(\mathbf{r})$ produced by the N dipoles as

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \sum_{i=1}^{N} \frac{\mathbf{q}(\mathbf{r}_i, t) \times (\mathbf{r} - \mathbf{r}_i)}{|\mathbf{r} - \mathbf{r}_i|^3} \tag{2.8}$$

Eq. 2.8 represents the magnetic field produced by N dipoles and it is the base for source inference in magnetic source imaging.

Current dipoles orthogonal to the scalp surface represent the main sources in EEG recordings. These orthogonal dipoles are mainly placed in the cortical gyri. An example of an orthogonal or radial dipole is shown in Figure 2.3 using the head's four sphere model in Figure 1.22.

Figure 2.3 shows the scalp layer and its electric potential distribution caused by an orthogonal dipole placed in the brain cortex (Nunez and Srinivasan , 2006). Notice here that the scalp potential is well localised, and the point of maximum voltage is just above the dipole position. In Figure 2.3, the dipole's position is shown with a red cone above the head's scalp, where the cone's vertex represent the positive pole. The dipole charge was of $1\mu C$ for this simulation.

Figure 2.4 shows the distribution of a current dipole tangential to the scalp cortex. Notice that for a tangential dipole, the potential is more dispersed on the scalp and also its magnitude is much lower than the one produced by an orthogonal dipole. Furthermore, the maximum is not above the dipole's position but some distance in front of the positive pole and in a similar way the minimum appears some distance from the negative pole.

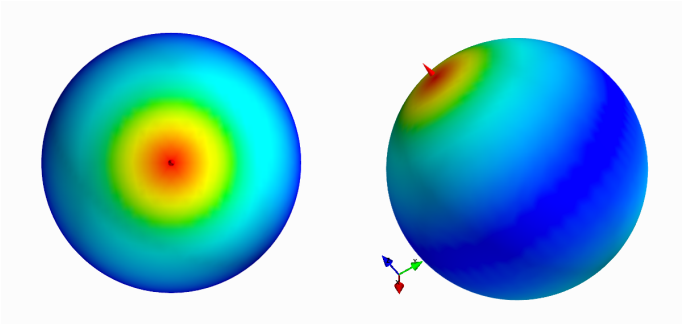

Figure 2.3: Orthogonal dipole on head's scalp. An orthogonal dipole is simulated using the four sphere model (Nunez and Srinivasan , 2006). The electric dipole was placed inside the brain. The figure shows the voltage field produced by the dipole on the head's scalp, which is shown on the outermost sphere (recall the four sphere model). Notice how the fields maxima is located just above the current dipole, whose position is represented by a red cone above the scalp. The image was programmed and created using Python-Mayavi2. Colour image can be seen in Figure B.4.

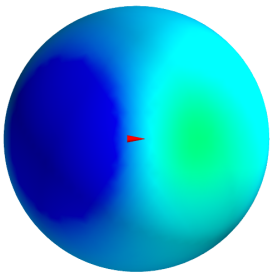

Figure 2.4: Tangential dipole on head's scalp. As in Figure 2.3, a current dipole was placed in the brain but in this case its direction is tangential to the scalp cortex. The position of the dipole inside the four sphere model is shown with a red cone with its vertex pointing to the positive pole. Notice that in this case, the field's distribution is not concentrated as in the orthogonal case. The field is much wider and its voltage lower. Also, the field shows one minimum and one maximum and the real position of the dipole is located in the middle of these two points. This pattern is characteristic of tangential dipoles. The image was programmed and created using Python-Mayavi2. Colour image can be seen in Figure B.5.

As we mentioned previously, MEG is more sensitive to dipoles that are tangential to the head's surface. These dipoles are commonly located in the sulci. Figure 2.5(left) shows the voltage distribution on the scalp produced by an orthogonal dipole. The orthogonal component of the dipole's magnetic field is shown on the sphere at the right in Figure 2.5. This sphere shows the orthogonal component magnitude of the magnetic field at four centimetres from the scalp. Four centimeters is the average distance between the scalp and the MEG sensors. Notice that the field acquired is uniformly zero, making impossible any localization attempt. The reason of this uniformity is because for spherical structures there are not orthogonal components in the magnetic field generated by an orthogonal dipole.

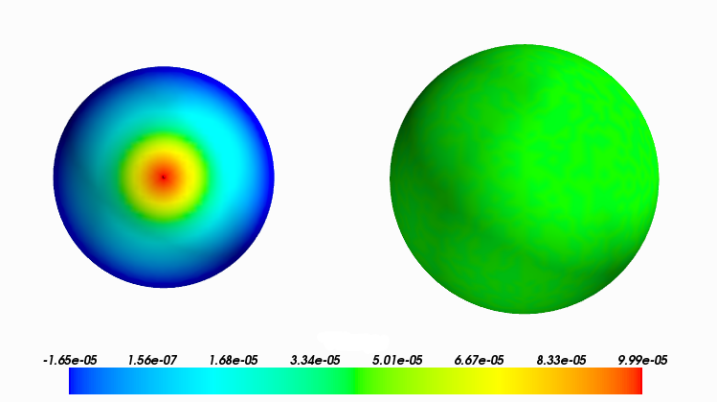

Figure 2.5: Magnetic field generated by an orthogonal dipole. The sphere at the right shows in colour the magnitude of the orthogonal components of the magnetic field that crosses the sphere. The sphere's surface is located four centimetres above the scalp sphere. Notice that the magnitude shown is zero (green colour), meaning that there are no orthogonal components. The sphere at the left shows the orthogonal dipole and its voltage field at scalp level. The image was programmed and created using Python-Mayavi2. Colour image can be seen in Figure B.6.

On the other hand, a tangential dipole presents clearly a minimum and a maximum in the field distribution which facilitate the localization of the generating dipole. As shown in Figure 2.6, a tangential dipole generates a magnetic field which has orthogonal components crossing the outer sphere, see Figure 2.6(left). Here the minimum and maximum are located where the orthogonal magnetic field magnitude is higher and the dipole is located exactly in the middle of both points. Basic source localization techniques in MEG use this minimum and maximum pattern of the field distribution to approximate the position of the generating sources.

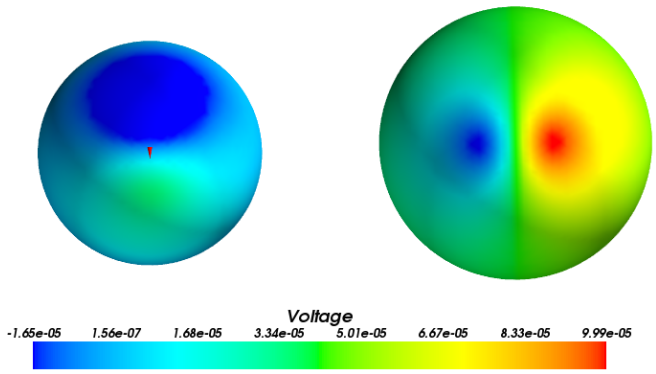

Figure 2.6: Magnetic field generated by a tangential dipole. The sphere at the right shows in colour the magnitude of the orthogonal components of the magnetic field that crosses the sphere. Notice that in this case a minimum and a maximum can be appreciated, and that the real position of the dipole is in the middle of these two points. The sphere at the left shows a tangential dipole and its voltage field at scalp level. The image was programmed and created using Python-Mayavi2. Colour image can be seen in Figure B.7.

2.2 Magnetoencephalographer

As we mentioned in Section 1.6.4, the magnetic fields generated by the brain are acquired using magnetometers. These are extremely sensitive to very small fields on the order of femtoteslas thanks to their high conductivity. The high conductivity is accomplished by lowering the magnetometers' temperature using liquid Helium which decreases their temperature to approximately 4 Kelvin, close to absolute zero. For this reason the magnetometers are located in a thermal drum to keep the low temperature inside. Noise can be reduced using special magnetometers called *gradiometers* which are an arrange of two magnetometers, see Figure 2.7. Gradiometers offer good performance because they only respond to magnetic fields whose source is close instead of sources that are far away. Therefore only magnetic fields coming from the brain are recorded (Preissl , 2005). The gradiometers used in MEG are very specialized devices called SQUIDs, which stands for superconductive quantum interference devices (Baillet et al. , 2001; Hämäläinen et al. , 1993).

There are few brands of MEG devices in the market. The newest ones surpass 300 SQUID sensors arranged in a semispherical helmet that surrounds the head. The MEG device at the York Neuroimaging Centre is a 4D Neuroimaging Magnes 3600 of 248 SQUIDs, shown in Figure 2.7-left. The layout of the 248 MEG SQUIDs

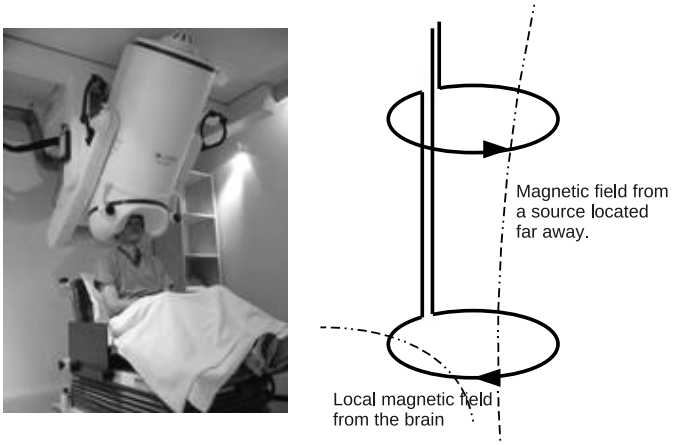

Figure 2.7: York Neuroimaging Centre (YNiC) (https://www.ynic.york.ac.uk/) magnetoencephalographer (left) and standard gradiometer (right). The gradiometer is designed to cancel fields coming from far away sources and sense only near sources.

is shown in Figure 2.8 where the SQUIDs are represented by white spheres and their axial direction is represented with blue arrows. In all simulations in this thesis the SQUIDs will be sensitive to magnetic field components parallel to them (parallel to the blue arrows in Figure 2.7).

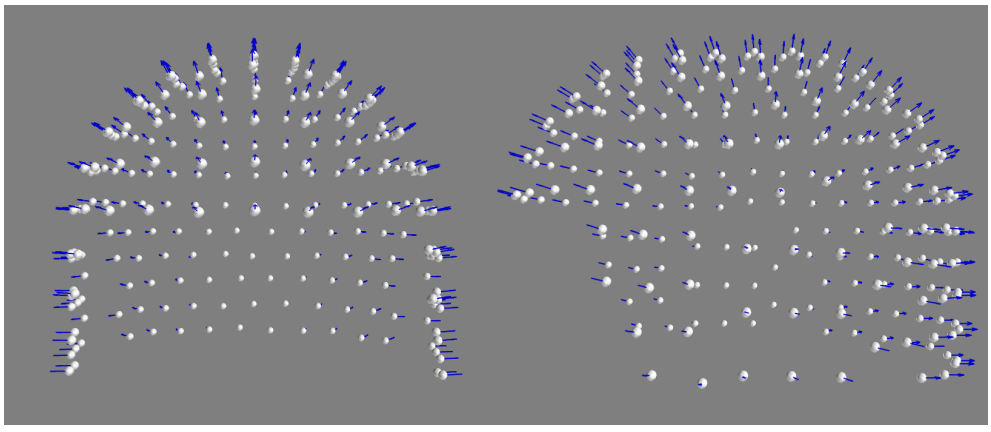

Figure 2.8: 248 SQUID layout from a 4D Neuroimaging MEG device. The blue arrows represent the SQUIDs' directions. Hence the SQUIDs are more sensitive to fields parallel to the blue arrows. The image was programmed and created using Python-Mayavi2. Colour image can be seen in Figure B.8.

2.3 Magnetic field acquisition

In MEG analysis it is commonly assumed that the SQUIDs are sensitive to magnetic fields parallel to the axial axis of the SQUID sensor. This means that the MEG sensors are much more sensitive to dipoles that are tangential to the head's scalp (Hämäläinen et al. , 1993; Malmivuo , 1997). This sensitivity can be modelled as

$$\mathbf{B}_n(\mathbf{r},t) = \frac{\mu_0}{4\pi}\mathbf{a}_n(\mathbf{r}) \cdot \sum_{i=1}^{N} \frac{\mathbf{q}(\mathbf{r}_i,t) \times (\mathbf{r}-\mathbf{r}_i)}{|\mathbf{r}-\mathbf{r}_i|^3} \ , \tag{2.9}$$

where we have added to Eq. 2.8 the dot product between the magnetic field and the SQUID directions $\mathbf{a}_n(\mathbf{r})$, represented as a unitary vector parallel to the sensors, see Figure 2.8. Eq. 2.9 can also be written in matrix notation as

$$\mathbf{x} = \sum_{i=1}^{N} \mathbf{H}(\mathbf{q}_i)\mathbf{m}(\mathbf{q}_i) + \mathbf{n} \ . \tag{2.10}$$

The time series vector $\mathbf{x}$ represents the acquired MEG signals which are proportional to the magnetic field $\mathbf{B}_n$, $\mathbf{H}(\mathbf{q}_i)$ is a matrix that represents the gain of the SQUIDs to a dipole $\mathbf{q}_i$ at the position $\mathbf{r}_i$, $\mathbf{m}(\mathbf{q}_i)$ represents the dipoles $\mathbf{q}(\mathbf{r}_i,t)$, and $\mathbf{n}$ is uncorrelated Gaussian noise that represents the instrumental noise.

In order to derive the matrices in Eq. 2.10 using Eq. 2.9, let's assume first that there is only one dipole in the brain space. Then, by recalling Eq. 2.9 we have that

$$b_n(\mathbf{r}) = \mathbf{a}_n(\mathbf{r}) \cdot \left[\frac{\mu_0}{4\pi} \frac{\mathbf{q}(r_i,t) \times (\mathbf{r}-\mathbf{r}_i)}{|\mathbf{r}-\mathbf{r}_i|^3} \right] \ . \tag{2.11}$$

Now, expanding the cross product inside the brackets of the previous equation gives

$$\begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = \frac{\mu_0}{4\pi|r-r_i|^3} \begin{bmatrix} 0 & r_z & -r_y \\ -r_z & 0 & r_x \\ r_y & -r_x & 0 \end{bmatrix} \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} . \tag{2.12}$$

By reintroducing the dot product with vector $\mathbf{a}_n(\mathbf{r})$, we obtain an expanded version of Eq 2.11 in matrix notation,

$$b_n(\mathbf{r}) = \frac{\mu_0}{4\pi|\mathbf{r}_{si}|^3} \begin{bmatrix} a_x & a_y & a_z \end{bmatrix} \begin{bmatrix} 0 & r_z & -r_y \\ -r_z & 0 & r_x \\ r_y & -r_x & 0 \end{bmatrix} \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} \tag{2.13}$$

where

$$\mathbf{r}_{si} = \mathbf{r}_s - \mathbf{r}_i = [r_x \; r_y \; r_z] \; , \tag{2.14}$$

and $\mathbf{r}_s$ and $\mathbf{r}_i$ are the position vectors of the SQUIDs and the dipole source respectively.

In order the generalize the previous result to N number of dipoles we define

$$\mathbf{a}_s(r) = [a_{sx} \;\; a_{sy} \;\; a_{sz}] \tag{2.15}$$

as the unitary vector parallel to each SQUID, here s defines the SQUID index. Also

$$\mathbf{R}_{si} = \frac{1}{|\mathbf{r}_{si}|^3} \begin{bmatrix} 0 & r_z & -r_y \\ -r_z & 0 & r_x \\ r_y & -r_x & 0 \end{bmatrix} \tag{2.16}$$

is the matrix of distances between the sth SQUID and the ith dipole. Using the previous definitions, matrix $\mathbf{H}$ can be represented as

$$\mathbf{H} = \frac{\mu_0}{4\pi} \begin{bmatrix} \mathbf{a}_1 & & & \mathbf{0} \\ & \mathbf{a}_2 & & \\ & & \ddots & \\ \mathbf{0} & & & \mathbf{a}_S \end{bmatrix} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \cdots & \mathbf{R}_{1N} \\ \mathbf{R}_{21} & \ddots & \dots & \mathbf{R}_{2N} \\ \vdots & & \ddots & \vdots \\ \mathbf{R}_{S1} & \mathbf{R}_{S2} & \cdots & \mathbf{R}_{SN} \end{bmatrix} . \tag{2.17}$$

Finally, by defining the set of active dipoles as the column vector

$$\mathbf{m}(\mathbf{q}) = [\; q_{1x} \;\; q_{1y} \;\; q_{1z} \;\; \dots \;\; q_{Nx} \;\; q_{Ny} \;\; q_{Nz} \;]^T \tag{2.18}$$

we obtain the matrix representation of Eq. 2.9

$$\mathbf{x} = \mathbf{H}(\mathbf{q})\mathbf{m}(\mathbf{q}) + \mathbf{n} \; . \tag{2.19}$$

Vector $\mathbf{x}$ has the times series acquired by the SQUIDs and the matrix $\mathbf{H}$ can be divided in several N submatrices

$$\mathbf{H}(\mathbf{q}) = [\mathbf{H}(\mathbf{q}_1) \;\; \mathbf{H}(\mathbf{q}_2) \;\; \cdots \mathbf{H}(\mathbf{q}_N)] \; , \tag{2.20}$$

where $\mathbf{H}$ is a $S \times 3N$ matrix, S is the total number of SQUIDs, and N is the hypothetical number of dipoles. Every submatrix in $\mathbf{H}$ can be used to compute the specific spatial filter for a hypothetical dipole source. Working with inferred sources is commonly called in the literature as source domain, while the contrary is called sensor domain.

2.4 Sensor domain

When working with MEG signals data can be analysed in the sensor domain or source domain. Sensor domain means that MEG signals used for analysis come directly from the SQUIDs. For instance Figure 2.9 shows the sensor acquisition of a tangential dipole. Recall that data from the SQUIDs are time series, hence the modelled brain dipoles fluctuate and change their polarity continuously. This is not fully accurate to real neuronal activity since neurons can not move and change their polarity. Nevertheless, for the simulations presented in this thesis the brain dipoles are allowed to change polarity in order to generate time series with zero mean.

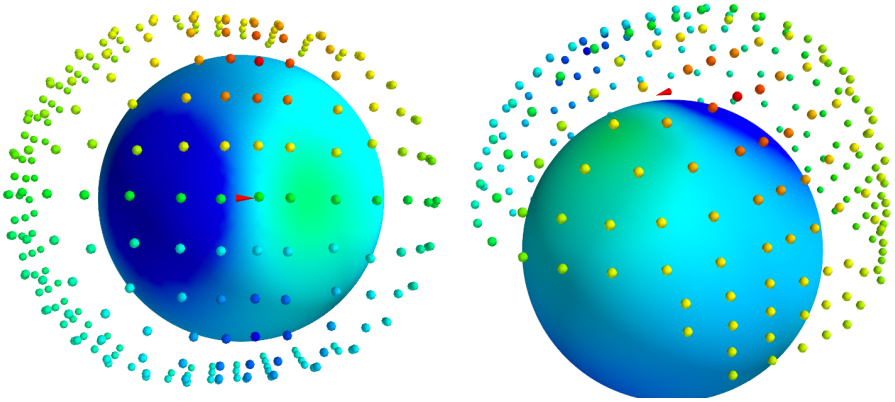

Figure 2.9: Magnetic field acquisition given by the MEG SQUIDs. Here the coloured smaller 248 spheres represent the magnitude of the acquired magnetic field using SQUIDs (one sphere per SQUID). This simulation was done by applying Eq. 2.9 to sense a tangential dipole. Also the scalp's potential distribution is shown using the four sphere model for EEG. Both images, left and right, show different views of the same magnetic field acquisition. The image was programmed and created using Python-Mayavi2. Colour image can be seen in Figure B.9.

Analysis in the source domain has several applications. As in EEG the brain rhythms can be read and used to find epileptic waveforms (Preissl , 2005). Evoked potentials (EP) can also be analysed using MEG in the sensor domain. The only disadvantage of this approach is that there is needed the average of approximately 100 EP experiments to see the desired response (Preissl , 2005). This is because the event responses have a very low voltage and they are completely overshadowed by the background brain activity. Using averaging the background activity will cancel since it is considered a random process, while the EP amplitude keeps approximately constant.

2.5 Source domain

Working in the source domain is the most appealing aim in MEG analysis. Source inference techniques will aim to find the physiological sources that emitted the acquired electromagnetic fields. This means that the position of the generating sources can be found either on the brain cortex or even in more internal structures of the brain, like the hypothalamus or the cingulate cortex.

Source inference in MEG has endless applications in brain studies, but it is not a straightforward task. The brain field activity might be generated by thousands of sources while the most modern MEG devices are just above 300 SQUID sensors. Hence there are more sources than sensors, causing an undetermined problem which means that there will be more than one solution for the position of the sources inside the brain. Current source inference techniques in MEG aim to find the best solution among the universe of them; the source positions within the brain volume that explain the best the MEG time series in the sensor domain.

2.5.1 Minimum Norm

Minimum norm (Wang et al. , 1993; Hämäläinen and Ilmoniemi , 1994; Wang et al. , 1992; Jeffs et al. , 1987) is an imaging technique which is based on the solution of Eq. 2.19. This can be attempted by writing the error function

$$\|\mathbf{e}\| = \|\mathbf{H}(\mathbf{q})\mathbf{m}(\mathbf{q}) - \mathbf{x}\ \|, \tag{2.21}$$

and finding a vector $\mathbf{m}$ that minimizes the error $\mathbf{e}$. Minimization of the previous equation is equivalent to minimize the squared error $\mathbf{e}^T\mathbf{e}$ which is easier to handle. Its derivative equated to zero leads to

$$\mathbf{H}^T\mathbf{H}\mathbf{m} = \mathbf{H}^T\mathbf{x}\ . \tag{2.22}$$

From which we can solve for $\mathbf{m}$ as

$$\hat{\mathbf{m}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{x}, \tag{2.23}$$

where $\hat{\mathbf{m}}$ stands for an estimated vector $\mathbf{m}$. Eq. 2.23 is known as the *least squares* solution for a vector of dipoles $\hat{\mathbf{m}}$ (Wang , 1993). Nevertheless as it was explained

previously, in MEG there are many solutions that can explain the current brain fields. In other words, very often the matrix $\mathbf{H}^T\mathbf{H}$ is nonsingular and its direct inverse does not exist. However, solution of Eq. 2.23 can be found by computing the Moore-Penrose inverse, often known as the pseudoinverse.

A simple way to compute the pseudoinverse is by SVD decomposition (Mosher et al. , 1992). SVD decomposition says that a $m \times n$ matrix can be decomposed in the matrix product

$$\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \ , \tag{2.24}$$

where matrices $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices or $\mathbf{U}\mathbf{U}^T = \mathbf{U}^T\mathbf{U} = \mathbf{I}$, $\mathbf{\Sigma}$ is a diagonal matrix $p \times p$, and $p = min\{m, n\}$. Hence, the inverse of $\mathbf{H}$ can be computed as

$$\mathbf{H}^\dagger = (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)^{-1} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T \ , \tag{2.25}$$

where † stands for pseudoinverse. Because matrix $\mathbf{\Sigma}$ has very low values, which produces very high values when computing its inverse, the lowest values are normally zeroed out. This process is commonly called "regularization" and it aims to filter noise from the equation system at the cost of lowering the spatial resolution for the estimated source places. Finally, the estimated dipoles can be computed by

$$\hat{\mathbf{m}} = \mathbf{H}^\dagger\mathbf{x} \ . \tag{2.26}$$

Figure 2.10 shows a minimum norm result of a single dipole. The red arrows represent the area where the dipole is more likely to be. This can be observed from the SQUIDs coloured values whose minimum and maximum are the sides of the estimated dipole position. For this example only the 15 highest values of the diagonal matrix $\mathbf{\Sigma}$ were preserved. The rest were zeroed out. An issue with minimum norm imaging technique is that it tends to find dipoles on the brain surface, even if the original source is located in a deeper region. Although there are weighting schemes to solve this problem, there is a technique that solves this issue in a more direct way called *beamforming* which is our next topic.

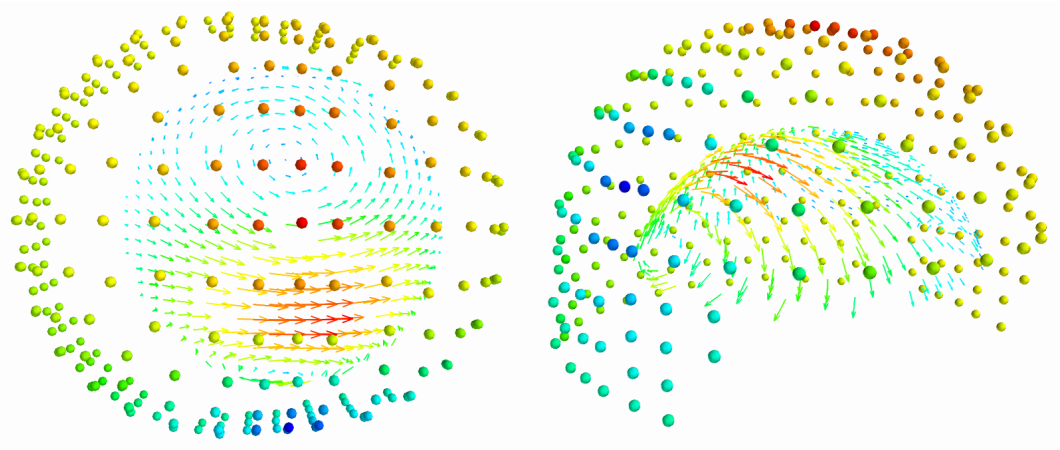

Figure 2.10: Minimum norm source estimation. For this simulation the surface of the scalp was tessellated to create regions of interest. Then, dipoles were estimated using regularised least squares or minimum norm. The red arrows indicate the most likely place where the dipole might be. This is the best dipole that explain the brain magnetic field whose magnitude is shown in coloured spheres. The image was programmed and created using Python-Mayavi2. Colour image can be seen in Figure B.10.

2.5.2 Beamforming

Beamforming as a brain imaging technique was introduced by Van Veen et al. (1997) as the linearly constrained minimum variance (LCMV) beamforming. Nevertheless, beamforming was already a very well known concept in the area of communication engineering (radars and sonars) (Buckley , 1987; Van Veen and Buckley , 1988). The aim of any beamforming technique is to find the position of a source of interest using an antenna array. In the case of radar the applications can be localisation of missile, airplanes or even people using cell phone communications. In sonar, beamforming also has plenty of applications such as localisation of submarines, whales, and even failure detection in car motors.

Brain imaging beamforming was envisioned in Van Veen et al. (1997) by bringing concepts from communication engineering to the brain imaging field. The concept is similar as in radar with the difference that the frequencies analysed in MEG are commonly below 100Hz. At these frequencies the electromagnetic fields can be considered static or in other words, there are no signal delays that can be sensed by the MEG SQUIDs. Hence, in MEG beamforming the time and frequency information can not be used for localisation as it occurs in radar.

Figure 2.11 shows how MEG beamforming works. Here the array of SQUIDs is used to create a spatial filter that will suppress sources coming from any region other than the region of interest. If nothing is transmitted from the region of interest, there will not be in theory any output from the beamforming filter. But if a real source is active in the analysed region, the beamforming output will be large. When the beamformer is oriented to sense only a particular region and ignore sources coming from other regions, the MEG spatial filter is commonly called a "virtual electrode", meaning that the MEG device is sensing virtually (without electrodes) that area.

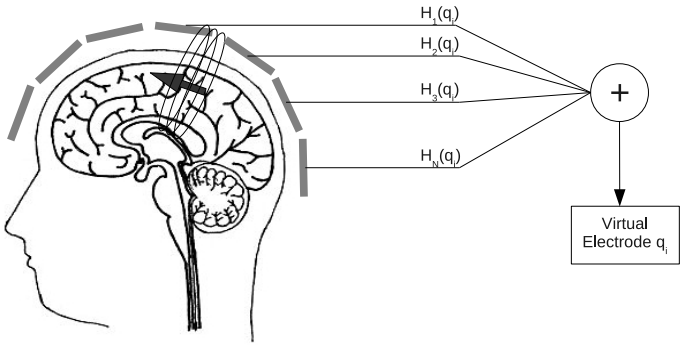

Figure 2.11: Beamforming creating a virtual electrode. The MEG SQUID array creates a virtual electrode by weighting the acquired vector using a spatial filter $\mathbf{W}$.

The LCMV beamforming derivation is fully presented in Van Veen et al. (1997). Here an extended version of their derivation is reproduced. Recall Eq. 2.19 and notice that it can be rewritten as

$$\mathbf{x} = \sum_{i=1}^{N} \mathbf{H}(\mathbf{q}_i)\mathbf{m}(\mathbf{q}_i) + \mathbf{n} \ , \tag{2.27}$$

where $\mathbf{H}$ is defined as in Eq. 2.20. Computing the expected value and covariance of vector $\mathbf{x}$ we obtain that

$$E\{\mathbf{x}\} = \sum_{i=1}^{N} \mathbf{H}(\mathbf{q}_i)E\{\mathbf{m}(\mathbf{q}_i)\} \ , \tag{2.28}$$

and

$$\begin{aligned} \mathbf{C}(\mathbf{x}) &= E\{[\mathbf{x} - E\{\mathbf{x}\}][\mathbf{x} - E\{\mathbf{x}\}]\} \\ &= \sum_{i=1}^{N} \mathbf{H}(\mathbf{q}_i)\mathbf{C}(\mathbf{q}_i)\mathbf{H}^T(\mathbf{q}_i) + \mathbf{D} \ , \end{aligned} \tag{2.29}$$

where $\mathbf{D}$ represents the covariance matrix of the noise $\mathbf{n}$ and it is supposed to be a diagonal matrix representing the instrumental noise variance of each SQUID in its elements. Assuming we have the solution for the system in Eq. 2.27, it can be written as

$$\mathbf{y} = \mathbf{W}^T(\mathbf{q}_o)\mathbf{x} \ , \tag{2.30}$$

where the solution vector $\mathbf{y}$ is the value of the estimated dipole and $\mathbf{W}^T$ is the matrix filter. This means that $\mathbf{W}$ and $\mathbf{H}$ are related by the property

$$\mathbf{W}^T(\mathbf{q}_o)\mathbf{H}(\mathbf{q}) = \begin{cases} \mathbf{I} & \mathbf{q} = \mathbf{q}_o \\ \mathbf{0} & \mathbf{q} \neq \mathbf{q}_o \\ & \mathbf{q} \in \Omega \end{cases} \ , \tag{2.31}$$

where Ω is the universe of dipole positions or regions of interest. Eq. 2.31 represents the spatial filter behaviour of matrix $\mathbf{W}$ which allows the acquisition of a possible source at location $\mathbf{q}_o$ while suppressing sources at other locations. However, this behaviour is not possible in real applications and its direct implementation might lead to solutions with very high source power (variance) at locations which are not source related. This problem can be solved with the use of Lagrange multipliers, which aims to keep the filtering property of Eq. 2.31 while minimising the variance of the estimated source. Hence, the name of minimum variance beamforming. This is represented with the following set of equations;

$$\mathbf{W}^T(\mathbf{q}_o)\mathbf{H}(\mathbf{q}_o) = \mathbf{I} \ , \tag{2.32}$$

$$\mathbf{W}^T(\mathbf{q}_o)\mathbf{H}(\mathbf{q}_s) = \mathbf{0} \ . \tag{2.33}$$

where the filter should be constrained to find the solution that gives the minimum variance of $\mathbf{y}$. This can be represented by

$$\min_W \quad tr[\mathbf{C}(\mathbf{y})] \quad . \tag{2.34}$$

where tr stands for the trace of the covariance matrix $\mathbf{C}(\mathbf{y})$. Notice that $\mathbf{y}$ is a 3-dimensional vector x, y, z, and its covariance is a 3×3 matrix. The previous equations can also be represented as

$$\min_W \quad tr[\mathbf{W}^T(\mathbf{q}_o)\mathbf{C}(\mathbf{x})\mathbf{W}(\mathbf{q}_o)] \tag{2.35}$$

$$subject\ to \quad \mathbf{W}^T(\mathbf{q}_o)\mathbf{H}(\mathbf{q}_o) = \mathbf{I}\ . \tag{2.36}$$

Eq. 2.35 and Eq. 2.36 represent the cost function to minimize. In order to achieve its optimization, Lagrange multipliers are used with the following Lagrangian definition;

$$L(\mathbf{W}, \mathbf{L}) = tr\{\mathbf{W}^T\mathbf{C}\mathbf{W} + (\mathbf{W}^T\mathbf{H} - \mathbf{I})2\mathbf{L}\}\ , \tag{2.37}$$

where $2\mathbf{L}$ is the matrix of Lagrange multipliers. Expanding the Lagrangian and recalling that $tr\{\mathbf{B}\} = tr\{\mathbf{B}^T\}$ we have that

$$L(\mathbf{W}, \mathbf{L}) = tr\{\mathbf{W}^T\mathbf{C}\mathbf{W} + (\mathbf{W}^T\mathbf{H} - \mathbf{I})\mathbf{L} + \mathbf{L}^T(\mathbf{H}^T\mathbf{W} - \mathbf{I})\}\ , \tag{2.38}$$

and completing the squares

$$\begin{aligned} L(\mathbf{W}, \mathbf{L}) =& tr\{(\mathbf{W}^T + \mathbf{L}^T\mathbf{H}^T\mathbf{C}^{-1})\mathbf{C}(\mathbf{W} + \mathbf{C}^{-1}\mathbf{H}\mathbf{L}) \\ & - \mathbf{L} - \mathbf{L}^T - \mathbf{L}^T\mathbf{H}^T\mathbf{C}^{-1}\mathbf{H}\mathbf{L}\} \end{aligned}\ . \tag{2.39}$$

It can be noticed that the nonfactorized terms do not depend on $\mathbf{W}$, and in consequence for $\mathbf{W}$ optimization purposes they can be ignored. Hence, in order to minimize the Lagrangian we have to choose a

$$\mathbf{W} = -\mathbf{C}^{-1}\mathbf{H}\mathbf{L} \tag{2.40}$$

which zero out the $\mathbf{W}$ dependent factors. Replacing $\mathbf{W}$ in the constraint rule in Eq. 2.36 gives

$$-\mathbf{L}^T\mathbf{H}^T\mathbf{C}^{-1}\mathbf{H} = \mathbf{I}\ . \tag{2.41}$$

And solving for $\mathbf{L}$ give us the Lagrangian matrix

$$\mathbf{L}^T = -(\mathbf{H}^T\mathbf{C}^{-1}\mathbf{H})^{-1}. \tag{2.42}$$

Finally, substituting the previous Lagrangian matrix in Eq. 2.40 gives the spatial filter solution

$$\mathbf{W}(\mathbf{q}_o) = [\mathbf{H}^T(\mathbf{q}_o)\mathbf{C}^{-1}(\mathbf{x})\mathbf{H}(\mathbf{q}_o)]^{-1}\mathbf{H}^T(\mathbf{q}_o)\mathbf{C}^{-1}(\mathbf{x}) \tag{2.43}$$

The filter $\mathbf{W}$ will be able to extract using a virtual electrode the times series of a source at position $\mathbf{q}_o$. The variance of the estimated source can estimated by

$$\widehat{var}(\mathbf{q}_o) = tr\{[\mathbf{H}^T(\mathbf{q}_o)\mathbf{C}^{-1}(\mathbf{x})\mathbf{H}(\mathbf{q}_o)]^{-1}\}\ . \tag{2.44}$$

This variance represents the power or level of activity of the brain region of interest. However, notice that the instrumental noise $\mathbf{n}$ and its variance $\mathbf{D}$ are always constant while the dipoles' power decreases in a square fashion due the Biot-Savart law (see Eq. 2.9). This means that at long distances the noise power will cover completely any possible source, making them invisible. This problem is solved by normalizing the dipole variance with the noise power as

$$\widehat{var}_N(\mathbf{q}_o) = \frac{tr\{[\mathbf{H}^T(\mathbf{q}_o)\mathbf{C}^{-1}(\mathbf{x})\mathbf{H}(\mathbf{q}_o)]^{-1}\}}{tr\{[\mathbf{H}^T(\mathbf{q}_o)\mathbf{D}^{-1}\mathbf{H}(\mathbf{q}_o)]^{-1}\}} \tag{2.45}$$

Notice that what Eq. 2.45 does is to take the power of the noise and place it just in the same position of the estimated dipole. Hence all dipoles will be normalized with an equivalent noise source. Eq. 2.45 is called in the literature as the *neural activity index* or NAI (Van Veen et al. , 1997).

Figure 2.12 shows an example of the LCMV beamforming performance. Here two dipoles were located in tangential orientation with respect to a spherical surface. Then a coronal slice within the sphere is defined as the area of analysis and finally tessellated to create regions of interest. The arrows represent the orientation of the estimated dipole $\mathbf{y}$ at every region and their colour is proportional to the NAI index value, being red the highest one and blue the lowest.

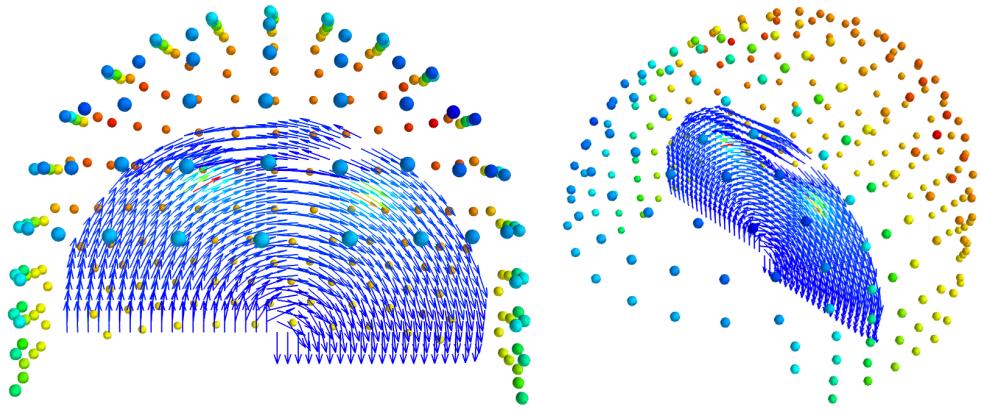

Figure 2.12: Beamforming simulation, localization of two dipoles. The LCMV beamforming finds the position of two tangential dipoles. Using a sphere to define the universe of possible places, a coronal slice was tessellated to define the regions of interest. Then, a dipole was estimated at each region. These dipoles are represented as coloured arrows being the red ones the best estimations. Both images, left and right, show the same simulation in different views. The image was programmed and created using Python-Mayavi2. Colour image can be seen in Figure B.11.

Figure 2.13 shows a heat plot of the same slice shown in Figure 2.12 with NAI

indices. The real position of the dipoles is exactly at the maximum values shown in red. For this simulation a signal to noise ratio $SNR = 25$ was used.

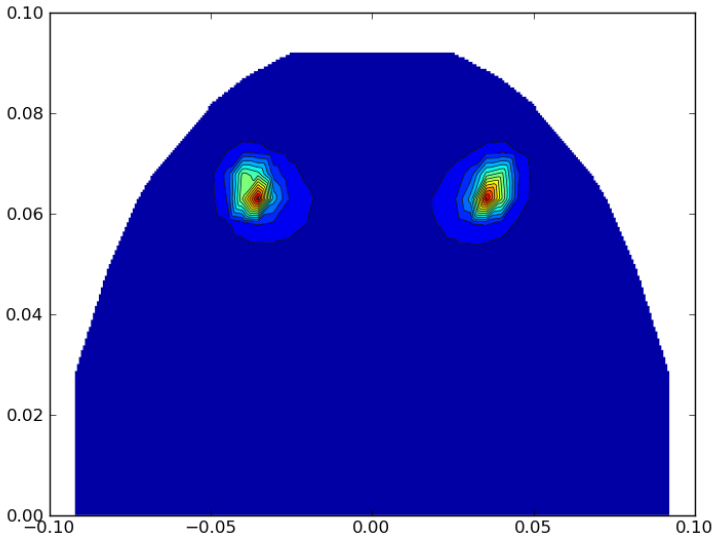

Figure 2.13: Same simulation as in Figure 2.12. Here the NAI index are shown using a heat map with spline interpolation. Notice that the LCMV beamforming can estimate accurately sources at deeper regions of the brain. The image was programmed and created using Python-Matplotlib. Colour image can be seen in Figure B.12.

2.6 LCMV Beamforming variants

LCMV beamforming (Van Veen et al. , 1997) has shown to be a very powerful technique for MEG studies. Nevertheless as every technique it has some issues that decrease its performance. For instance it is vulnerable to highly correlated signals (Vrba , 2002). However, it was previously demonstrated in Van Veen et al. (1997) that the technique is robust under certain limits of distance and correlation among the related sources.

For the reason mentioned before, several research groups have worked on improvements of the primary LCMV beamforming to tackle the source correlation issue, which is of paramount importance for brain connectivity analysis using MEG. Broadly speaking there are four kind of MEG beamfoming variants, catalogued by Huang et al. (2004) as Type I, II, III and IV.

2.6.1 Vectorized Type I Beamforming

The Type I beamforming is exactly the same LCMV beamforming whose NAI expression is rewritten here:

$$NAI(\mathbf{q}_o) = \frac{tr\{(\mathbf{H}^T(\mathbf{q}_o)\mathbf{C}^{-1}\mathbf{H}(\mathbf{q}_o))^{-1}\}}{tr\{(\mathbf{H}^T(\mathbf{q}_o)\mathbf{D}^{-1}\mathbf{H}(\mathbf{q}_o))^{-1}\}} \tag{2.46}$$

Huang et al. (2004) have rewritten Eq. 2.46 in a vectorized form as

$$NAI(\mathbf{q}_o) = \frac{(\mathbf{h}_x^T(\mathbf{q}_o)\mathbf{C}^{-1}\mathbf{h}_x(\mathbf{q}_o))^{-1} + (\mathbf{h}_y^T(\mathbf{q}_o)\mathbf{C}^{-1}\mathbf{h}_y(\mathbf{q}_o))^{-1} + (\mathbf{h}_z^T(\mathbf{q}_o)\mathbf{C}^{-1}\mathbf{h}_z(\mathbf{q}_o))^{-1}}{(\mathbf{h}_x^T(\mathbf{q}_o)\mathbf{D}^{-1}\mathbf{h}_x(\mathbf{q}_o))^{-1} + (\mathbf{h}_y^T(\mathbf{q}_o)\mathbf{D}^{-1}\mathbf{h}_y(\mathbf{q}_o))^{-1} + (\mathbf{h}_z^T(\mathbf{q}_o)\mathbf{D}^{-1}\mathbf{h}_z(\mathbf{q}_o))^{-1}} \tag{2.47}$$

where matrix $\mathbf{H}$ has been divided in its components; $\mathbf{H} = [\mathbf{h}_x \quad \mathbf{h}_y \quad \mathbf{h}_z]$ and every $\mathbf{h}$ is a column vector $N \times 1$. In order to normalize each dipole direction x, y, z separately with respect to the noise power. The "vectorized" LCMV beamforming is written then as

$$NAI(\mathbf{q}_o) = \frac{(\mathbf{h}_x^T(\mathbf{q}_o)\mathbf{C}^{-1}\mathbf{h}_x(\mathbf{q}_o))^{-1}}{(\mathbf{h}_x^T(\mathbf{q}_o)\mathbf{D}^{-1}\mathbf{h}_x(\mathbf{q}_o))^{-1}} + \frac{(\mathbf{h}_y^T(\mathbf{q}_o)\mathbf{C}^{-1}\mathbf{h}_y(\mathbf{q}_o))^{-1}}{(\mathbf{h}_y^T(\mathbf{q}_o)\mathbf{D}^{-1}\mathbf{h}_y(\mathbf{q}_o))^{-1}} + \frac{(\mathbf{h}_z^T(\mathbf{q}_o)\mathbf{C}^{-1}\mathbf{h}_z(\mathbf{q}_o))^{-1}}{(\mathbf{h}_z^T(\mathbf{q}_o)\mathbf{D}^{-1}\mathbf{h}_z(\mathbf{q}_o))^{-1}} \; . \tag{2.48}$$

The purpose for the beamforming vectorization is to avoid shadowing of the lower dipole components by the strongest one. For instance, in the common spherical head model the radial dipole does not produce any field outside the head (see Figure 2.5). Nevertheless, in a realistic head model this radial field component will not be zero but it will be anyway much weaker than the other components and in consequence its presence will be hidden by the other two components. Eq. 2.48 solves this shadowing issue.

2.6.2 Vectorized Type II Beamforming

Sekihara et al. (2001, 2002) published a beamforming based on the work of Borgiotti and Kaplan (1979), which has higher spatial resolution than the common

LCMV beamforming. This beamforming is designed with the following constraints (Sekihara et al. , 2001),

$$\begin{aligned} &min_{w_x}\mathbf{w}_x\mathbf{C}\mathbf{w}_x \\ &\qquad subject\ to\ \ \mathbf{w}_x^T\mathbf{w}_x = 1,\ \ \mathbf{w}_x^T\mathbf{h}_y = 0,\ \ and\ \ \mathbf{w}_x^T\mathbf{h}_z = 0, \\ &min_{w_y}\mathbf{w}_y\mathbf{C}\mathbf{w}_y \\ &\qquad subject\ to\ \ \mathbf{w}_y^T\mathbf{h}_x = 0,\ \ \mathbf{w}_y^T\mathbf{w}_y = 1,\ \ and\ \ \mathbf{w}_y^T\mathbf{h}_z = 0, \\ &min_{w_z}\mathbf{w}_z\mathbf{C}\mathbf{w}_z \\ &\qquad subject\ to\ \ \mathbf{w}_z^T\mathbf{h}_x = 0,\ \ \mathbf{w}_z^T\mathbf{h}_y = 0,\ \ and\ \ \mathbf{w}_z^T\mathbf{w}_z = 1, \end{aligned} \tag{2.49}$$

whose minimization procedure derives to the following vectorized beamforming activity index (Huang et al. , 2004)

$$NAI(\mathbf{q}_o) = \frac{\mathbf{h}_x^T(\mathbf{q}_o)\mathbf{C}^{-1}\mathbf{h}_x(\mathbf{q}_o)}{\mathbf{h}_x^T(\mathbf{q}_o)\mathbf{C}^{-2}\mathbf{h}_x(\mathbf{q}_o)} + \frac{\mathbf{h}_y^T(\mathbf{q}_o)\mathbf{C}^{-1}\mathbf{h}_y(\mathbf{q}_o)}{\mathbf{h}_y^T(\mathbf{q}_o)\mathbf{C}^{-2}\mathbf{h}_y(\mathbf{q}_o)} + \frac{\mathbf{h}_z^T(\mathbf{q}_o)\mathbf{C}^{-1}\mathbf{h}_z(\mathbf{q}_o)}{\mathbf{h}_z^T(\mathbf{q}_o)\mathbf{C}^{-2}\mathbf{h}_z(\mathbf{q}_o)}\ . \tag{2.50}$$

Sekihara et al. (2001) claimed that this beamforming is more robust than Type I against coherent sources and also that it has better SNR. Nevertheless, Huang et al. (2004) showed that it also tends to give false positive dipoles mainly at low SNR.

2.6.3 Vectorized Type III Beamforming

This beamforming is known as synthetic aperture magnetometry (SAM) (Vrba and Robinson , 2001; Vrba , 2002). The NAI index for SAM is

$$NAI(\mathbf{q}_o) = \frac{\mathbf{w}_x^T(\mathbf{q}_o)\mathbf{C}\mathbf{w}_x(\mathbf{q}_o)}{\mathbf{w}_x^T(\mathbf{q}_o)\mathbf{D}\mathbf{w}_x(\mathbf{q}_o)} + \frac{\mathbf{w}_y^T(\mathbf{q}_o)\mathbf{C}\mathbf{w}_y(\mathbf{q}_o)}{\mathbf{w}_y^T(\mathbf{q}_o)\mathbf{D}\mathbf{w}_y(\mathbf{q}_o)} + \frac{\mathbf{w}_z^T(\mathbf{q}_o)\mathbf{C}\mathbf{w}_z(\mathbf{q}_o)}{\mathbf{w}_z^T(\mathbf{q}_o)\mathbf{D}\mathbf{w}_z(\mathbf{q}_o)}\ . \tag{2.51}$$

A more understandable representation is the following;

$$\begin{aligned} NAI(\mathbf{q}_o) =& \frac{\mathbf{h}_x^T(\mathbf{q}_o)\mathbf{C}^{-1}\mathbf{h}_x(\mathbf{q}_o)}{\mathbf{h}_x^T(\mathbf{q}_o)\mathbf{C}^{-1}\mathbf{D}\mathbf{C}^{-1}\mathbf{h}_x(\mathbf{q}_o)} + \frac{\mathbf{h}_y^T(\mathbf{q}_o)\mathbf{C}^{-1}\mathbf{h}_y(\mathbf{q}_o)}{\mathbf{h}_y^T(\mathbf{q}_o)\mathbf{C}^{-1}\mathbf{D}\mathbf{C}^{-1}\mathbf{h}_y(\mathbf{q}_o)} \\ &+ \frac{\mathbf{h}_z^T(\mathbf{q}_o)\mathbf{C}^{-1}\mathbf{h}_z(\mathbf{q}_o)}{\mathbf{h}_z^T(\mathbf{q}_o)\mathbf{C}^{-1}\mathbf{D}\mathbf{C}^{-1}\mathbf{h}_z(\mathbf{q}_o)}\ . \end{aligned} \tag{2.52}$$

From which we can see the similarities to Type II beamforming. If the instrumental noise is considered independent univariate Gaussian noise, matrix $\mathbf{D} = \mathbf{I}$ will equate both beamforming expressions. For this reason this beamforming also tends to show false positive dipole estimations (Huang et al. , 2004).

2.6.4 Vectorized Type IV Beamforming

The fourth beamforming type was proposed in Huang et al. (2004). It is basically the LCMV beamforming but using a higher order covariance matrix. The covariance order recommended is $n = 3$.

$$NAI(\mathbf{q}_o) = \frac{(\mathbf{h}_x^T(\mathbf{q}_o)\mathbf{C}^{-n}\mathbf{h}_x(\mathbf{q}_o))^{-1}}{(\mathbf{h}_x^T(\mathbf{q}_o)\mathbf{D}^{-n}\mathbf{h}_x(\mathbf{q}_o))^{-1}} + \frac{(\mathbf{h}_y^T(\mathbf{q}_o)\mathbf{C}^{-n}\mathbf{h}_y(\mathbf{q}_o))^{-1}}{(\mathbf{h}_y^T(\mathbf{q}_o)\mathbf{D}^{-n}\mathbf{h}_y(\mathbf{q}_o))^{-1}} + \frac{(\mathbf{h}_z^T(\mathbf{q}_o)\mathbf{C}^{-n}\mathbf{h}_z(\mathbf{q}_o))^{-1}}{(\mathbf{h}_z^T(\mathbf{q}_o)\mathbf{D}^{-n}\mathbf{h}_z(\mathbf{q}_o))^{-1}} \ . \quad (2.53)$$

The intention of using a higher order covariance matrix is to increase the SNR beamforming performance which at the same times makes this technique more robust against coherent sources when compared with the standard LCMV beamforming. Huang et al. (2004) reported that this beamformer lacks of the false positive issues presented in Type II and III. In this matter it behaves similar to Type I LCMV. However, its advantage relies on its higher resolution when dealing with coherent sources.

2.7 MEG for brain connectivity

The power of MEG and beamforming for finding the generating brain sources, makes them a very attractive technology for brain connectivity research. At the time of writing this thesis, the amount of publications related to MEG and connectivity is not vast. This area is mainly dominated by studies that use fMRI as primary research tool. Nevertheless, the designing of newer beamforming algorithms added to the high time resolution of the MEG device (much higher compared with fMRI), increase the interest of the research community to work with MEG for brain connectivity studies. For instance Fuchs (2007) presented a review about the four beamforming types and their applications to connectivity analysis. In the same manner,

Schoffelen and Gross (2009) tested beamforming for functional connectivity and its performance when dealing with volume conduction in MEG.

Interesting work in MEG for brain connectivity was published by Stam et al. (2009); Stam (2010) in a study on brain networks in Alzheimer patients. Although Stam et al. (2009) worked in the sensor domain, this work shows the potential of MEG for these kind of studies. Brookes et al. (2004) published a MEG version of the general linear model (GLM), which is a common connectivity technique for fMRI. MEG has been also used to analyse networks when performing memory tasks as published by Brookes et al. (2011a) and also for Parkinsonian resting tremor networks (Timmermann et al. , 2003). These previous studies show the potential of MEG for brain connectivity.

There are also open source softwares dedicated to MEG analysis and beamforming like Fieldtrip (Oostenveld et al. , 2011), Nutmeg (Dalal , 2004), and statistical parametric mapping (SPM) (Penny et al. , 2007) which allow the research community the sharing and testing of techniques, accelerating research and advancements in MEG. A similar process was achieved by the fMRI community, where the sharing of open source software made fMRI the most successful tool for brain studies.

The methodology followed in this thesis for brain connectivity is explained in Chapter 7 where the real MEG data sets are analysed.

Chapter summary

In this chapter an introduction to MEG and beamforming were given. MEG is a reliable technique for brain studies. Its time resolution is higher than other imaging techniques like fMRI, although MEG offers lower spatial resolution. Beamforming techniques allow to estimate the generating magnetic sources in the brain, not only sources in the cortex but also in deeper regions.

More research is needed to improve MEG beamforming and different research groups continue working on this task. In this thesis, the Type I beamforming is used for all simulations and for the clinical datasets the YNiC's beamforming software which is also Type I is used, with the advantage that it offers MRI coregistration.

The next chapter offers an introduction to networks and network analysis, which will be necessary to understand further chapters.

Chapter 3

Introduction to network analysis

In Section 1.5 an introduction to brain networks was given. We talked about the small-world topology and also some of the current published work in the field (Achard and Bullmore , 2007; Basset and Bullmore , 2006; Brookes et al. , 2011a; Bullmore and Sporns , 2009). This chapter covers networks; their analysis, metrics, and their importance in brain research. The concept of *causality* is presented here, which is important to analyse directed networks.

3.1 An introduction to networks

The theory of networks has found applications in almost every field in science. In the literature we can find applications in urbanism, designing of subways and bus routes for big cities. In social networks, the main interest relies on finding the interactions among individuals. From this network we can study the interests of each subgroup (subnetwork) and see if they are connected by friendship, family, or a simple hobby. In Biology, the most famous networks are the protein networks, where the correlation of protein translation in the cell is studied. The favourite cells for these kind of studies are yeast and E. Coli.

Other examples are the Internet, which forms a network of computers, also telephone networks, electricity distribution grids, oil pipe networks, and airline route networks. In our daily life we can find networks everywhere and we can use network theory to study these systems, and infer properties from them. The obvious interest

in this thesis is the nervous system, in particular the brain and its cortex.

The neural network can also be studied using network theory. From the microscale level for neuron to neuron interactions to macroscale for region to region interaction. The former one requires invasive techniques such as micro electrode arrays, where tiny needle electrodes pierce the neural tissue and are able to acquire individual neuron firing. This procedure normally destroys the brain tissue and for this reason has limited applications in humans. However, macroscale level studies can be done using noninvasive techniques. As we saw in Section 1.6, EEG, MEG, and fMRI can sense functional brain activity coming from either the brain cortex or deeper regions, being this latter the case of fMRI and MEG.

When working with brain networks from MEG signals, there are two options: sensor domain and source domain (see Section 2.4 and 2.5). In sensor domain, time series are analysed practically raw from the MEG device and in the source domain a source localization technique must be applied in order to estimate the source's position and record its activity using a virtual electrode. In both cases, the result is a database of time series from which inference about their relations is performed to build networks and study their properties.

A network is composed of *nodes* which represent the sources and *edges* which are lines representing the relation between two sources. The edges can be undirected (a line) or directed (an arrow). When a network is composed of directed edges it is commonly called *directed network* or *causal network*, and in the brain imaging field it is also called *effective network*. Figure 3.1-right shows two networks of similar structure, the network at the left is an undirected network and the one at the right a directed network. In general, arrows in a directed network highlight causality, meaning that a node's behaviour is a consequence of another node, or in other words "a cause points to an effect or result". For example, in Figure 3.1 we can see that node 1 causes node 2, or another way to say this is that node 1 is a parent of child node 2. The description of network motifs using parents, children, and other family terms is common in network theory.

The most practical way to represent mathematically a network is using a connectivity matrix $\mathbf{C}$. where the column and row indices represent the nodes and the

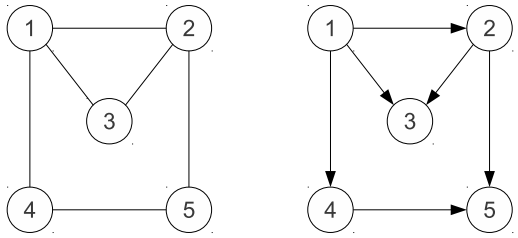

Figure 3.1: Undirected (left) and directed (right) network examples. Networks can be undirected or directed. In the undirected case the relation among two nodes is indicated by a simple line, nevertheless with this representation the flow of information is not indicated. If the cause and the effect is known within the network, causality can be represented by arrows, where the direction of the arrows indicates information flow.

matrix elements the connectivity structure among them. 1 stands for an edge and 0 no edge. This can be seen in Eq. 3.1 for the undirected matrix. Here $\mathbf{C}_u$ explains mathematically the connectivity among the five nodes of the undirected network in Figure 3.1,

$$\mathbf{C}_u = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} . \tag{3.1}$$

Notice that for the undirected case the connectivity matrix is symmetric. For the case of the directed network in Figure 3.1-right, $\mathbf{C}_d$ in Eq. 3.2 describes its directed connectivity. Here 1 not only represents the edge but also its direction. For example in the first row which represents node one, we can see that there are 1s at column 2, 3, and 4, indicating that node 1 is parent of nodes 2, 3 and 4, which are called children nodes. Also notice that following the parent-child logic, node 2 is also parent of node 5, and that 5 is the only node with no children or descendants. Hence, in this case the rows of the matrix represents parents and the columns children (although this convention can be changed) and the matrix $\mathbf{C}$ is

asymmetric. For the example in Figure 3.1-right the directed matrix is defined as

$$\mathbf{C}_d = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} . \tag{3.2}$$

3.1.1 Network metrics

The network connectivity matrix $\mathbf{C}$ allows to study the network from a mathematical perspective. The matrices we saw in the previous section are called directed and undirected matrices. However, they can also be called binary directed and undirected matrices to emphasize that these matrices have only two values, 1s and 0s to define edge and no-edge. There are other kind of matrices called *weighted matrices*, which are the result of estimating the relation among the sources at hand.

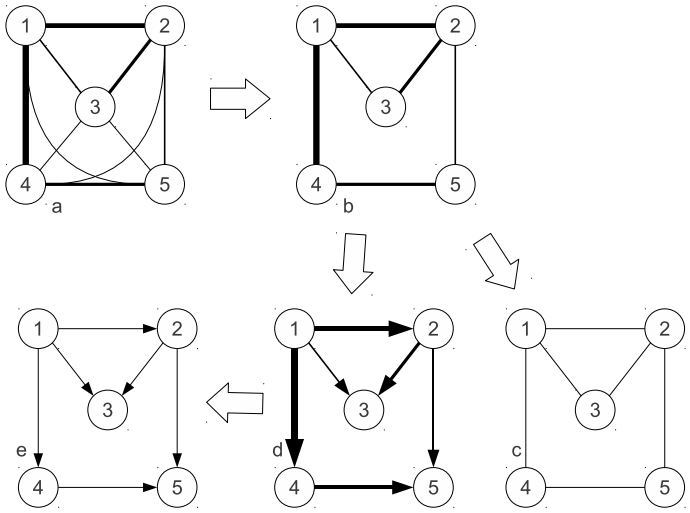

Figure 3.2: Directed and undirected weighted networks. Here we present the standard steps in network structure inference. a) First all edges are estimated creating a full connected network. The width of the edges are proportional to the strength of the relation between two sources. b) Then the edges are *thresholded* in order to keep the most important connections in the matrix/network. c) and d) From the thresholded network we can infer causality (d) or just connectivity by defining the undirected binary matrix (c). e) Finally, using the estimated causality we can define the network's directed binary matrix.

When edge estimation techniques are used, these commonly offer bounded scalar

values that define the edge strength, see Figure 3.2a. These values are normally called *network weights*. Weighted connectivity matrices can be used to describe a weighted network which is fully connected even though the strength of its connections is not equal. In order to keep the strongest or statistically significant edges only, a threshold is applied. The edges above the threshold will be set to 1 and below to it will be zeroed out. After the thresholding procedure what remains most of the time is a scatter network whose connectivity matrix has fewer 1s than 0s, and makes the network analysis more tractable. Furthermore, using causality estimation the edge direction can be defined. The network estimation and thresholding steps are depicted in Figure 3.2.

Directed or undirected, the maximum number of edges a matrix can have is

$$max\ edges\ in\ a\ network = \frac{N(N-1)}{2}\ . \tag{3.3}$$

Eq. 3.3 defines the *full connectivity* bound for a network. For instance a 5-node network can have a maximum of $5 * 4/2 = 10$ edges. *Reachability* is another concept that tells if all network nodes are connected, meaning that a node can be reached by any other node in the network. Reachability can be found by computing the power of the connectivity matrix.

$$\sum_{n=1}^{N-1} \mathbf{C}_{ab}^{n} = 0 \quad Disconneted\ nodes \tag{3.4}$$

$$\neq 0 \quad Connected\ otherwise \tag{3.5}$$

where the a and b indices indicate the row and column matrix elements respectively. If the sum of the elements (a, b) equals zero, it indicates that the node a can not be reached by node b. Hence the column and the row of the disconnected node will be full of zeros. Let's see this in the following example:

The connectivity matrices at Eq. 3.6 and 3.7 describe the network in Figure 3.3a and 3.3b respectively. Notice that in network a, node 5 is disconnected from the network, and that in network b there are two subnetworks where nodes 4 and 5 create one.

$$\mathbf{C}_1 = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.6}$$

$$\mathbf{C}_2 = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.7}$$

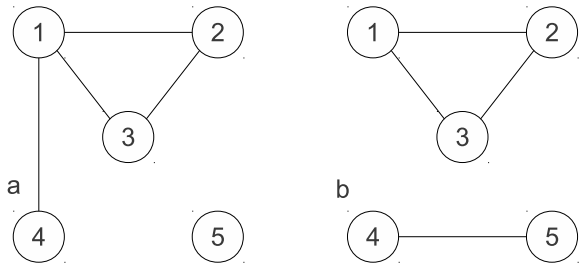

Figure 3.3: Connectivity matrix examples. a) This network has one disconnected node, whose row and column appear as zeros in the powered connectivity matrix, see Eq. 3.8. b) This network is composed of two disconnected networks, this will cause that the powered connectivity matrix be composed of the two independent submatrices one for each subnetwork, see Eq. 3.9

.

Computing the sum of the powered matrices we get the following results

$$\sum_{n=1}^{N-1} \mathbf{C}_1^n = \begin{bmatrix} 16 & 12 & 12 & 6 & 0 \\ 12 & 11 & 11 & 6 & 0 \\ 12 & 11 & 11 & 6 & 0 \\ 6 & 6 & 6 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.8}$$

$$\sum_{n=1}^{N-1} \mathbf{C}_2^n = \begin{bmatrix} 10 & 10 & 10 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 2 & 2 \end{bmatrix} \tag{3.9}$$

The zero elements in the previous matrices represent unconnected nodes, indicating that those nodes can not be reached. Notice that the network in Figure 3.3-b, the powered matrix is formed by two submatrices, indicating the number of unconnected subnetworks present in the system.

Another important measure in network analysis is *node degree*. Node degree measures how important is a node in the network indicating the number of edges that connect to it. The node degree is defined by

$$k_i = \sum_{j \in N} a_{ij} \ , \tag{3.10}$$

where a_{ij} represents an edge between the current node i and all its j neighbours and $a_{ij} = 1$ if there is an edge and $a_{ij} = 0$ otherwise. For directed networks, it is needed to define the *out degree* and the *in degree* to divide the edges that point outwards and inwards the node respectively. *In degree* and *out degree* are defined by

$$k_i^{out} = \sum_{j \in N} a_{ij} \ , \tag{3.11}$$

$$k_i^{in} = \sum_{j \in N} a_{ji} \ . \tag{3.12}$$

And for weighted networks the node degree is

$$k_{wi} = \sum_{j \in N} w_{ij} \tag{3.13}$$

where w_{ij} is the edge weight.

In network theory it is also interesting to know how distant is one node from another by counting the number of edges that separate them. A metric that account for this is the *shortest path distance* which is defined between two nodes i and j by

$$d_{ij} = \sum_{a_{uv} \in i \leftrightarrow j} a_{uv} \tag{3.14}$$

where $i \leftrightarrow j$ stands for shortest path between node i and j. Hence d_{ij} describes the number of steps (edges) between two nodes.

Furthermore it is possible to integrate the path distances in a network by computing the *average network path length* which gives us an indicator of how far in average the nodes are from each other. This measure is defined by

$$L = \frac{1}{n}\sum_{j\in N} L_j = \frac{1}{n}\sum_{j\in N} \frac{\sum_{k\in N, k\neq j} d_{jk}}{n-1}, \tag{3.15}$$

Notice that L_j is the average distance between the node j and the rest of the network nodes. A problem in the path length measure L is that when one node is completely disconnected from the network, d_{jk} becomes undetermined (some literature might also say infinite) and Eq. 3.15 is not valid in these cases. For disconnected networks there is another path length estimator called *harmonic mean distance*

$$L' = \left(\frac{1}{n}\sum_{j\in N} \frac{\sum_{k(\neq j)} \frac{1}{d_{jk}}}{n-1}\right)^{-1}. \tag{3.16}$$

Harmonic distance is designed to deal with the undetermined/disconnected paths by assuming these paths as infinitely long and their inverse equal to zero.

Another important measure is *clustering*, which measure the level of segregation or grouping in a network. The clustering coefficient C is defined as

$$C = \frac{1}{n}\sum_{j\in N} C_j = \frac{1}{n}\sum_{j\in N} \frac{2t_j}{d_j(d_j-1)} \tag{3.17}$$

where $C_j = 0$ for $d_j < 2$, and

$$t_j = \frac{1}{2}\sum_{k,h\in N} a_{jk}a_{jh}a_{kh} \tag{3.18}$$

is the number of triangles around the node j. There are in the literature more network metrics which highlight other network characteristics that will not be mentioned in this thesis. For a deeper reading on this area Newman (2010) is recommended.

3.1.2 Small-world and scale-free networks

Among the multiple characteristics that can be measured from network systems, their structure results to be possibly the most important one. The network struc-

ture or pattern defines the behaviour of the system that creates the network. Section 1.5.1 gave an introduction to small-world networks which are characterized by high clustering and low short path length. It is possible to concentrate these two measures in just one called *small-worldness* S, defined by

$$S = \frac{C/C_{rand}}{L/L_{rand}} , \tag{3.19}$$

where C_{rand} and L_{rand} are the clustering coefficients and the average distance path respectively of a random network with the same number of nodes. If $S \gg 1$ the network is considered a small-world network. Small-world networks can be found almost everywhere, and very frequently in nature. It has been pointed out that small-world networks offer the optimal trade off between connectivity and wiring cost. Wiring cost is an important measure in nature because for biological systems to maintain and nurture several connections is very expensive, so it is very likely that a biological system such as the brain tries to optimize connectivity and save resources when possible. Some literature suggest this logic is followed by neurons to create their networks. See for instance Achard and Bullmore (2007), and Kaiser and Hilgetag (2006).

Another famous network structure for biological interest is the *scale-free network*. For instance, it has been shown that protein networks in the cell show a scale-free architecture and in other fields, the internet and the world-wide-web also show a scale-free pattern. Scale-free networks are networks where the node degree distribution is a power law. This means that the node degree distribution decays exponentially. In other words, scale-free networks are composed of few very high clustered nodes while the rest of the nodes are not.

The most common explanation for the scale-free generation in networks is preferential attachment, also known as "the rich get richer". This implies that new nodes in the network are more likely to generate connections to current older and highly connected nodes. In the world-wide-web this phenomenon is very common. New webpages tend to create a link to older well known webpages creating in this way the scale-free architecture. In protein networks, the architecture has an evolutionary explanation. As the cellular organisms evolved from simple forms with few

chemical processes to more complex forms with more chemical processes, when a new process was created it tended to reuse the current proteins generated in the cell in combination with new proteins generated by mutation of DNA. This caused that some proteins are practically always needed for most of the chemical processes in the cell.

Figure 3.4 shows examples for scale-free (left) and small-world (right) networks. Notice that the scale-free network is composed of one very high connected node and several other nodes presenting less number of edges. On the other hand the small-world network has a structure more similar to a mesh, which is commonly called a more "democratic" system when compared with scale-free networks where the "rich get richer". Figure 3.4-bottom also shows the same networks in a circular layout. Figure 3.5 shows the node degree histograms of the networks shown in Figure 3.4. Notice the exponential fall of the scale-free network.

For the case of the brain and brain networks, some literature suggests that the brain is small-world and other authors scale-free. Both opinions might be correct for two reasons; first it has been demonstrated that scale-free networks are also small-world network, called super small-world networks, and the second reason is because from a functional and connectivity perspective the brain evolved to optimize connectivity while reducing the wiring cost, hence functional brain networks present a small-world architecture. However, recall that just as the proteins in the cell, the mammal nervous systems evolved in different stages. Our nervous system has neural structures that appeared above older ones according to our evolution. Hence newer neuronal tissue tended to create connections with older structures, creating a scale-free architecture from the nervous system perspective. Nevertheless, these topics are currently debated by the Neuroscience community, see for instance Biswal et al. (2010); Sporns et al. (2004); Bullmore and Sporns (2009); Kaiser and Hilgetag (2006); Sporns and Honey (2006).

3.1.3 Network comparisons

Besides having common network metrics it is important to know how to use them to make network comparisons. It is common in network inference research to com-

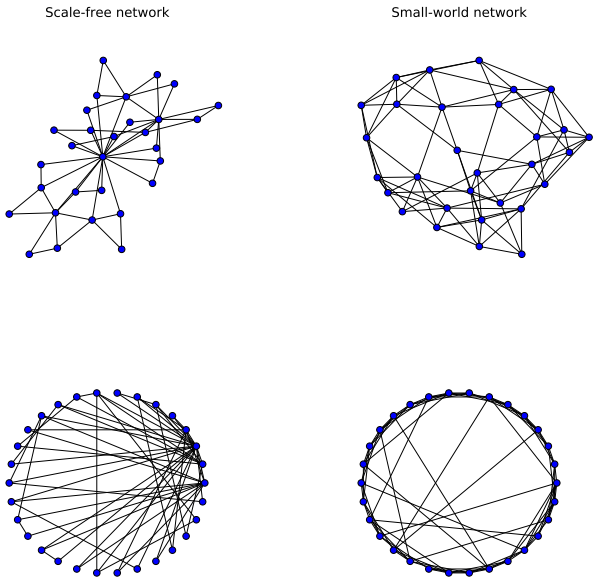

Figure 3.4: Scale-free and small-world network examples. The network at the left is a scale-free network presented in two different layouts. The layout at the top shows the clustering property of scale-free networks where few nodes are highly connected. This kind of construction is thought to occur because new nodes in the network tend to create connections to older and highly connected nodes, a generation process called "the rich get richer". On the other hand, small-world network shows a more democratic structure pattern. Both networks are composed of 30 nodes. Image programmed and created using Python module: networkx.

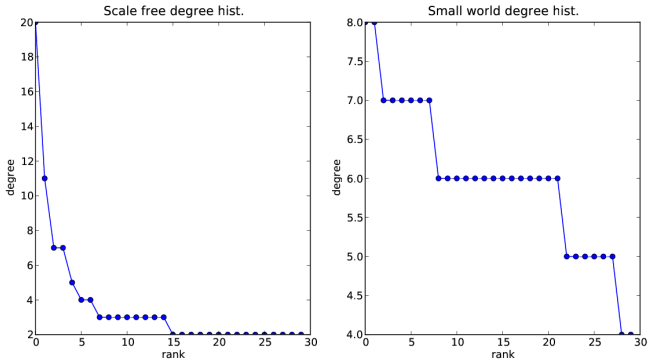

Figure 3.5: Node degree plots of the networks shown in Figure 3.4. Notice the exponential decay of the scale-free network. In the small-world network an important number of nodes have degree of 6 and difference between the highest degree and the lowest is not large, showing the democratic behaviour of this kind of network structure.

pare the inferred network using one or many techniques. Every network measure highlights different characteristic of the network depending on the model used. For this reason it is important to have a standard on network comparison in order to measure the performance of different network inference methods.

The most applied method of comparison is average degree. If two networks were inferred from the same system using two different methods, the average degree of both should be the same if structure comparison wants to be made. Another similar approach is to take the P edges with the highest weight. Notice that the small-world measure in Eq. 3.19 uses this when normalizing the clustering and the network path length with the ones of a random network which has same number of edges. In this thesis the P higher network weights are taken for comparisons among network inference techniques.

Chapter summary

This chapter gave a brief review of brain theory and network metrics highly important for brain network analysis. Since the neuronal tissue in the brain cortex and in general all the nervous system produces a massive connected network, it is important to use network theory tools to study the brain network architecture. Some network metrics are clustering which measures how well connected a node is with its neighbours, network path length which measures how far a node is from another, and also small-worldness which is a measure that highlights if the system of interest presents a small-world architecture which is a very common pattern in nature.

Small-world networks are considered the optimum tradeoff between high connectivity and wiring cost, being this the more plausible cause of why many biological systems follow this pattern including the brain. In the next two chapters connectivity inference tools will be covered, including the thesis proposal: Fourier Bayesian networks explained in Chapter 5.

Chapter 4

Standard connectivity measures

In this chapter we review measures for network edge inference widely used in the research literature. The simplest and more common of them is by far Pearson's correlation, which is still commonly used in fMRI connectivity studies. Although there are currently more sophisticated techniques, Pearson's correlation is still the best option to test as starting point.

4.1 Correlation and partial correlation

Correlation is defined as the normalized cross-covariance between two processes x and y. This relation is expressed as

$$\rho_{xy} = \frac{cov(x,y)}{\sigma_x \sigma_y} = \frac{E[(x-\mu_x)(y-\mu_y)]}{\sigma_x \sigma_y} \ , \tag{4.1}$$

where E is the expected value, and σ_x, σ_y are the standard deviations of x and y respectively. In practice since the time series x and y are finite, correlation is estimated using the sampled correlation (Stuart and Ord , 1994) defined by

$$\begin{aligned} \hat{\rho}_{xy} &= \frac{\sum_{i=1}^{M}(x(i)-\hat{\mu}_x)(y(i)-\hat{\mu}_y)}{(n-1)\hat{\sigma}_x\hat{\sigma}_y} \\ &= \frac{\sum_{i=1}^{M}(x(i)-\hat{\mu}_x)(y(i)-\hat{\mu}_y)}{\sum_{i=1}^{M}(x(i)-\hat{\mu}_x)^2 \sum_{j=1}^{M}(y(j)-\hat{\mu}_y)^2} \ , \end{aligned} \tag{4.2}$$

where M is the length of the time series x and y. Furthermore, it is possible to compute partial correlation. This measure estimates the correlation between two

sources x and y after removing the influence of a third one z. Partial correlation (Whittaker , 1990) is computed by

$$\rho_{xy\cdot z} = \frac{\rho_{xy} - \rho_{xz}\rho_{yz}}{\sqrt{1-\rho_{xz}^2}\sqrt{1-\rho_{yz}^2}} \ , \tag{4.3}$$

where ρ_{xy} is estimated using Eq. 4.2 and the notation $xy \cdot z$ has similar meaning to conditional probability and can be read as "correlation between x and y given z".

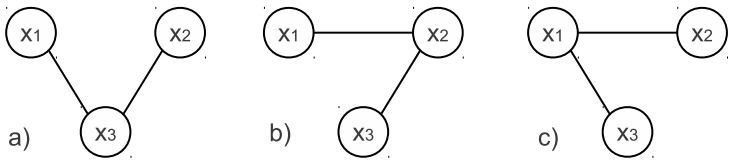

Figure 4.1: Three-node network structures. If the relation among nodes is measured using standard correlation, the real structure of the networks shown can not be inferred. However, if partial coherence is applied, it is possible to find the relation between two sources after removing the influence of a third one. This allows the inference of network structures as the ones shown in a), b) or c).

Partial correlation is very useful to estimate spurious connections. This is shown in Figure 4.1, where Figure 4.1a shows the probable connectivity found between three correlated nodes x_1, x_2 and x_3. Nevertheless this structure can also be explained by Figure 4.1b and c. Partial correlation is able to infer which of them is true. For instance, if $\rho_{12\cdot 3} = 0$ this means that structure a) is true, or if $\rho_{23\cdot 1} = 0$, then option c) will be the true structure.

If we have a vector of times series $\mathbf{x} = [x_0\ x_1\ \ldots\ x_N]$ we can define the correlation matrix $Corr$ as

$$Corr = \begin{bmatrix} \rho_{11} & \cdots & \rho_{1N} \\ \vdots & \ddots & \vdots \\ \rho_{N1} & \cdots & \rho_{NN} \end{bmatrix} \tag{4.4}$$

where $\rho_{ii} = 1$ for $i = 1, \ldots, N$. Using the correlation matrix $Corr$ it is also possible to compute partial correlation between two sources x and y after removing the influences of all the rest of sources in the matrix. This is commonly written

as $\rho_{xy \cdot N/xy}$, where N/xy means "all N sources but not counting x and y". We can verify this with a simple 3×3 correlation matrix example.

Suppose we have the following correlation matrix

$$Corr = \begin{bmatrix} \rho_{11} & \rho_{12} & \rho_{13} \\ \rho_{21} & \rho_{22} & \rho_{23} \\ \rho_{21} & \rho_{32} & \rho_{33} \end{bmatrix} , \tag{4.5}$$

and we compute its inverse which is

$$Corr^{-1} = \frac{1}{\det \mathbf{Corr}} \begin{bmatrix} 1-\rho_{23}\rho_{32} & \rho_{13}\rho_{32}-\rho_{12} & \rho_{12}\rho_{23}-\rho_{13} \\ \rho_{31}\rho_{23}-\rho_{21} & 1-\rho_{31}\rho_{13} & \rho_{21}\rho_{13}-\rho_{23} \\ \rho_{21}\rho_{32}-\rho_{31} & \rho_{12}\rho_{31}-\rho_{32} & 1-\rho_{12}\rho_{21} \end{bmatrix} , \tag{4.6}$$

Normalizing the diagonal of the previous matrix to 1, gives the following expression

$$Corr_{xy \cdot N/xy} = \begin{bmatrix} 1 & -\frac{\rho_{12}-\rho_{13}\rho_{32}}{\sqrt{1-\rho_{23}^2}\sqrt{1-\rho_{31}^2}} & -\frac{\rho_{13}-\rho_{12}\rho_{23}}{\sqrt{1-\rho_{31}^2}\sqrt{1-\rho_{21}^2}} \\ -\frac{\rho_{12}-\rho_{31}\rho_{23}}{\sqrt{1-\rho_{23}^2}\sqrt{1-\rho_{31}^2}} & 1 & -\frac{\rho_{23}-\rho_{21}\rho_{13}}{\sqrt{1-\rho_{12}^2}\sqrt{1-\rho_{31}^2}} \\ -\frac{\rho_{31}-\rho_{21}\rho_{32}}{\sqrt{1-\rho_{23}^2}\sqrt{1-\rho_{12}^2}} & -\frac{\rho_{32}-\rho_{12}\rho_{31}}{\sqrt{1-\rho_{13}^2}\sqrt{1-\rho_{12}^2}} & 1 \end{bmatrix} , \tag{4.7}$$

Notice that the partial correlation notation $xy \cdot N/xy$ is used instead of the matrix inverse. Comparing the elements of the partial correlation matrix in Eq. 4.7 with Eq. 4.3 we see that these expressions coincide but with negative values, from which their modulus can be computed to obtain the absolute values.

Once the partial correlation matrix has been estimated, it is possible to build a connectivity matrix $\mathbf{C}$ using only the matrix values that are statistically significant. This can be achieved by applying Fisher's transformation for partial correlation,

$$z_\rho = \frac{1}{2} \ln \left(\frac{1+\rho_{xy \cdot \mathbf{z}}}{1-\rho_{xy \cdot \mathbf{z}}} \right) , \tag{4.8}$$

where $\rho_{xy \cdot z}$ represents the estimated partial correlation between nodes x and y after removing the influence of the nodes in vector $\mathbf{z}$ and z_ρ is the transformed partial correlation coefficient. Then, we can reject the null hypothesis (independent sources) with a significance level α if

$$\sqrt{M - size(\mathbf{z}) - 3} \times z_\rho > \Phi^{-1}(1-\alpha/2) \tag{4.9}$$

where $size(\mathbf{z})$ represents the number of elements in vector $\mathbf{z}$, M is the length of the time series and Φ^{-1} is the Gaussian cumulative distribution function (cdf).

4.1.1 Time lagged partial correlation matrix

A method to infer causality using the partial correlation matrix concept is using a time lagged version of it, the time-lagged partial correlation matrix (Peraza and Halliday , 2010a) . This matrix is computed by concatenating the time series vector $\mathbf{x}$ with a delayed version of itself. This is written as

$$\mathbf{x}_c = [x_1(t)\ x_2(t)\ \ldots\ x_N(t)\ x_1(t-1)\ x_2(t-1)\ \ldots\ x_N(t-1)]\ . \tag{4.10}$$

This step is similar to network unfolding in Bayesian networks, see Section 5.3. Using the concatenated version of the data vector we compute the partial correlation matrix

$$Corr_{xy\cdot N/xy} = \left[\begin{array}{c|c} \mathbf{A_{t,t}} & \mathbf{B_{t,t-1}} \\ \hline \mathbf{B_{t-1,t}} & \mathbf{A_{t-1,t-1}} \end{array}\right]\ . \tag{4.11}$$

Notice that the resultant matrix can be divided in four submatrices. Matrix $\mathbf{A}_{t,t}$ represents the partial correlation between the first vector x at time t, $\mathbf{A}_{t-1,t-1}$ has the partial correlations between the nodes of the delayed $\mathbf{x}$ at time $t-1$, and $\mathbf{B}_{t-1,t}$ has the partial correlations between both vectors $\mathbf{x}(t)$ and $\mathbf{x}(t-1)$, this means that the elements in $\mathbf{B}_{t-1,t}$ are estimates of the edges that cross between the time gap, from $t-1$ to t showing causality. We can visualize this with the example shown in Figure 4.2.

Figure 4.2 shows a network composed of 11 nodes which was defined as a multivariate autoregressive process of order 1 (MVAR(1)). Figure 4.3 shows the same network in an extended version which shows the delay in the defined MVAR system. Using matrix $\mathbf{B}_{t-1,t}$ it is possible to infer the inter-slice edges taking the most statistically significant elements in $\mathbf{B}_{t-1,t}$, as explained in the previous section. Causality is then defined by the construction of vector $\mathbf{x}_c$ from $t-1$ to t.

Notice also that matrix $\mathbf{B}_{t,t-1}$ has no physical meaning and can be ignored. For an application of the time-lagged partial correlation matrix in network structure inference see Peraza and Halliday (2010a).

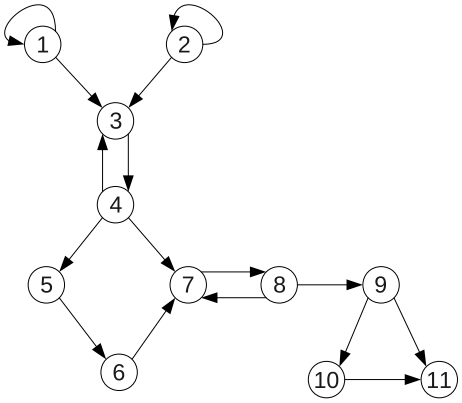

Figure 4.2: Multivariate autoregressive network system as originally published in Peraza and Halliday (2010a). This system is used as example test for the time-lagged partial correlation matrix.

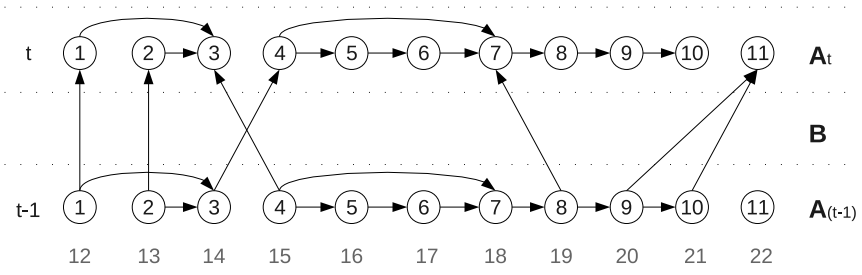

Figure 4.3: Same network system shown in 4.2 but here the network has been unfolded one time delay. Also, the inter-slice edges between both network, the original and delayed version, are shown.

4.1.2 Correlation function

The *correlation function* defines how related two time series are through time. This is commonly defined as $\rho_{xy}(\tau)$, where τ is the delay index between time series x and y. In different words it is the correlation between $x(t)$ and $y(t-\tau)$. If the maximum of $\rho_{xy}(\tau)$ is not at $\tau = 0$, there is causal relation between x and y.

The correlation function $\rho_{xy}(l,k)$ is defined by

$$\rho_{xy}(l,k) = E[x(l)y(k)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xyp_{lk}(x,y)dxdy \ , \tag{4.12}$$

where $p_{lk}(x,y)$ represents the second order probability density function or pdf (León-Garcia , 1994), and l,k are time indices. Because the processes x and y are assumed as stationary, l and k can be substituted by $\tau = l - k$ which defines the delay index, leading to

$$\rho_{xy}(\tau) = E[x(k+\tau)y(k)] \ . \tag{4.13}$$

Thus, $\rho_{xy}(\tau)$ is a function of the time delay τ. The *autocorrelation*, $\rho_{xx}(\tau)$ is defined in the same manner as Eq. 4.12, but here it is wanted to know how correlated a signal x is with itself. Autocorrelation is expressed by

$$\rho_{xx}(\tau) = E[x(k+\tau)x(k)] = E[x(k)x(k+\tau)] = \rho_{xx}(-\tau) \ . \tag{4.14}$$

Notice that autocorrelation in Eq. 4.14 is always an even function (León-Garcia , 1994) and its maximum is always at $\tau = 0$.

Because in real applications it is not possible to have infinite data segments, in order to estimate $\rho_{xy}(\tau)$ an unbiased estimator of the correlation function (Haykin , 2001) is used, which is

$$\hat{\rho}_{xy}(\tau) = \begin{cases} \frac{1}{M-|\tau|}\sum_{k=0}^{M-\tau-1} x(k+\tau)y(k) & \tau \geqslant 0 \ , \\ \\ \hat{\rho}_{xy}(-\tau) & \tau < 0 \ , \end{cases} \tag{4.15}$$

where M defines the length of the time series at hand.

4.2 Spectral synchronicity measures

Besides correlation which searches for synchronization in the time domain, it is possible to estimate a similar measure in the spectral domain and even to infer at

which frequency the relation between two sources is stronger.

The first of these techniques reviewed in this section is called coherence. Coherence is commonly seen as the spectral version of correlation and just as correlation, coherence is also a bounded measure between 0 and 1, where 1 indicates maximum synchronization or complete linear dependence. All the techniques reviewed in this section have their roots in coherence. Hence these synchronization techniques are also able to give synchronization information at specific points in the spectrum and are also bounded between 0 and 1.

4.2.1 Coherence

There is a direct relation between the autocorrelation function $\rho_{xx}(\tau)$ and the spectrum of x, f_{xx}, given by the Fourier transform of the autocorrelation function. This relation is

$$f_{xx}(\lambda) = \int_{-\infty}^{\infty} \rho_{xx}(\tau) e^{-j2\pi\lambda\tau} d\tau \ . \tag{4.16}$$

Eq. 4.16 is known as the Wiener-Khinchin theorem and it is a common tool for computing spectral densities using parametric methods (Kay , 1988). In this case the parameters are elements of the autocorrelation function ρ_{xx}. Coherency is similar in its construction to the correlation coefficient ρ as shown in Eq. 4.1 (Brillinger , 1981). This measure is defined as

$$R_{xy}(\lambda) = \frac{f_{xy}(\lambda)}{[f_{xx}(\lambda) f_{yy}(\lambda)]^{1/2}} \ , \tag{4.17}$$

where $f_{xy}(\lambda)$ is the cross-spectrum and λ is the frequency index. Coherency $R_{xy}(\lambda)$ provides a strength measure of the relation between times series x and y in the spectral domain. Also notice that $R_{xy}(\lambda)$ is a complex function which carries in its phase information about the delay between both signals. Its modulus square $|R_{xy}(\lambda)|^2$ is called *coherence*,

$$|\hat{R}_{xy}(\lambda)|^2 = \frac{|\hat{f}_{xy}(\lambda)|^2}{\hat{f}_{xx}(\lambda)\hat{f}_{yy}(\lambda)}, \tag{4.18}$$

where estimations of $\hat{f}_{xy}$, $\hat{f}_{xx}$, and $\hat{f}_{yy}$ can be computed using the fast Fourier transform (FFT) algorithm. Furthermore, spectral estimation techniques for spec-

tral smoothing such as Welch's periodogram may be applied to obtain smoother coherence and coherency estimates (Kay , 1988).

Correlation and coherency explain causality between two sources x and y by means of the phase delay between both signals. Phase delay is a common assumption in brain source causality because axonal communication among neurons does not occur at the velocity of light. For instance, the axon of a motor neuron has a conduction velocity of approximately 100 m/s (Abeles , 1991). Sometimes it is possible that two sources x and y are highly correlated with almost no phase delay between them. There are two possibilities for this scenario, one might be that x and y are really mutually connected, this means that both sources have a bidirectional communication between them, and the second scenario is that it might be a third hidden source z influencing x and y, as shown in Figure 4.4. A measure to deal with this scenario is partial coherence which is similar to partial correlation.

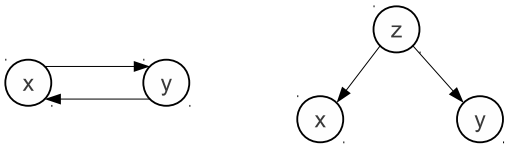

Figure 4.4: Network with hidden node. Node z is a hidden node which controls processes x and y.

4.2.2 Partial coherence

Partial coherency finds the relation in the frequency domain between two sources y and x after removing the linear influence of a third one z (Brillinger , 1981; Halliday et al. , 1995; Rosemberg et al. , 1989). Partial coherency is defined by

$$R_{xy\cdot z}(\lambda) = \frac{f_{xy\cdot z}(\lambda)}{[f_{xx\cdot z}(\lambda) f_{yy\cdot z}(\lambda)]^{1/2}} \ , \tag{4.19}$$

where

$$f_{xy\cdot z}(\lambda) = f_{xy}(\lambda) - \frac{f_{xz}(\lambda) f_{zy}(\lambda)}{f_{zz}(\lambda)}. \tag{4.20}$$

Just as normal coherency in the previous section, partial coherency also carries information about the causality between x and y in its phase. And in the same way,

its modulus square $|R_{xy\cdot z}(\lambda)|^2$ is called *partial coherence*.

Because we are assuming linearity in our system, it is possible to do multiple linear regressions as the one performed in Eq. 4.20 for multiple signals, and obtain partial coherence between x and y after removing the influence of two or more sources. For instance in Rosemberg et al. (1998) it is given the formula

$$|R_{xy\cdot wz}(\lambda)|^2 = \frac{|f_{xy\cdot wz}(\lambda)|^2}{f_{xx\cdot wz}(\lambda) f_{yy\cdot wz}(\lambda)} \ , \tag{4.21}$$

where

$$\begin{aligned} f_{xy\cdot wz}(\lambda) = f_{xy}(\lambda) - & \begin{bmatrix} f_{xw}(\lambda) & f_{xz}(\lambda) \end{bmatrix} \\ & \times \begin{bmatrix} f_{zz}(\lambda) & f_{zw}(\lambda) \\ f_{wz}(\lambda) & f_{ww}(\lambda) \end{bmatrix}^{-1} \begin{bmatrix} f_{wy}(\lambda) \\ f_{zy}(\lambda) \end{bmatrix} \ , \end{aligned} \tag{4.22}$$

for a network as the one shown in Figure 4.5. Here, it is desired to know the relation between the sources x and y after removing the influence of sources z and w.

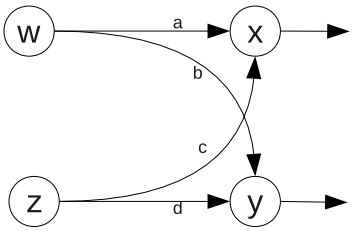

Figure 4.5: This diagram shows the relation among two input sources w and z, and two output sources x and y. Partial correlation can be applied in this case to find if x and y are correlated due to the influence of w and z or not.

In the case that w and z are independent as shown in Figure 4.5, the partial cross-spectrum in Eq. 4.22 can be simplified (Rosemberg et al. , 1998) to

$$f_{xy\cdot wz}(\lambda) = f_{xy}(\lambda) - \frac{f_{xz}(\lambda) f_{zy}(\lambda)}{f_{zz}(\lambda)} - \frac{f_{xw}(\lambda) f_{wy}(\lambda)}{f_{ww}(\lambda)} \ . \tag{4.23}$$

It is also possible to compute partial coherence in the same manner as in Eq. 4.21 for N number of sources. Partial coherency (Brillinger , 1981) is defined for the general case as

$$R_{12\cdot 3,\ldots,N}(\lambda) = \frac{f_{12\cdot 3,\ldots,N}(\lambda)}{[f_{11\cdot 3,\ldots,N}(\lambda) f_{22\cdot 3,\ldots,N}(\lambda)]^{1/2}} \ , \tag{4.24}$$

where

$$f_{12\cdot 3,\ldots,N}(\lambda) = f_{12}(\lambda) - [f_{13}(\lambda) \ \cdots \ f_{1N}(\lambda)] \times \begin{bmatrix} f_{33}(\lambda) & \cdots & f_{3N}(\lambda) \\ \vdots & \ddots & \vdots \\ f_{N3}(\lambda) & \cdots & f_{NN}(\lambda) \end{bmatrix}^{-1} \begin{bmatrix} f_{32}(\lambda) \\ \vdots \\ f_{N2}(\lambda) \end{bmatrix}, \tag{4.25}$$

and the source indices x, y were replaced by indices from 1 to N. Hence, Eq. 4.24 computes partial coherence between sources 1 and 2 after removing the linear influences from the rest of the available sources, $3, ..., N$, and the same can be defined for any other pair of sources in N. Notice that Eq. 4.25 is in the frequency domain, therefore a matrix inversion is required at every frequency component λ.

4.2.3 Imaginary coherence

Imaginary coherence was proposed in Nolte et al. (2004) as an option to avoid volume conduction in EEG and MEG recordings. What it does is to take the imaginary part of coherency, $R_{12}(\lambda)$. The quasi-static approximation of the Maxwell's equations proposes that there is no time delay in the currents transmitted by the scalp, recall Eq. 2.1. As a consequence of this, if two signals have instant synchronicity, coherency will not show any imaginary part. Therefore, it is assumed that volume conduction does not cause time delay and it will not be reflected on the imaginary part of coherency.

Having two time series x and y, it is possible to represent their Fourier transform as

$$\begin{aligned} f_x &= a_x e^{(-j2\pi\lambda\tau_x)} \\ f_y &= b_y e^{(-j2\pi\lambda\tau_y)} \ . \end{aligned} \tag{4.26}$$

Hence, the cross spectrum is defined by

$$f_{xy} = a_x b_y e^{(-j2\pi\lambda\Delta_\tau)} \ , \tag{4.27}$$

where $\Delta_\tau = \tau_x - \tau_y$ is the phase difference between both signals x and y. As shown in Nolte et al. (2004), coherency can be expressed as

$$R_{xy}(\lambda) = \frac{|f_{xy}(\lambda)|}{[f_{xx}(\lambda) f_{yy}(\lambda)]^{1/2}} e^{(-j2\pi\lambda\Delta_\tau)} \ . \tag{4.28}$$

Here, the statistic $Ph_{xy} = e^{-j2\pi\lambda\Delta_\tau}$ defines *synchronicity* between x and y. Also if x and y are independent, Ph_{xy} will be a uniform random measure whose mean value will be zero and imaginary coherency will also be zero.

In two subsequent papers, Nolte and colleagues published that imaginary part of coherence can not solve the volume conduction problem, since as normal coherence this measure is also affected by the amplitudes of the analysed sources and in some cases it provides much worst results than standard coherence (Wheaton et al. , 2005; Stam et al. , 2007).

Coherence, coherency and its imaginary part, imaginary coherence are non-parametric techniques because the only assumption that is taken is stationarity of the process, at least in the wide sense (mean and variance are constant). Another technique that infers relations among signals is *partial directed coherence* (PDC) and this is a parametric technique based on the multivariate autoregressive model (MVAR).

4.2.4 Partial directed coherence

Granger's causality (Granger , 1969) is based on the autoregressive (AR) model for stationary random signals. The AR model is defined as a FIR filter whose weights are the AR coefficients with unit variance Gaussian noise η as the filter's input. The order of the model p is defined by the number of AR coefficients. Suppose that the discrete signals x_1 and x_2 are defined by the following equations

$$\begin{aligned} x_1(n) &= \sum_{k=1}^{p} a_{11}(k)x_1(n-k) + \sum_{k=1}^{p} a_{12}(k)x_2(n-k) + \eta_1(n) \ , \\ x_2(n) &= \sum_{k=1}^{p} a_{21}(k)x_1(n-k) + \sum_{k=1}^{p} a_{22}(k)x_2(n-k) + \eta_2(n) \ , \end{aligned} \tag{4.29}$$

where $a_{11},\ a_{12}, ..., a_{22}$ are the AR coefficients and n is the discrete time index. Eq. 4.29 describes a multivariate AR (MVAR) process and it is important to notice that the MVAR model can also model bidirectional causality thanks to the AR coefficients a_{21} and a_{12}. These coefficients regulate the causal relation between both AR processes.

Expanding the MVAR model to N number of processes we have that

$$\begin{bmatrix} x_1(n) \\ \vdots \\ x_N(n) \end{bmatrix} = \sum_{k=1}^{p} \mathbf{A}_k \begin{bmatrix} x_1(n-k) \\ \vdots \\ x_N(n-k) \end{bmatrix} + \begin{bmatrix} \eta_1(n) \\ \vdots \\ \eta_N(n) \end{bmatrix} . \tag{4.30}$$

Notice that $\mathbf{A}_k$ is a $N \times N$ matrix and $\mathbf{A}$ is a three-dimensional matrix $N \times N \times p$ where the dimensional index of p is controlled by k. The joint spectral density is defined by

$$\mathbf{A}_{xy}(\lambda) = \begin{cases} 1 - \sum_{k=1}^{p} a_{xy}(k) e^{-i2\pi\lambda k} & if\ x = y \\ \\ -\sum_{k=1}^{p} a_{xy}(k) e^{-i2\pi\lambda k} & otherwise \end{cases} \tag{4.31}$$

In this case we have used x and y as indices to denote the rows and columns of matrix $\mathbf{A}$, thus $x = 1, .., N$ and $y = 1, ..., N$. With the definition of Eq. 4.31, partial directed coherence (PDC) (Baccala , 2001; Schelter , 2005; Kaminski et al. , 2001) is defined by

$$PDC_{x \leftarrow y}(\lambda) = \frac{|\mathbf{A}_{xy}(\lambda)|}{\left[\sum_{m=1}^{N} |\mathbf{A}_{my}(\lambda)|^2\right]^{1/2}} , \tag{4.32}$$

where $x \leftarrow y$ indicates causality. Eq. 4.32 computes the partial directed coherence using the MVAR coefficients, thus the PDC solution relies on the estimation of the MVAR coefficients. In practice this can be done using the well known Nuttal-Strand algorithm which is available for Matlab (Schlogl , 2006). In order to show how PDC works, an example is reproduced here from Baccala (2001) using MVAR synthesized signals. Let's define the MVAR equations for a system as the one shown in Figure 4.6 as

$$\begin{aligned} x_1(n) &= 0.95\sqrt{2}x_1(n-1) - 0.9025x_1(n-2) + \eta_1(n) \\ x_2(n) &= 0.5x_1(n-2) + \eta_2(n) \\ x_3(n) &= -0.4x_1(n-3) + \eta_3(n) \\ x_4(n) &= -0.5x_1(n-2) + 0.25\sqrt{2}x_4(n-1) + 0.25\sqrt{2}x_5(n-1) + \eta_3(n) \\ x_5(n) &= -0.25\sqrt{2}x_4(n-1) + 0.25\sqrt{2}x_5(n-1) + \eta_5(n) \end{aligned} \tag{4.33}$$

The PDC result using the estimator in Eq. 4.32 is shown in Figure 4.7. In this figure the rows show the outputs of the MVAR system and the columns the inputs.

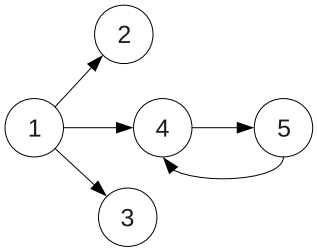

Figure 4.6: 5-node MVAR network originally published by Baccala (2001). This network is mathematically described by Eq. 4.33.

It can be appreciated that the diagonal shows a high self-PDC which is consistent with the model in Eq. 4.33.

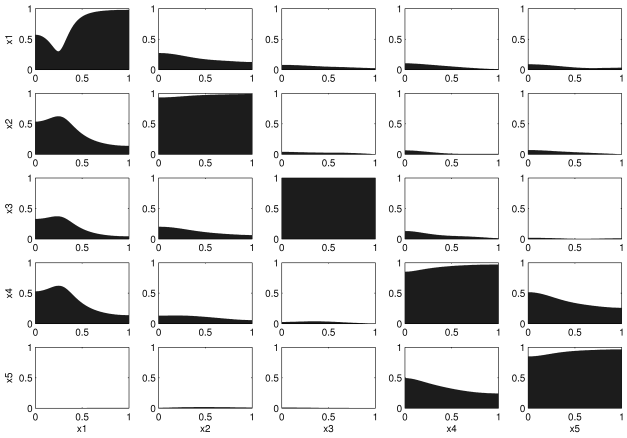

Figure 4.7: PDC estimation for the network in Figure 4.6. The MVAR signals were synthesized and the AR coefficients estimated using the Nuttal-Strand algorithm.

PDC has been used in many applications like gene regulatory networks (GRN) and naturally brain networks. Its main advantage is that the Nuttal-Strand algorithm is a very efficient and fast estimator for the MVAR coefficients, thus the computation of PDC is also fast. Some of its disadvantages are that as the number of sources N increases the accuracy of the Nuttal-Strand algorithm decreases. Furthermore, the Nuttal-Strand algorithm depends on long signal segments to estimate correctly the MVAR coefficients, and in some real applications long data segments are not available. Another objection against PDC is that it is a parametric technique, meaning that it relies on the signal model. Therefore, if the signal does not

follow a MVAR model, PDC will not be able to find an accurate result.

In signal processing there is a rule of thumb to know if a signal can be modelled as an AR process by just estimating the power spectrum using non-parametric techniques, like Welch's periodogram (Kay , 1988). If the estimated spectrum is *narrowband*, say it has a well defined peak, then the signal can be modelled as an AR process.

4.2.5 Phase coherence and phase lag index

Instead of computing the imaginary part of coherence only, it is much better to estimate the phase using the imaginary and real part of the signal spectrum. Although this can be done directly from the estimated phase spectrum in $f_{xy}(\lambda)$, some authors find also useful to estimate this phase difference in the time domain. Phase coherence (PC) and phase lag index (PLI) use instant phase difference estimation to find synchronicity and instead of the Fourier transform which finds the Fourier coefficients at all Nyquist frequency values, the Hilbert transform is used.

PC comes from the concept of phase locking or phase synchrony as explained in Tass et al. (1998); Mormann et al. (2000). Here it is desired to measure if there is consistency between the phase difference $\Delta\phi_{n,m}(t)$ of two time series x and y. This phase difference is commonly represented in the literature as

$$|\Delta\phi_{n,m}(t)| = |n\phi_x(t) - m\phi_y(t)| < const, \tag{4.34}$$

where n and m are integers, and $\phi_{x,y}$ are the phase of time series x and y. Also notice that $\Delta\phi_{n,m}(t)$ depends on the time index t, which indicates that phase consistency is being searched through time and not frequency. When this concept is applied to Electrophysiology, the isofrequency case (where $n = m = 1$) is of most interest (Tass et al. , 1998; Quiroga et al. , 2002). The difference in Eq. 4.34 behaves as a stochastic variable on the unit circle whose angle difference will tend to a constant value if both times series are synchronized. Using this phase difference consistency, phase coherence is defined by

$$PC = \left| \frac{1}{M} \sum_{l=0}^{M-1} e^{i\Delta\phi(t_l)} \right| . \tag{4.35}$$

Notice that PC depends only on the phase difference. This makes PC insensitive to the amplitude of both time series and it is not affected by this as are coherence and imaginary part of coherence (Stam et al. , 2007).

The most applied method to extract instantaneous phase for PC is by using the analytical signal

$$x^H(t) = x(t) + i\tilde{x}(t), \tag{4.36}$$

where x^H is complex valued, $x(t)$ is the original time series, and $\tilde{x}(t)$ is its Hilbert transform. Hence, $x^H(t)$ can be represented as

$$x^H(t) = A(t)e^{i\phi(t)} \ , \tag{4.37}$$

where

$$A(t) = \sqrt{[x(t)]^2 + [\tilde{x}(t)]^2} \ , \tag{4.38}$$

and

$$\phi(t) = \arctan \frac{\tilde{x}(t)}{x(t)} \ . \tag{4.39}$$

Figure 4.8 shows phase histograms of the real and imaginary part of phase differences between two signals under three conditions. The first column shows the case of two correlated signals with no delay induced, simulating in this manner a volume conduction case. In the second column a signal delay was added, and the third column shows phase difference between two independent signals. Notice how the phase difference histogram in the correlated case is centred at zero while the independent source case shows a histogram with uniform distribution.

Just as PC, another measure that uses phase difference is phase lag index or PLI. PLI was proposed in Stam et al. (2007) as a measure for synchronicity which also avoids the volume conduction problem in EEG recordings. The logic of PLI is based on the fact that the imaginary part of coherency can not be explained by volume conduction. In consequence the imaginary part reflects true interaction among different brain regions, and because it uses phase information as PC, PLI is also immune to the source amplitude problem (Guevara et al. , 2005). PLI is defined by

$$PLI = \left| \frac{1}{M} \sum_{l=0}^{M-1} \mathtt{sign}(\Delta\phi(t_l)) \right| \ . \tag{4.40}$$

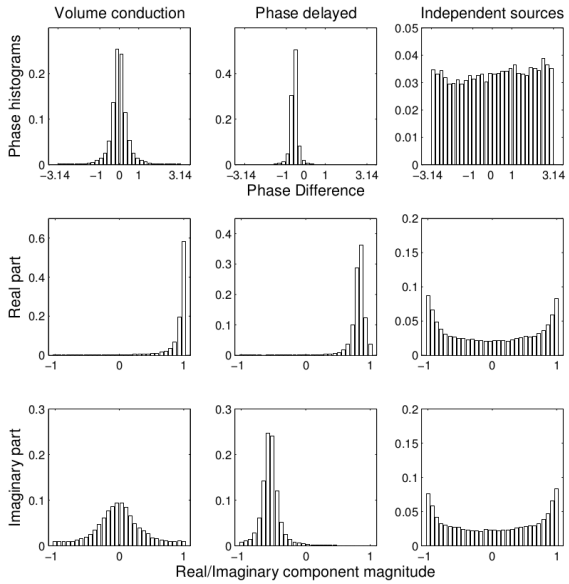

Figure 4.8: Phase histograms of dependent and independent signals. The first column shows the phase histograms of two independent sources affected by volume conduction, which is a common problem in EEG. The second column shows the phase difference between two truly correlated and delayed signals. The third column shows phase histograms of two independent signals.

Because of its close relation to PC, PLI can also be expressed as

$$PLI = \left| \frac{1}{M} \sum_{l=0}^{M-1} \mathtt{sign}(\mathtt{Im}(e^{i\Delta\phi(t_l)})) \right|. \tag{4.41}$$

Notice PC and PLI are bounded measures between 0 and 1.

Although PLI was designed to avoid the volume conduction and the source amplitude problem, there are more issues for synchronicity measures to solve in Electrophysiology. For instance EEG may use a bipolar montage to acquire brain signals. This means that EEG records the electrode of interest above the area of the scalp where it is desired to obtain brain activity but also the activity of a reference which is needed to obtain a differential voltage. This reference is commonly attached to the ears. In the EEG 10/20 system the ears are labelled as A1 and A2.

Figure 4.9 shows the effects of the reference at different magnitudes. The x-axis at each graph of Figure 4.9 shows reference/source ratio. It can be seen that all measures PLI and PC are highly affected by the reference as its amplitude becomes higher.

Synchrony is highly affected by the EEG reference which might cause spurious

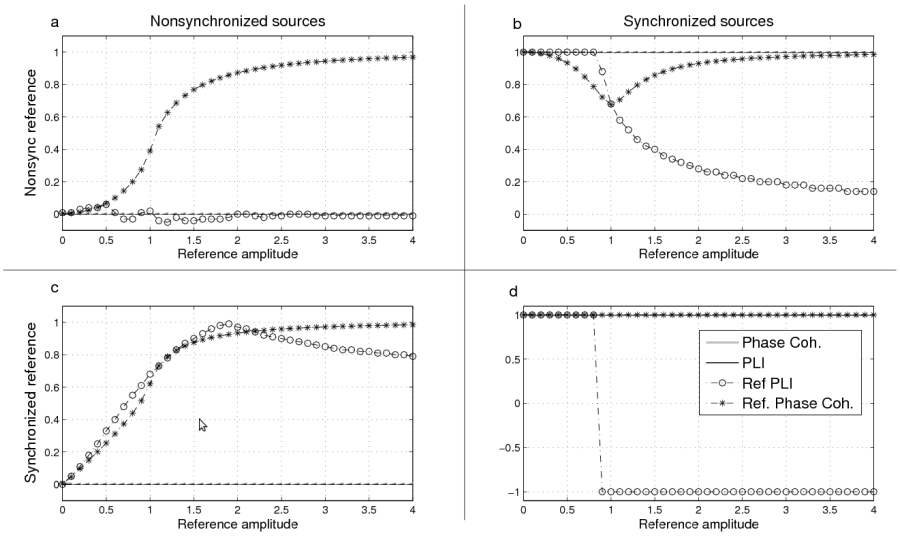

Figure 4.9: Effect produced by a common reference on synchronicity measures. A synchronized reference means that the reference has synchrony with one of the sources, and non-synchronized reference means that the reference is independent of the sources. The figure shows the behaviour of two sensor acquisition using a bipolar mounting which is normal in EEG tests where the reference electrode is common for both scalp electrodes. Ref-PLI means PLI computations using electrodes with common reference and similarly for Ref-Phase Coh.

synchronization. The reference issue is discussed in Guevara et al. (2005) for EEG where the phenomenon is studied with no apparent solution to it. Nevertheless it is possible to diminish the effect of the bipolar recording by re-referencing the EEG channels digitally and use the EEG average as the common reference, also known as the average montage. We can see this by analysing the EEG recordings with volume conduction.

Volume conduction produces a linear mixing of the cortical sources S, which can be represented by

$$\mathbf{X} = \mathbf{HS} \ , \tag{4.42}$$

where $\mathbf{H}$ is the matrix that defines the head volume conductivity. Now, if an EEG recording is made using a common reference, *ref*, we can represent the acquired set of bipolar signals $\mathbf{X}_B$ as

$$\mathbf{X}_B = \mathbf{X} - ref \ . \tag{4.43}$$

By referencing digitally the EEG channels to the average EEG we have that

$$\begin{aligned} \mathbf{X}_A &= \mathbf{X} - ref - \frac{1}{N}\sum_{n=1}^{N} x_{Bn}(t) \\ &= \mathbf{X} - \frac{1}{N}\sum_{n=1}^{N} x_n(t) \ , \end{aligned} \tag{4.44}$$

where $\mathbf{X}_A$ are EEG signals using average reference. Eq. 4.44 shows that average reference in EEG solves in part the bipolar synchrony problem mentioned in Guevara et al. (2005) by cancelling the references, but volume conduction still persists.

4.3 Behaviour of standard techniques on volume conduction

In network inference using techniques such as Coherence, PC or PLI, volume conduction affects the estimated network by producing high clustered networks (Nolte et al. , 2004; Peraza et al. , 2012). Figure 4.10 shows the behaviour of coherence, PC and PLI when a network is inferred using EEG signals under the null hypothesis. The null hypothesis indicates that all the present sources are independent. For this simulation the four-sphere head model was implemented and 64 independent sources were simulated and placed at the brain cortex level. Then, their voltages at scalp level are recorded and from them, networks were inferred. In order to avoid a cluttered network a threshold was applied by setting the average network degree to 3 (Peraza et al. , 2012).

For coherence, the network inferred under the null hypothesis resembles a mesh, where every node is connected to its closer neighbours. Notice that since the average node degree is 3 in Figure 4.10, every node is connected to 3 or sometimes 4 neighbour nodes. PC suffers exactly the same problem. On the other hand, PLI seems to infer a random network. It was hypothesized in Peraza et al. (2012) that if PLI is immune to volume conduction as commonly stated in the literature, the networks inferred by this measure must resemble a Random network. It was shown that PLI when affected by volume conduction shows small-worldness when compared with PLI under the non-volume conduction case.

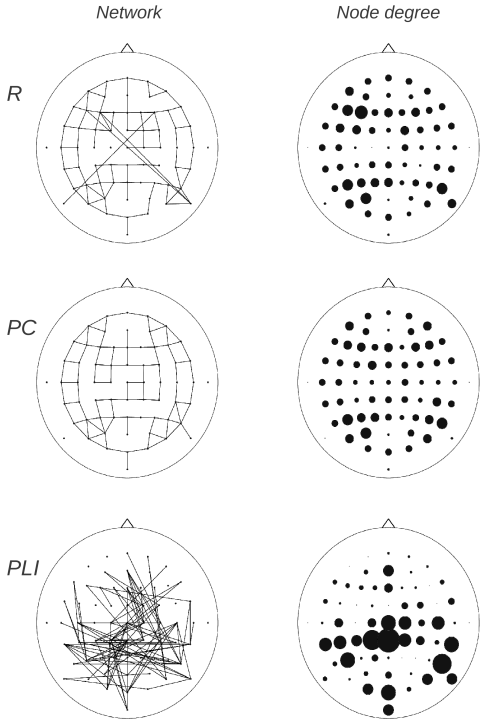

Figure 4.10: Network inference affected by volume conduction. Here coherence, phase coherence and PLI are tested for network inference in EEG. The model used was the four sphere head model to simulate volume conduction. Image adapted from Peraza et al. (2012).

Chapter summary

In this chapter some standard techniques for relation inference were reviewed. Correlation and partial correlation work in the time domain and search for synchronization through all time series' samples. Coherence techniques which include coherence itself, PDC, PLI, and PC, are able to search for synchronization at specific frequencies. PLI and PC use the Hilbert transform to find synchronization in the time domain after narrow band filtering.

PDC is a causal inference technique that also gives frequency information about connectivity based on Granger's causality and the MVAR model. PDC has been applied widely in Neuroimaging showing promising results. Its only disadvantage is that it uses the MVAR model to find connectivity, and if the analysed system does not follow this model, PDC is likely to fail.

There is another group of techniques that follow a probabilistic approach known as Bayesian networks. Bayesian networks aim to fit a network model to the time series at hand.

Chapter 5

Structure inference using Bayesian networks

In Chapter 3 networks were covered, their metrics and comparison strategies. This chapter talks about Bayesian networks (BNs), dynamic Bayesian networks (DBNs), network score functions, and network structure searching techniques widely used in this field. This chapter also presents the proposed technique of this thesis, Fourier Bayesian networks or FBNs, which use Bayesian network theory for network structure inference but the relation parents-children is inferred using Fourier transformed time series.

There are different approaches in network structure searching. The algorithms are mainly divided in stochastic and rule-based. Rule-based algorithms are the faster ones. These kind of algorithms follow a series of steps to find the best network. Most of them are based on network pattern searching, the convergent network the most used one. Stochastic algorithms are also widely used for network searching. These algorithms perform a searching from a pool of probable networks to sample those networks that describe the current data in the best way possible.

Before starting working on network inference, the researcher must decide which kind of algorithm will be used. Each technique has different properties and might require data pre-processing. For instance, it must be decided first if the algorithm will be fed with discrete or continuous valued data.

5.1 Discrete or continuous?

It is possible to work with Bayesian networks using discrete or continuous valued time series. Discrete Bayesian networks have the property of inferring causality from non-linear systems which makes them a powerful tool to analyse all kind of systems regardless of the system's model. However, when the complexity of the recorded data increases, this means if the number of discrete levels increases, the complexity of the Bayesian inference algorithm also increases due to evaluation of the probability tables. More options mean more probabilities to define and in consequence more computations.

If the amount of information in the continuous time data is so important that it requires several discrete levels, it is possible to work with the datasets directly as continuous time series. The only disadvantage of this approach is that a model should be chosen for this. The most used model in the literature is the Gaussian model, which states that the linear regression error of one or more several signals in the network system onto another signal can be represented as Gaussian noise. Then the power of the Gaussian error is taken as cost function for the analysis. The aim of this approach is to decrease this error as much as possible. Gaussian Bayesian networks lose the non-linearity property of discrete Bayesian networks but gain the advantage of avoiding discretization saving computational cost.

Figure 5.1 shows an example from Smith et al. (2002) showing a discretization approach for Bayesian networks. In Smith et al. (2002) a gene transcription simulator was implemented and from the generated time series a discretization was performed to make the time series a discrete signal with 3 levels. Then, a Bayesian network analysis using multinomial distributions was implemented to infer causality and network structure.

The same signal discretization approach implemented for genetic datasets, can also be done in Electrophysiology. For instance Smith et al. (2006) used the signal envelope of physiological signals recorded from birds' brain cortices, in specific the Zebra Finch bird. Once the power envelope is obtained it is discretized to perform multinomial Bayesian network inference. See Figure 5.2.

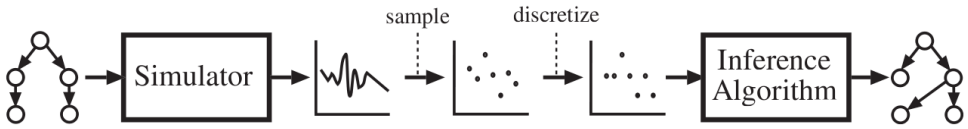

Figure 5.1: Network structure inference using discretizations. In Smith et al. (2002) a discretization procedure was performed to simulate datasets in order to use Dynamic BNs with a multinomial probability distribution. These allowed the researchers to make network structure inference on simulated gene networks. Image from Smith et al. (2002).

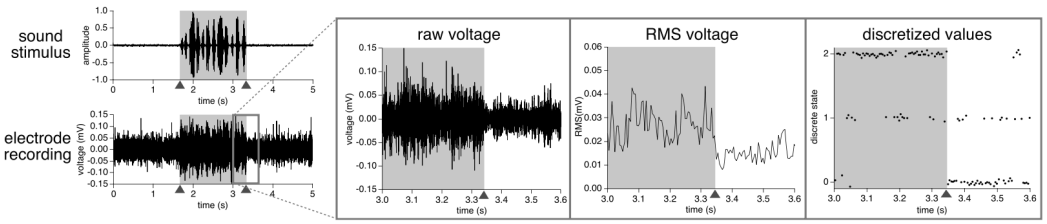

Figure 5.2: Network structure inference using envelope estimation. Here, Smith et al. (2006) estimated the envelope of the recorded time series and then the envelope was discretized. The aim of this is to apply the same Dynamic BNs algorithm as in Smith et al. (2002) but now from micro electrode array datasets recorded from the Zebra Finch bird's brain. Image from Smith et al. (2006).

5.2 Causality and Bayesian networks

When working with or studying networks sometimes it is not enough just knowing the structure that describes the interaction among the nodes, but also the *causality* of all node connections. This means to find which nodes are influenced by their neighbours and/or if they influence other nodes. Causality can be explained using one of the most classic examples and the one that will be used for the rest of the chapter, the season-slippery network (Pearl and Russel , 2001) shown in Figure 5.3.

The season-slippery network in Figure 5.3 is a causal graph composed of five nodes named *season*, *sprinkler*, *rain*, *wet*, and *slippery*. The network describes the possible causes for a person to slip due to a wet floor. Hence, the fact the floor is wet causes people to slip, and it is said that the node *wet* is cause of *slippery*. In the same way there are two reasons that explain why floor is wet. One might be that the garden sprinklers were turned on or it might be because of the rain. The example goes a little further trying to find the reason that influences nodes *sprinkler* and

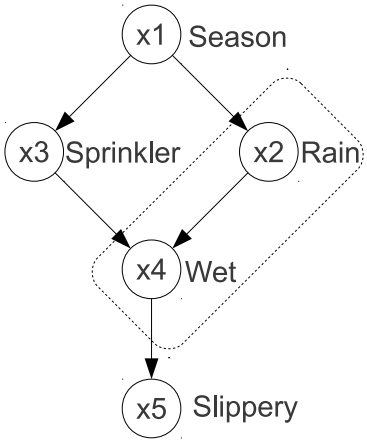

Figure 5.3: Season-slippery causal network example. The network attempts to explain the reasons of people falling because of a wet floor. The network shows two causes of this, one are the sprinklers and the other is raining. Sprinklers and rain are also controlled by season, if it is dry season it is more likely that the sprinklers will turn on, while if it is the rain season, rain is more probable. Adapted from Pearl and Russel (2001).

wet. The season influences these. If node *season* is the dry season, the sprinklers will turn on, otherwise if it is the rain season, it is more likely that the floor will be wet due to the rain.

The previous example explains in principle causality, meaning that one node or process influences another and in this way directionality is implied in the definition. For instance, in the season-slippery example it would sound awkward to say: "it rains because the floor is wet", this causal relation is very unlikely.

Causality and edge directionality is also commonly described in the literature using family analogy. For example in Figure 5.3 causality starts in node x_1 and ends in node x_5, making x_1 the highest node in the network hierarchy. Hence, we can say that x_1 is parent of x_2 and x_3 and these both are in consequence *brother nodes*. x_2 and x_3 are both parents of x_4, being this last one a parent of node x_5.

Causality can be mathematically represented using conditional probability, in the logic that knowing the cause will increase the certainty of the effect. Take for instance the subnetwork composed by the nodes *wet* and *rain* shown in Figure 5.4. If we want to compute the probability of wet floor $p(wet)$, it can be estimated by sampling the humidity of the floor during the year and taking the ratio

wet days/all days. However, imagine now that you already know that it is raining. Will this knowledge increase the probability of having a wet floor? Certainly this probability will increase and it can be represented as $p(wet|rain)$ which should be read as "the probability of wet floor *given* it is raining".

Figure 5.4: Rain wet floor subnetwork from Figure 5.3.

The realm of conditional probabilities allows to represent the causal structure of a network using mathematical representations. The probability of wet floor given it is raining can be represented as

$$p(wet|rain) = \frac{p(wet,\ rain)}{p(rain)}\ , \tag{5.1}$$

or also

$$p(wet,\ rain) = p(wet|rain)p(rain)\ . \tag{5.2}$$

Eq. 5.2 represents the joint probability of variables *wet* and *rain*. It also represents the probability of the entire subnetwork shown in Figure 5.4.

Following the same probability logic we can represent the joint probability of the entire season-slippery network in Figure 5.3 as

$$p(x_1, x_2, x_3, x_4, x_5) = \\ p(x_5|x_4, x_3, x_2, x_1)p(x_4|x_3, x_2, x_1)p(x_3|x_2, x_1)p(x_2|x_1)p(x_1)\ . \tag{5.3}$$

The property of representing a causal conditional system as a product of probabilities is called in the literature the *theorem of total probability* which is expressed mathematically as

$$p(x_1....x_N) = p(x_1)\prod_{n=2}^{N} p(x_n|x_{n-1}...x_1)\ , \tag{5.4}$$

where N is the total number of nodes in the network. Notice that the joint network probability can be factorized by nodes and as expected every node is dependant on all its ancestors. However, in conditional probability if the current state of a node's parents is known, any knowledge coming from higher ancestors will be meaningless. For instance, assume you want to compute the probability $p(slippery|wet, sprinkler, rain)$. If it is already known that the floor is wet, the current knowledge about the rain and the state of the sprinklers is not required to infer if people will slip, since we already know that the floor is wet. This is commonly called in the literature as *conditional independence*. Hence, we can say that "slippery is conditionally independent of sprinkler and rain given that floor is wet" or $p(slippery|wet, sprinkler, rain) = p(slippery|wet)$. In consequence, we can state that in a causal networks a node is only dependant on its parents. Using conditional independence we can rewrite Equation 5.3 as

$$p(x_1, x_2, x_3, x_4, x_5) = p(x_5|x_4)p(x_4|x_3, x_2)p(x_3|x_1)p(x_2|x_1)p(x_1) \ . \tag{5.5}$$

or using the theorem of total probability as

$$p(x_1...x_N) = \prod_{n=1}^{N} p(x_n|\mathbf{pa}_{x_n}) \ , \tag{5.6}$$

where $\mathbf{pa}_{x_n}$ represents the set of parents for the node x_n. Eq. 5.5 describes the structure of the season-slippery network using a probability model. This modelling becomes very useful to find network structure and its causal properties in plenty of applications, and an uncountable number of algorithms have been published using the conditional independence principle. Unfortunately, the network structure description that relates Eq. 5.5 and Figure 5.3 is not unique. This means that there might be a whole family of networks that are as equally probable as the season-slippery one. We can explain this by using Bayes' theorem which relates the conditional probability of two dependent processes A and B as

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \ , \tag{5.7}$$

where

$$p(B) = \int_A p(B|A)p(A) \ . \tag{5.8}$$

Using Bayes' theorem it is possible to find another network with the same joint probability as the season-slippery one. Let's for instance change the direction of the edge between node *sprinkler* and node *season*. This will give us the following probability modelling

$$p(x_1, x_2, x_3, x_4, x_5) = p(x_5|x_4)p(x_4|x_3, x_2)p(x_1|x_3)p(x_2|x_1)p(x_3) \ . \tag{5.9}$$

The previous change was possible due to Bayes' theorem. Recall that

$$p(x_3|x_1) = \frac{p(x_1|x_3)p(x_3)}{p(x_1)} \tag{5.10}$$

The equivalent network is shown in Figure 5.5. Notice that although its probability is exactly the same, it does not make sense that the sprinkler be the cause for the season changes. The network equivalence problem requires the creation of algorithms and strategies to find among all the family of equivalent networks the ones that make sense.

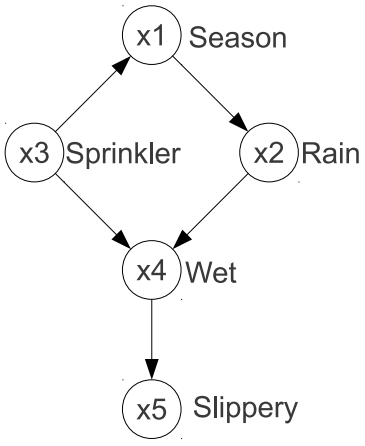

Figure 5.5: Season-slippery equivalent network. The direction of the edge between node x_1 and x_3 has been changed, producing the different network that has the same probability value than the network in Figure 5.3.

Knowing conditional probability, it is time to define the basic network structures. Any directed network can be divided by sets of three node networks. These are the

linear network, divergent network, and convergent network. These basic structures are shown in Figure 5.6, where networks a and c are linear networks, b is divergent and d is convergent. It can be proved easily using Bayes' theorem that networks a, b, and c are equivalent, this means that they have the same probability and in consequence they can not be differentiated by any algorithm. On the other hand, the convergent network d is not equivalent to a, b, or c. Its probability value is unique and for this reason almost all algorithms that infer network structure are based on the analysis of convergent networks.

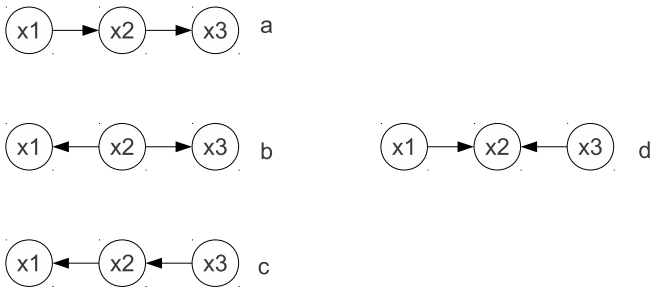

Figure 5.6: Three-node networks. The networks at the left have the same joint probability value $p(x_1, x_2, x_3)$, while the convergent network at the right has not. Many network inference algorithms in the literature focus on the searching of convergent networks due to this property.

The networks estimated using the framework explained in this section are called in the literature Bayesian networks because their analysis is based on Bayes' theorem. There is one last point that is worth mention, the concept of *cycles*. Cycles in a network are edge paths whose causality leads to a closed loop. Cycles examples are shown in Figure 5.7. The importance of cycles is that in Bayesian networks the cyclic structures are forbidden. The reason for this is because these structures do not follow the theorem of total probability which is the base of causal networks. Furthermore, the total probability theorem also causes that Bayesian networks always have at least one node with no parents and at least one node without children.

Bayesian networks are also commonly called in the literature as *static* Bayesian networks to remark the fact that the computed probability is an instantaneous probability. This means that time lags among processes are ignored and only in-

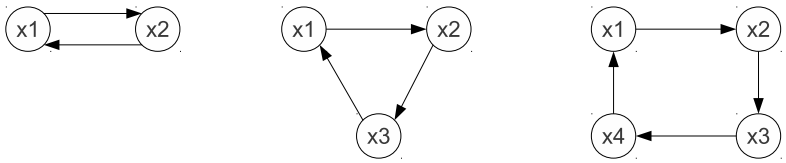

Figure 5.7: Cyclic network examples. The figure shows three examples of network cycles, which are forbidden in static network theory because this kind of structure does not obey the theorem of total probability.

stantaneous interaction is used for network inference. Nevertheless, if the researcher already knows that the analysed processes present time lagged correlation, this extra information can be used for network inference. This can be achieved by a variant of Bayesian networks called *dynamic* Bayesian networks.

5.3 Dynamic Bayesian networks

Dynamic Bayesian networks use the same principles and algorithms as static Bayesian networks but instead of using the original time series set only, a time delayed version of the original nodes is added to the network. This step is called *network unfolding* and the complete network is called the *unfolded network*.

An important property of the unfolded network is that with this arrangement now cycles are allowed. Since the delayed nodes are taken as different processes the total probability theorem is not broken by these delayed cycles. Figure 5.8 shows an example of network unfolding. Here a static network is shown as example in Figure 5.8a and its unfolded counterpart is shown in Figure 5.8b where the original network has been unfolded twice. Notice that the delay can be forwards or backwards without affecting the algorithm's performance. For instance in Figure 5.8b the unfolding was performed as $[t,\ t+1,\ t+2]$, but this can also be done backwards as $[t-1,\ t,\ t+1]$ and it represents the same system.

In the literature, every network unfolding is also called *network slice*. For instance, the network in Figure 5.8b has three slices t, $t+1$, and $t+2$ using forward delay. The edges that cross from one slice to another are called *inter slice edges*

and the edges that stay within the network slice are called *intra slice edges*. For example, in Figure 5.8 the edge between x_4 and x_5 is an intra slice edge while the rest are inter slice.

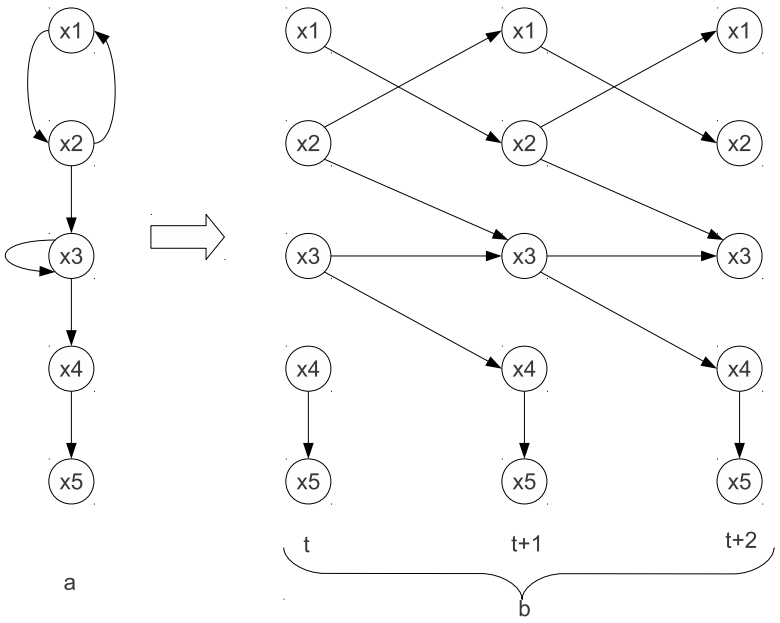

Figure 5.8: Dynamic networks and network unfolding. The figure shows an example of a static network that has two cycles, one between nodes x_1 and x_2 and the other is a self cycle in x_3. Cycles are forbidden in Bayesian network theory, but they can be represented or estimated by unfolding the static network, and making it dynamic. Dynamic networks have the disadvantage of increasing the number of nodes, which also increases exponentially the number of possible networks to infer and the computational load.

There is an important disadvantage of dynamic Bayesian networks related to the network unfolding and this is that the number of nodes in the network system is increased N nodes at every slice of the new network. This also increases the complexity of the problem and the computational cost. The number of possible networks increases exponentially with the number of nodes, making the network inference an unbearable problem if the number of nodes increases beyond allowed levels. For this reason most of the current algorithms only unfold the network once from t to $t + 1$, which for most applications seems to be enough. Another option is to constrain the network searching. For instance, if the interest is on causality only, the DBN algorithm can be designed to search for inter slice edges only and ignore the intra slice ones. This decreases the network searching space considerably. An

example of this approach can be found in Rajapakse and Zhou (2007).

5.4 Network score functions

BNs are a natural method to encode network causality and summarize the joint pdf of a system with all its variables in a directed graphical model which allows to make inferences. For these reasons, BNs have received all kind of names in the literature such as *influence diagrams*, *expert systems*, and *causal networks*. Most of the authors agree to call them Bayesian networks because of their philosophy of using all the knowledge at hand and even intelligent guesses to make inferences.

Bayes' theorem defines the probability of the system's parameters θ given the data D and the background knowledge ξ as

$$p(\theta|D,\xi) = \frac{p(\theta|\xi)\ p(D|\theta,\xi)}{p(D|\xi)} \tag{5.11}$$

where

$$p(D|\xi) = \int p(D|\theta,\xi)p(\theta|\xi)d\theta \tag{5.12}$$

$p(\theta|\xi)$ and $p(\theta|D,\xi)$ are called the prior and the posterior probability for variable θ, and $p(D|\theta,\xi)$ is known as the likelihood. Hence

$$posterior = \frac{prior \times likelihood}{normalizing\ factor}. \tag{5.13}$$

When Bayesian analysis is performed for network structure searching, the structure of the network becomes part of the system parameters. The hypothetical network structure S_h defines the interrelation among the network nodes and defines the system parameters. Hence, from now on the Bayesian probability of the hypothetical network S_h is defined as

$$p(S_h|D) = \frac{p(D|S_h)p(S_h)}{p(D)}\ , \tag{5.14}$$

where again the denominator is defined by

$$p(D) = \sum_S p(D|S_h)p(S_h)\ . \tag{5.15}$$

In all Bayesian algorithms for network structure inference, the hypothetical network S_h is proposed first to evaluate the Bayes' theorem, then if the evaluation is good or better than a previous choice, the network is accepted or rejected depending on the algorithm's rules. The proposed network S_h is then evaluated using a network score function of the form

$$Score\ function = Network\ likelihood \times Punishment\ factor\ . \tag{5.16}$$

When working with probabilities, it is more useful to work with logarithms in order to represent multiplications of several factors as sums;

$$Score = \log[Network\ likelihood] + \log[Punishment\ factor]\ . \tag{5.17}$$

The network likelihood generally gives a measure of how good the proposed structure S_h describes the data. Unfortunately, the more connected is the network the better is the data described. This means that a full connected network is able to describe all processes in the network, but a full connected network will not give useful information about the studied system. For this reason, the Occam's razor principle is applied and simpler structure models S_h are chosen instead of full connected ones.

Simpler models are found by introducing a cost factor also known as punishment factor into the network score function. The punishment factor has the task of decreasing the score of those networks that are too complex. The level of complexity can be adapted by multiplying the punishment factor by a constant which controls its influence in the score function. This procedure is commonly called in the literature as score tuning. In the following sections, well known score function are reviewed.

5.4.1 Maximum a posteriori, MAP

The maximum a posteriori is a general estimation method based on the Bayes' theorem. It is very related to the maximum likelihood estimation but here the prior distribution of the variable of interest is included. Generally the MAP estimator is found by maximizing the posterior distribution (hence the name) of a network S_h given the data. This is represented as

$$S_h \rightarrow \arg\max_{S_h} p(S_h|D) \tag{5.18}$$

which using the Bayes' theorem, becomes

$$S_h \rightarrow \arg\max_{S_h} \frac{p(D|S_h)p(S_h)}{p(D)} \tag{5.19}$$

Recall from Eq. 5.15 that $P(D)$ does not depend on S_h, hence it can not be minimized by S_h. This simplifies the MAP structure estimator to

$$S_h \rightarrow \arg\max_{S_h} p(D|S_h)p(S_h) \ . \tag{5.20}$$

Finally, the network score function using the MAP estimator becomes

$$Score_{MAP} = \log p(D|S_h) + \log p(S_h) \ . \tag{5.21}$$

In Eq. 5.21 the punishment factor is represented by the network prior probability $p(S_h)$. Here, $p(S_h)$ will regulate which networks have higher probability and which do not. For instance, it is very unlikely that the real system is either a full connected network or a full disconnected one. The disadvantage of applying Eq. 5.21 directly is that it requires the prior distribution $p(S_h)$ which might be difficult to know. Nevertheless, there are algorithms that deal with the prior distribution as explained in Section 5.8.

5.4.2 Bayesian Dirichlet equivalence, BDe

The Bayesian Dirichlet equivalence (BDe) evaluates multinomial Bayesian networks using a Dirichlet probability distribution as prior distribution. In the case of binomial variables, the Dirichlet distribution reduces to a Beta distribution which is a special case where the analysed variables have only two values. Using the Beta distribution and Bayes' theorem, the posterior probability of a binomial process can be represented as

$$p(\theta|D,\xi) = \frac{\theta^h(1-\theta)^t p(\theta|\xi)}{p(D|\xi)} \ , \tag{5.22}$$

where h and t are the available known data D, the number of heads and tails outcomes from the tossing process respectively. The prior Beta distribution, which

gives the prior knowledge on h and t is defined by

$$p(\theta|\xi) = Beta(\theta|\alpha_h, \alpha_t) = \frac{\Gamma(\alpha)}{\Gamma(\alpha_h)\Gamma(\alpha_t)}\theta^{\alpha_h - 1}(1-\theta)^{\alpha_t - 1} \ , \tag{5.23}$$

where α_h and α_t are the distribution parameters and are known in Bayesian analysis as *hyperparameters*, $\alpha = \alpha_h + \alpha_t$, and Γ is the *Gamma* function. The hyperparameters reflect the current knowledge about the behaviour of the system. For instance, if we assume that a coin that is being tossed is fair, we can use a perfectly symmetric Beta function or even if we do not known anything about the coin, we can assume a uniform distribution as prior (Neapolitan , 2004; Heckerman , 1996; Heckerman et al. , 1995). Using the previous defined prior and posterior distribution, the Bayes' theorem in Eq. 5.22 can be rewritten as

$$\begin{aligned} p(\theta|D,\xi) &= \frac{\Gamma(\alpha + M)}{\Gamma(\alpha_h + h)\Gamma(\alpha_t + t)}\theta^{\alpha_h + h - 1}(1-\theta)^{\alpha_t + t - 1} \\ &= Beta(\theta|\alpha_h + h, \alpha_t + t), \end{aligned} \tag{5.24}$$

where M is the total observed data $M = h + t$. Notice that the denominator in Eq. 5.22 is completely ignored in Eq. 5.24. This is because the denominator is just a *normalizing factor* that assures that the integral of the posterior probability equals one. Because Eq. 5.24 is already a fully Beta normalized distribution the denominator can be just ignored. This is the advantage of using a Beta distribution as prior distribution. The posterior will also be a Beta distribution which can be easily normalized. Furthermore, the expectation of a Beta process is also a well known result

$$\int \theta Beta(\theta|\alpha_h, \alpha_t)d\theta = \frac{\alpha_h}{\alpha} \tag{5.25}$$

Using the previous equation, the expectation of Eq. 5.24 will give us the Bayesian probability that the next toss results in heads,

$$p(x_{M+1} = heads|D,\xi) = \frac{\alpha_h + h}{\alpha + M} \ . \tag{5.26}$$

The Beta prior distribution works well for the coin tossing example, but in the case of multinomial sampling where more than two options are observed (like in a dice for example), the likelihood function of this kind of processes is given by

$$p(X = x_k|\underline{\theta},\xi) = \theta_k, \quad k = 1, ..., r, \tag{5.27}$$

where $\underline{\theta} = \{\theta_2, ..., \theta_r\}$ are the multinomial probability parameters and r is the total number of outcome options (for instance, in a dice $r = 6$). From the observed data $D = \{X_1, ..., X_M = x_M\}$ we can obtain the sufficient statistic $\{M_1, ..., M_r\}$ where M_k is the number of outcomes x_k in the observed dataset. The multinomial prior distribution chosen for this problem is the Dirichlet distribution (Neapolitan , 2004; Heckerman , 1996), which is nothing else than a Beta distribution generalization. The Dirichlet distribution is defined by

$$p(\underline{\theta}|\xi) = Dir(\underline{\theta}|\alpha_1, ..., \alpha_r) = \frac{\Gamma(\alpha)}{\prod_{k=1}^{r} \Gamma(\alpha_k)} \prod_{k=1}^{r} \theta_k^{\alpha_k - 1} \ , \tag{5.28}$$

where $\alpha = \sum_{k=1}^{r} \alpha_k$. As in the binomial example, the posterior distribution will also be a Dirichlet probability

$$p(\underline{\theta}|D, \xi) = Dir(\underline{\theta}|\alpha_1 + M_1, ..., \alpha_r + M_r). \tag{5.29}$$

As its name suggests, the BDe score is based on the Dirichlet distribution and its derivation is fully presented in Heckerman et al. (1995). The BDe is defined for a network structure S_h as

$$BDe(S_h) = p(D|S_h) = \prod_{i=1}^{N} \prod_{j=1}^{O_v} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + M_{ij})} \prod_{k=1}^{r_v} \frac{\Gamma(\alpha_{ijk} + M_{ijk})}{\Gamma(\alpha_{ijk})} \tag{5.30}$$

where i is the index of current node x_i, O_v is the total number of combinations (instantiations) of the parents of x_i, and r_v is the total number of possible values of x_i.

The BDe score is the most common score function applied in Bayesian networks. Notice that O_v and r_v depends on how many values the variables in the system have. Hence a discretized signal with several discrete levels will increase the complexity of the BDe evaluation by increasing the number of combinations in O_v and r_v. For this reason many of the current applications do not use discrete signals with multiple levels.

To evaluate networks using the BDe score function, a MAP approach can be followed or any other approach for structure searching like Markov chain Monte Carlo or stochastic hill climbing (Neapolitan , 2004). More about this is discussed in Section 5.8

5.4.3 Bayesian information criterion, BIC

The Bayesian information criterion or BIC score function is also widely used in network inference. The BIC score function is defined as

$$BIC(S_h|D) = \log_2 p(D|S_h, \hat{\xi}_s) - \frac{size(S_h)}{2} \log_2(M) \ , \tag{5.31}$$

where $P(D|S_h, \xi_s)$ is the likelihood function and the second term is the punishment factor which is proportional to the number of edges in the network.

The BIC score using Dirichlet distributions (Jensen and Nielsen , 2007) is mathematically expressed by

$$BIC(S_h|D) = \sum_{i=1}^{N} \left[\sum_{j=1}^{O_v} \sum_{k=1}^{r_v} M_{ijk} \log \left(\frac{M_{ijk}}{M_{ij}} \right) - \frac{1}{2} O_v (r_v - 1) \log M \right] . \tag{5.32}$$

where the terms in Eq. 5.32 are defined as in Eq. 5.30. The BIC score function can be applied alone without requiring any prior distribution or algorithms that deal with the prior function. BIC is commonly used in rule based learning approaches where a series of rules evaluates different network possibilities until the one with the highest BIC score is found.

Using the conditional independence property of Bayesian networks, the BIC score function can be factorized as

$$BIC(S_h|D) = \sum_{n=1}^{N} \left\{ \log p(x_n|\mathbf{pa}_n, S_n, \hat{\xi}_n) - \frac{size(S_n)}{2} \log(M) \right\} \ , \tag{5.33}$$

where $\mathbf{pa}_n$ is the vector of parents of node x_n, and S_n is a convergent subnetwork formed by x_n and its parents $\mathbf{pa}_n$.

A common practice using BIC is to add a *tuning parameter* for the punishment factor, which leads to the following BIC score

$$BIC(S_h|D) = \sum_{n=1}^{N} \left\{ \log p(x_n|\mathbf{pa}_n, S_n, \hat{\xi}_n) - alpha \frac{size(S_n)}{2} \log(M) \right\} \ . \tag{5.34}$$

The tuning parameter *alpha* allows the researcher to modify the punishment level during the network searching, see for instance Varshavsky et al. (2007).

5.5 Gaussian model

BDe and multinomial BIC score functions are designed for multinomial discrete data. If the data at hand are time series sampled from continuous variables one option is to round the samples to assigned number of levels to create multinomial variables. This approach has two main problems. To round the time series to create multinomial variables destroys valuable information from the original sampled recordings and the second problem is that the size of the probability tables in multinomial Bayesian network for either network estimation or prediction increases exponentially with the number of multinomial levels, which increases the complexity of the original problem.

Instead of using time series rounding to create multinomial variables, another approach is to use *continuous time BNs*, which use a Gaussian model to describe data. The Gaussian model relies on a linear regression from the parent nodes into the child node. In other words, what it is done here is to regress the probable parents into the child in order to decrease the regression error. The parents that decrease this regression error, might be assumed to be the true parents of the child node.

The disadvantages for the Gaussian model in Bayesian networks is that the time series should be stationary and the regression error must follow a Gaussian distribution, also the nonlinear property of multinomial BNs is lost. Fortunately the stationary assumption is very common in many current algorithms and does not represent an issue, and Gaussian processes are the most common processes in nature, which make Gaussian BNs applicable to a broad range of problems.

In a Gaussian model what we want to find is the conditional probability $f(x_1|x_2)$ of a network as the one shown in Figure 5.9. Notice that now f is used to define a continuous probability distribution instead of p which was used for discrete variables. Figure 5.9 shows a basic model from which it is desired to know how much information in x_2 is able to explain the behaviour of x_1.

A straightforward approach to derive $f(x_1|x_2)$ is given in Anderson (2003). This derivation is reproduced in this section in extended version. First of all, let's define

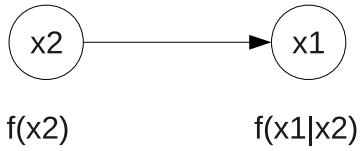

Figure 5.9: Gaussian model for BN showing conditional probability with a causal network.

the multivariate Gaussian distribution,

$$G(\mathbf{x}|\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}) = \frac{1}{2\pi^{d/2}|\Sigma_{\mathbf{x}}|^{1/2}} e^{\left\{-\frac{1}{2}(\mathbf{x}-\mu_{\mathbf{x}})^T \Sigma_{\mathbf{x}}^{-1}(\mathbf{x}-\mu_{\mathbf{x}})\right\}}, \tag{5.35}$$

where $\Sigma_{\mathbf{x}}$ is the multivariate variance matrix, $\mathbf{x}$ is the vector of variables, $\mu_{\mathbf{x}}$ is the vector of means, and $|*|$ represents the determinant of a matrix. Suppose now that two Gaussian signals y_1 and y_2 are related by following equation;

$$\begin{aligned} y_1 &= x_1 + Bx_2 \\ y_2 &= x_2 \end{aligned} \tag{5.36}$$

and that a constant B must be found in order to leave y_1 and y_2 uncorrelated. To start this, lets compute first the covariance matrix between y_1 and y_2,

$$C(y_1, y_2) = E[(y_1 - E(y_1))(y_2 - E(y_2)^T)] \ . \tag{5.37}$$

First the expected value of Eq. 5.36 is computed,

$$\begin{aligned} E[y_1] &= E[x_1 + Bx_2] = \mu_1 + B\mu_2 \ , \\ E[y_2] &= \mu_2 \ . \end{aligned} \tag{5.38}$$

Substituting the expected values in the covariance equation, leads to

$$\begin{aligned} C(y_1, y_2) &= E[(x_1 + Bx_2 - \mu_1 - B\mu_2)(x_2 - \mu_2)^T] = 0 \ , \\ &= E[[(x_1 - \mu_1) - B(x_2 - \mu_2)][x_2 - \mu_2]^T] = 0 \\ &= E[(x_1 - \mu_1)(x_2 - \mu_2)^T + B(x_2 - \mu_2)(x_2 - \mu_2)^T] = 0 \\ &= E[(x_1 - \mu_1)(x_2 - \mu_2)^T] + BE[(x_2 - \mu_2)(x_2 - \mu_2)^T) = 0 \ . \end{aligned} \tag{5.39}$$

From Eq. 5.39 it can be noticed that the expected values are the cross-covariance between x_1 and x_2 and the variance of x_2, defined as Σ_{12} and Σ_{22} respectively. Hence the previous equation can be rewritten as

$$\Sigma_{12} + B\Sigma_{22} = 0 \tag{5.40}$$

From which B can be easily obtained

$$B = -\Sigma_{12}\Sigma_{22}^{-1} \ . \tag{5.41}$$

Using the previous result, the Gaussian signals y_1 and y_2 in Eq. 5.36 can be redefined as

$$\begin{aligned} y_1 &= x_1 - \Sigma_{12}\Sigma_{22}^{-1}x_2 \\ y_2 &= x_2 \end{aligned} \ . \tag{5.42}$$

The next step now is to find the covariance matrix for y_1 and y_2. Start by computing the mean vector $\mu_{\mathbf{y}}$

$$\begin{aligned} E[y_1] &= \mu_1 - \Sigma_{12}\Sigma_{22}^{-1}\mu_2 \\ E[y_2] &= \mu_2 \ . \end{aligned} \tag{5.43}$$

Expressing Eq. 5.43 in matrix notation we have that

$$E\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = E\begin{bmatrix} I & -\Sigma_{12}\Sigma_{22}^{-1} \\ 0 & I \end{bmatrix}\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} \mu_1 - \Sigma_{12}\Sigma_{22}^{-1}\mu_2 \\ \mu_2 \end{bmatrix} = \mu_{\mathbf{y}} \ . \tag{5.44}$$

The covariance can be expressed as

$$C(\mathbf{y}) = E[(\mathbf{y} - \mu_{\mathbf{y}})(\mathbf{y} - \mu_{\mathbf{y}}^T)] \ , \tag{5.45}$$

which leads to

$$C(\mathbf{y}) = \begin{bmatrix} E[(y_1 - \mu_1)(y_1 - \mu_1)^T] & E[(y_1 - \mu_1)(y_2 - \mu_2)^T] \\ E[(y_2 - \mu_2)(y_1 - \mu_1)^T] & E[(y_2 - \mu_2)(y_2 - \mu_2)^t] \end{bmatrix} \ . \tag{5.46}$$

The only expected value that is nontrivial in Eq. 5.46 is the variance of y_1. Algebraic manipulation of Eq. 5.46 results in

$$C(\mathbf{y}) = \begin{bmatrix} \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} & 0 \\ 0 & \Sigma_{22} \end{bmatrix} \ . \tag{5.47}$$

Therefore, with the covariance matrix $C(\mathbf{y})$ and the expected value $E(\mathbf{y})$ it is possible now to write the multivariate Gaussian probability distribution for $\mathbf{y}$ as

$$f(y_1, y_2) = G(y_1|\mu_1 + B\mu_2, \Sigma_{11} + B\Sigma_{21})G(y_2|\mu_2, \Sigma_{22}) \tag{5.48}$$

where y_1 and y_2 are defined in Eq. 5.36. Notice that $f(y_1, y_2)$ is factorized because y_1 and y_2 were defined as orthogonal or uncorrelated. Hence in the wide sense it can be said that both signals are independent.

At the beginning of the model derivation we wanted the probability distribution $f(x_1|x_2)$ and not $f(y_1, y_2)$. Nevertheless using the linear mixing in equations in Eq. 5.42 and the *transformation theorem* it is possible to map the probability distribution from vector $\mathbf{y}$ to vector $\mathbf{x}$. The transformation theorem is defined as

$$g(x_1, ...x_p) = f[y_1(x_1, ...x_p), ..., y_p(x_1, ..., x_p)]J(y_1, ..., y_p) \ , \tag{5.49}$$

where $J(x_1, ..., x_p)$ is the Jacobian. For this derivation the Jacobian matrix is

$$J(y_1, y_2) = \begin{vmatrix} \frac{dy_1}{dx_1} = I & \frac{dy_1}{dx_2} = B \\ \frac{dy_2}{dx_1} = 0 & \frac{dy_2}{dx_2} = I \end{vmatrix} = 1 \ . \tag{5.50}$$

Now, by defining

$$\Sigma_{11\cdot 2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}, \tag{5.51}$$

and applying the transformation theorem, we have that

$$\begin{aligned} f(x_1, x_2) = & \frac{1}{(2\pi)^{q/2}|\Sigma_{11\cdot 2}|^{1/2}} \times \\ & \exp\left\{-\frac{1}{2}(x_1 + Bx_2 - \mu_1 - B\mu_2)^T\Sigma_{11\cdot 2}^{-1}(x_1 + Bx_2 - \mu_1 - B\mu_2)\right\} \times \\ & \frac{1}{(2\pi)^{(p-q)/2}|\Sigma_{22}|^{1/2}} \exp\left\{\frac{1}{2}(x_2 - \mu_2)^T\Sigma_{22}^{-1}(x_2 - \mu_2)\right\}. \end{aligned} \tag{5.52}$$

Notice that $f(x_1, x_2)$ is factorized in two Gaussian distributions. Now recall the definition of conditional probability that says

$$f(x_1|x_2) = \frac{f(x_1, x_2)}{f(x_2)} \ . \tag{5.53}$$

Hence, using the definition of Eq. 5.53 in Eq. 5.52 leads to desired result,

$$f(x_1|x_2) = G(x_1|\mu_1 - B(x_2 - \mu_2), \Sigma_{11\cdot 2}) \ . \tag{5.54}$$

Eq. 5.54 can be easily extended to vector notation in order to work with multiple variables (Zou and Feng , 2009). For instance $f(x_1|x_2, ...x_v)$. Using a more general notation, Eq. 5.54 can be written as

$$f(\mathbf{x_a}|\mathbf{x_b}) = G(\mathbf{x_a}|\mu_{\mathbf{x_a}|\mathbf{x_b}}, \Sigma_{\mathbf{aa\cdot b}}) \ , \tag{5.55}$$

where $\mathbf{x} = [\mathbf{x_a}\ \mathbf{x_b}]^T$ and $\mu_{\mathbf{x_a}|\mathbf{x_b}} = \mu_\mathbf{a} + \Sigma_\mathbf{ab}\Sigma_\mathbf{bb}^{-1}(\mathbf{x_b} - \mu_\mathbf{b})$. $\mathbf{x_a}$ is the vector of children while $\mathbf{x_b}$ is the vector of parent nodes. Notice that in Bayesian networks, the children vector $\mathbf{x_a}$ will always be a univariate time series, hence x_a is a scalar child.

Eq. 5.55 can be used directly for the BIC in Eq 5.33 using the conditional independence property of Bayesian networks.

5.6 Fourier Bayesian networks

As mentioned previously, DBNs are a great technique for network structure and causality inference. This is performed by unfolding the original network and using standard network structure searching algorithms. Nevertheless, in Electrophysiology, which is the case of MEG and EEG, it is not possible to define a static delay as the only causal relation between nodes and unfolding the network several slices will also increase the complexity of the original problem by increasing the size of the network.

This thesis proposes a novel approach for network structure and causality inference for Electrophysiological studies that is named Fourier Bayesian networks or FBNs. FBNs are able to infer causality without network unfolding and delays by using the complex valued information embedded in the time series as the input of the BN algorithm. This is done by extracting the frequency power and the frequency phase using the Fourier transform, hence the name of FBNs. In the following the mathematical derivation for FBNs will be explained and in Chapter 6 properties of FBNs will be studied.

Recall Bayes' theorem for the probability of a hypothetical or proposed network given the stored datasets,

$$p(S_h|\mathbf{D}) = \frac{p(\mathbf{D}|S_h)p(S_h)}{p(\mathbf{D})}\ , \tag{5.56}$$

where

$$p(\mathbf{D}) = \sum_S p(\mathbf{D}|S_h)p(S_h)\ . \tag{5.57}$$

$p(\mathbf{D})$ in Eq 5.57 is the probability of the time series stored in $\mathbf{D}$ and can be obtained by summing out all probable network structures S_h as shown in Eq 5.57. $p(\mathbf{D})$ mainly works as a normalizing factor for $p(S_h|\mathbf{D})$ which is needed in Eq 5.56 to have a total probability of 1 (Heckerman , 1996). Nevertheless, since the universe of networks in S_h can be extremely large depending on the number of nodes in the system, the computation of $p(\mathbf{D})$ is intractable or simply not possible. However, there are in the literature Markov chain Monte Carlo (MCMC) algorithms that are able to sample and evaluate networks from Eq. 5.56 without the need of computing Eq. 5.57 (Walsh , 2004).

Now, if the Fourier transform of the data matrix $\mathbf{D}$ is computed, the time series can be decomposed in frequency power $\mathbf{F}$ and frequency phase $\mathbf{\Theta}$ matrices. These two matrices represent the same data and information stored in $\mathbf{D}$ but in the Fourier domain. Hence, the posterior probability in Eq. 5.56 can also be represented as

$$p(S_h|\mathbf{D}) \Longrightarrow p(S_h|\mathbf{F},\mathbf{\Theta}) \ . \tag{5.58}$$

Substituting the previous equation in the Bayes' theorem definition leads to

$$p(S_h|\mathbf{F},\mathbf{\Theta}) = \frac{p(\mathbf{F},\mathbf{\Theta}|S_h)p(S_h)}{p(\mathbf{F},\mathbf{\Theta})} \ . \tag{5.59}$$

At this point is worth to recall that in a Fourier transformed signal, frequency power and frequency phase are independent, meaning that $p(\mathbf{F},\mathbf{\Theta}) = p(\mathbf{F})p(\mathbf{\Theta})$. This independence property allows to factorize the joint probability of $\mathbf{F}$ and $\mathbf{\Theta}$ in Eq. 5.59 as

$$p(S_h|\mathbf{F},\mathbf{\Theta}) = \frac{p(\mathbf{F}|S_h)p(\mathbf{\Theta}|S_h)p(S_h)}{P(\mathbf{F})p(\mathbf{\Theta})} \ . \tag{5.60}$$

Here, $p(\mathbf{F}|S_h)$ is the likelihood of the spectral matrix $\mathbf{F}$ given the network structure, $p(\mathbf{\Theta}|S_h)$ is the likelihood of the phase matrix $\mathbf{\Theta}$ given the structure, and $p(S_h)$ is the prior probability of the proposed network structure.

Eq. 5.60 can be used to sample networks from the universe of possible S_h solutions using Monte Carlo methods for Bayesian inference (Chen et al. , 2000; Walsh , 2004). In order to achieve this, it is necessary to derive the mathematical representation of $p(\mathbf{F}|S_h)$ and $p(\mathbf{\Theta}|S_h)$ which are explained in the next sections.

5.6.1 Spectral likelihood

In this Section the spectral likelihood $p(\mathbf{F}|S_h)$ is derived. It can be started by adopting the regression model widely used in Gaussian networks (Schachter and Kenley , 1989; Geiger and Heckerman , 1994; Peraza and Halliday , 2010b), which states that the likelihood distribution $p(\mathbf{D}|S_h)$ is represented as the error of a linear regression of Gaussian processes. Using the Gaussian network model, allows to define the likelihood as

$$p(\mathbf{D}|S_h) = \prod_{n=1}^{N} \frac{(2\pi)^{-1/2}}{|\Sigma_{xx|\mathbf{pa}_n}|^{1/2}} \times \exp\left\{ -\frac{1}{2}(x_n - \mu_{x|\mathbf{pa}_n})\Sigma^{-1}_{xx|\mathbf{pa}_n}(x_n - \mu_{x|\mathbf{pa}_n}) \right\} , \quad (5.61)$$

where $\mathbf{pa}_n$ is the vector of parent nodes of x_n,

$$\mu_{x|\mathbf{pa}_n} = \mu_x - \Sigma_{x\mathbf{pa}_n}\Sigma^{-1}_{\mathbf{pa}_n}(\mathbf{pa}^T_n - \mu_{\mathbf{pa}_n}) , \quad (5.62)$$

$$\Sigma_{xx|\mathbf{pa}_n} = \Sigma_x - \Sigma_{x\mathbf{pa}_n}\Sigma^{-1}_{\mathbf{pa}_n}\Sigma_{\mathbf{pa}_n x} , \quad (5.63)$$

$$\Sigma = \begin{bmatrix} \Sigma_x & \Sigma_{x\mathbf{pa}_n} \\ \Sigma_{\mathbf{pa}_n x} & \Sigma_{\mathbf{pa}_n} \end{bmatrix} , \quad (5.64)$$

μ_x and $\mu_{\mathbf{pa}_n}$ are the means of x_n and $\mathbf{pa}_n$ respectively, and Σ is the covariance matrix of the vector $[x_n\ \mathbf{pa}_n]$.

Notice that x_n in Eq. 5.61 is a univariate vector, hence the likelihood equation can be rewritten using the sum of elements in x_n,

$$p(\mathbf{D}|S_h) = \prod_{n=1}^{N} \frac{(2\pi)^{-1/2}}{|\Sigma_{xx|\mathbf{pa}_n}|^{1/2}} \exp\left\{ -\frac{1}{2}\frac{\sum_{i=1}^{M}(x_n(i) - \mu_{x|\mathbf{pa}_n})^2}{\Sigma_{xx|\mathbf{pa}_n}} \right\}, \quad (5.65)$$

where i is the sample index for the time series x_n. Taking the logarithm of Eq. 5.65 we have that

$$\ln p(\mathbf{D}|S_h) = \sum_{n=1}^{N} \left(-\frac{1}{2}\ln(2\pi\Sigma_{xx|\mathbf{pa}_n}) - \frac{M}{2}\frac{\hat{\Sigma}_{xx|\mathbf{pa}_n}}{\Sigma_{xx|\mathbf{pa}_n}} \right) , \quad (5.66)$$

where $\hat{\Sigma}_{xx|\mathbf{pa}_n}$ is the variance estimator of the regression error ϵ defined by

$$var[\epsilon] = var[x_n - \mu_x + \Sigma_{x\mathbf{pa}_n}\Sigma^{-1}_{\mathbf{pa}_n}(\mathbf{pa}_n - \mu_{\mathbf{pa}_n})] \tag{5.67}$$

$$= \Sigma_{xx|\mathbf{pa}_n} \tag{5.68}$$

Assuming that x_n is long enough and stationary, Eq. 5.66 can be represented as

$$\ln p(\mathbf{D}|S_h) = -\frac{1}{2}\sum_{n=1}^{N}\ln(\Sigma_{xx|\mathbf{pa}_n}) - \frac{N}{2}\ln(2\pi) - \frac{MN}{2} \tag{5.69}$$

Because of the Gaussian and stationary assumption on ϵ, the variance of this error can be rewritten as

$$\Sigma_{xx|\mathbf{pa}_n} = \int_\lambda f^n_{\epsilon\epsilon}(\lambda)d\lambda \ . \tag{5.70}$$

In the previous equation the Parseval's theorem is used to represent the variance of the regression error as the total energy of its spectrum. Hence, $f^n_{\epsilon\epsilon}$ is the error spectrum, the spectrum of the regression of vector $\mathbf{pa}_n$ on x_n. $f^n_{\epsilon\epsilon}$ can be obtained directly using spectral regression (Priestley , 1983) as follows,

$$f^n_{\epsilon\epsilon}(\lambda) = f_{xx}(\lambda) - \mathbf{A}f^T_{\mathbf{pa}_n x}(\lambda) \ , \tag{5.71}$$

where

$$\mathbf{A} = f_{x\mathbf{pa}_n}(\lambda)f^{-1}_{\mathbf{pa}_n}(\lambda) \ , \tag{5.72}$$

$f_{xx}(\lambda)$ is the second order spectra of x_n, $f_{x\mathbf{pa}_n}(\lambda)$ represents the cross spectrum of x_n with the vector of parents $\mathbf{pa}_n$, $\mathbf{A}$ is the complex regression coefficient matrix (Priestley , 1983), $f_{\mathbf{pa}_n}$ is the cross spectral matrix of $\mathbf{pa}_n$, and λ is the frequency index. Also notice that $f^n_{\epsilon\epsilon}(\lambda)$ is a scalar positive number whose magnitude is independent of the time series' phases. This fact is of paramount importance in this step of the derivation, since it indicates that the error spectrum is invariant to the phase of all signals.

Finally, using the error spectrum definition, the data likelihood can be represented as

$$\ln p(\mathbf{F}|S_h) = -\frac{1}{2}\sum_{n=1}^{N}\ln\int_\lambda f^n_{\epsilon\epsilon}(\lambda) - \frac{N}{2}\ln(2\pi) - \frac{MN}{2} \ . \tag{5.73}$$

Because it is wanted to optimize is the likelihood function only, all constant terms can be eliminated from the previous expression leading to

$$\ln p(\mathbf{F}|S_h) \equiv -\frac{1}{2}\sum_{n=1}^{N}\ln\int_\lambda f^n_{\epsilon\epsilon}(\lambda) \tag{5.74}$$

The invariance property of the error spectrum to the phase of the time series will cause that the likelihood $p(\mathbf{F}|S_h)$ gives a high probability to those networks that fit the system's network structure but ignoring causality completely. This steps works in some way as a "phase normalization" of all the recorded time series, and focus the interest on how the proposed network S_h is able to explain the signal power manifested by the system. Furthermore, this step also recalls heuristic approaches in DBNs where the envelope of the time series is used as input of the BNs algorithm, see for instance Smith et al. (2006). Causality however is inferred by the phase likelihood $p(\mathbf{\Theta}|S_h)$.

5.6.2 Phase likelihood

To propose a likelihood function for the phase, a heuristic approach is followed. For the convergent model network having a single child and several parents, define the vector of phases $\mathbf{\Theta}_n$ as

$$\mathbf{\Theta}_n = \theta_{x\mathbf{pa}_n} = [\theta_{xpa(1)}\ \theta_{xpa(2)}\ ...\ \theta_{xpa(p)}]\ , \tag{5.75}$$

where $\theta_{xpa(1)}$ represents the estimated cross phase between the first parent in vector $\mathbf{pa}_n$ and the child x_n. $\mathbf{\Theta}_n$ has then all the cross phase estimations between the child x_n and its parents in $\mathbf{pa}_n$. Hence, the probability of this convergent network is

$$p(\mathbf{\Theta}_n|S_n) = p(\theta_{xpa(1)},\ \theta_{xpa(2)}, \cdots, \theta_{xpa(p)}|S_n)\ , \tag{5.76}$$

where S_n stands for the network composed by x_n and its p parents in $\mathbf{pa}_n$.

Now assume that the phase differences between a child x_n with any of the parents in $\mathbf{pa}_n$, $\theta_{x\mathbf{pa}_n}$, are independent. Hence, the total phase probability can be represented as a product of phase probabilities

$$p(\Theta_n|S_n) = p(\theta_{x\mathbf{pa}_n}|S_n) = \prod_{i=1}^{p} p(\theta_{xpa_n(i)}) \tag{5.77}$$

Notice that Eq. 5.77 is not strictly true, since the common child node x_n and its phase adds a level of dependency on the phase difference of all the arriving edges from the parent nodes. This is because x_n works as a common reference for all parents. A well explained case of this issue is discussed by Guevara et al. (2005).

In order to obtain a measure for the entire network S_h, it is needed to include the rest of the nodes and their parents into Eq. 5.77. This gives the final expression for the network phase probability as

$$p(\mathbf{\Theta}|S_h) = \prod_{n=1}^{N} p(\theta_{x_n \mathbf{pa}_x}|S_x) = \prod_{n=1}^{N}\prod_{i=1}^{p} p(\theta_{x_n pa(i)}) \ . \tag{5.78}$$

Taking the logarithm of the previous equation leads to

$$\ln p(\mathbf{\Theta}|S_h) = \sum_{n=1}^{N}\sum_{i=1}^{p} \ln p(\theta_{x_n pa(i)}) \ . \tag{5.79}$$

Eq. 5.79 can be seen as a score function that evaluates the phase of the entire network. This means that for every proposed causal structure S_h, $p(\mathbf{\Theta}|S_h)$ will search for the one that fits the cross phase difference among the datasets, meaning that the phase likelihood is focused on finding causality and not network structure.

Notice that the spectral nature of FBNs allows to study the network probabilty at specific or desired frequency bands, being these defined by λ in Eq. 5.74 and the phase can be estimated also for the same λ band of frequencies. This property of FBNs is studied in Chapter 6 through simulations.

5.6.3 Phase estimation

When working with real electrophysiological recordings it is not easy to estimate the cross-phase difference between two biological signals. This is mainly because as all things in nature nothing is linear and stationary. Biological systems are complex ever changing systems which provide time series that are rich in information, but also difficult to work with. Nevertheless, in this thesis stationarity and linearity are assumed as properties of the analysed system. This allows to estimate the cross-phase between two sources using a modified method originally published in Rosemberg et al. (1989) which uses the slope of the phase spectrum.

The phase spectrum $\theta(\lambda)$ between two signals x and y can be obtained from the argument of coherency (Rosemberg et al. , 1989) defined as

$$R_{yx}(\lambda) = \frac{f_{yx}(\lambda)}{\sqrt{f_{yy}(\lambda)f_{xx}(\lambda)}}. \tag{5.80}$$

Hence $\theta_{yx}(\lambda) = arg\{R_{yx}(\lambda)\} = arg\{f_{yx}(\lambda)\}$. Assume now that the time series y is a time lagged version of x, then

$$f_{yx}(\lambda) = e^{-i\lambda\tau} f_{xx}(\lambda) \ . \tag{5.81}$$

where the phase spectrum $\theta_{yx} = -\lambda\tau$ and the slope τ defines the time delay between y and x. Nevertheless as mentioned at the beginning of this section, a time series acquired from a real biological system does not produce a constant slope in its phase spectrum and can not be fitted using a straight line. For this reason a weighted slope approach is used for the FBNs implemented in this thesis by computing first the cross-phase at every frequency using the derivative of the phase spectrum. This is implemented as

$$\tau_{yx}(\lambda) = \frac{\partial\theta_{yx}(\lambda)}{\partial\lambda} \ . \tag{5.82}$$

Notice also that τ_{yx} can have a range from $-\infty$ to ∞. Using the phase spectrum $\theta_{yx}(\lambda)$ the delay between two signals is estimated as

$$\hat{\tau}_{yx}(\lambda) = \frac{\Delta\theta_{yx}(\lambda)}{\Delta\lambda} = \frac{\theta(\lambda+\Delta\lambda) - \theta(\lambda-\Delta\lambda)}{2\Delta\lambda} \ . \tag{5.83}$$

A negative gradient in the previous equation will suggest that the chosen model is correct, meaning x is leading y or x is cause of y, $x \to y$.

At this point, we have to take into account that the estimated delay is only important where coherence R_{yx} is high. A way to do this is by weighting the estimated delay using squared coherence $R^2_{yx}(\lambda)$ as weighting function. $R^2_{yx}(\lambda)$ falls faster with low coherence values, providing a better suppression of those τ estimations at frequencies where coherence is low. The final expression for weighted delay τ^w_{yx} is

$$\tau^w_{yx} \equiv \frac{1}{\Delta\lambda}\int_{\Delta\lambda} \tau_{yx}(\lambda) R^2_{yx}(\lambda)\partial\lambda \tag{5.84}$$

Finally, in order to apply this delay estimation in the FBNs algorithm, it is necessary to express the delay as a probability. This can be implemented by mapping $\tau_{yx}(\lambda)$ onto the inverse logistic function L,

$$p(\theta_{yx}) \equiv L(\tau_{yx}) = \frac{1}{1+\exp(2\beta\tau_{yx})} \ . \tag{5.85}$$

Although Eq. 5.85 is not a phase estimation since a delay is used instead, it provides a numerical score to assess the idea of Fourier Bayesian networks using a spectral likelihood and a phase likelihood as a novel network structure searching approach and also test their properties.

5.7 Fourier BIC

Using the concept of spectral likelihood it is possible to create a Fourier version of the original BIC score function. This idea was originally published in Peraza and Halliday (2010b) as the FBIC score function. FBIC uses the spectral likelihood directly in the BIC function. FBIC is defined as

$$FBIC(S_h|D) = \sum_{y=1}^{N} \left(\ln \int_{\lambda} f_{\epsilon\epsilon_y}(\lambda) d\lambda - alpha * k \ln(M) \right) , \tag{5.86}$$

where the tuning parameter *alpha* has been added to the punishment term as in BIC.

FBIC behaves similarly to the standard BIC criterion for the case of static BNs, with the advantage of the phase normalization given by the frequency power. As shown in Peraza and Halliday (2010b), this score is good in finding network structure and its advantage relies that the phase information can be used to test if the direction of the found edge is correct. Also as in the case of FBNs, the FBIC can be used to find the best network in a specific band of frequencies defined by the λ variable.

The design of network probability distributions or score functions, which are another representation of these probabilities, needs to be implemented in algorithms that search through the network distribution for all those networks that fit the data. There are in the literature an large number of algorithms created to optimize network structure searching. In the next section some of these algorithms are studied.

5.8 Network structure searching

In this section some algorithms for network structure searching are reviewed. The previous section covered some network score metrics which are used to evaluate a

proposed or hypothetical network S_h which is chosen from the pool of all possible networks available. Nevertheless, this universe of hypothetical networks can be extremely large and it is computationally expensive to evaluate all of them in order to find the network or family of networks that fits the dataset. For this reason, the research community has developed strategies to sample networks from the universe S_h and keep the most significant ones. From all the approaches, Markov chain Monte Carlo or MCMC family of algorithms is the most successful and applied one. This section starts explaining the most efficient and basic of the structure searching algorithms; the K2 algorithm.

5.8.1 K2 Algorithm

The K2 algorithm (Cooper and Herskovits , 1992) is a simple method to find network structure which derives from another one called "Katutó" (Hersovits and Cooper , 1991) hence the name. The K2 algorithm is considered a greedy searching algorithm since it evaluates different networks and chooses the one with the highest score. Its only disadvantage is that the node order hierarchy (from parents to children) must be provided in advance. In some real applications this is possible, a physician might say that flu causes cough and a hierarchy can be settled, but in many other applications this is not an option. However, the K2 algorithm is the first approach to solve any problem and it is also used as a measure point to compare the performance of new algorithms. See for instance Chen et al. (2008) where a variant of the K2 algorithm is proposed which uses a node ordering algorithm before running the K2 algorithm.

The pseudocode for the K2 algorithm can be found in Neapolitan (2004), which is reproduced here;

```
Network structure <- function (data D, node order V-->1)
{
 for (i=1;i<=n; i++)
 {
   PAi=0 ;
   Pold=score(D,Xi,PAi);
```

```
findmore=true;
// u= max of parents allowed per child node.
while(findmore && |PAi|< u)
  {
  Z=new node that maximizes score(D,Xi,PAi+Z);
  Pnew=score(D,Xi,PAi+Z) ;
  if(Pnew > Pold) {
    Pold=Pnew ;
    PAi=PAi+Z;  }
  else
    findmore=false;
  } } }
```

In the previous pseudocode `u` is the maximum number of parents allowed per child node, `PAi` is the vector of parents which can grow, and `score()` is the score function, for instance BIC. The K2 algorithm is fast and very efficient not counting the node ordering disadvantage.

5.8.2 A DBN algorithm using time-lagged partial correlation

This algorithm was published in Peraza and Halliday (2010a) and has four steps or rules that are executed in a consecutive fashion. The first step finds all the inter-slice edges of the network by applying the time-lagged partial correlation matrix reviewed in Sec 4.1.1. Then, the second step analyses all the convergent three-node structures and uses a score function to evaluate them. The third rule evaluates cycles and finally the fourth rule analyses all remaining edges.

The advantage of this algorithm compared with K2 is that a network node pre-ordering is not needed and also it is fast finding the network structure. However, it is commonly said in the literature that these kind of algorithms might also fall in a local minimum (or maximum), meaning that sometimes they will not be able to find the true network that describes the analysed system. These are trade offs that

researchers must take into account when using this or any other kind of algorithm.

In the following the four rules of this algorithm are studied.

Rule 1: Time-lagged partial correlation matrix

The time-lagged partial correlation concept was explained in Section 4.1.1. The goal of this rule is to obtain the dynamic connection matrix **DM** which will be used in further rules. **DM** is found by applying a threshold to the partial correlation matrix defined in Eq. 4.11. This threshold will decide if a statistically significant edge exists or not between two nodes a and b. This is found using Fisher's transformation explained previously in Section 4.1 with a desired level of significance. For instance in Peraza and Halliday (2010a) a significance level of 0.05 was used as edge threshold.

After thresholding Eq. 4.11, **DM** will be of the form

$$\mathbf{DM} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{B} & \mathbf{A} \end{bmatrix}, \tag{5.87}$$

where $\mathbf{A} = \lfloor \mathbf{A}_{t,t} \rfloor$ and $\mathbf{B} = \lfloor \mathbf{B}_{t-1,t} \rfloor$ are thresholded versions of the submatrices in Eq. 4.11 and whose elements above the threshold are replaced with 1s and 0s otherwise.

As explained before, a network can be unfolded several time slices. However each slicing will increase the system complexity and also the number computations performed. This means that if the Bayesian network is unfolded k slices, **DM** will have submatrices until $\mathbf{A}_{t-k,t-k}$. However, here the network is unfolded up to $t-1$ as it was defined in Eq. 4.11.

In Rule 1 it is important to define all the dynamic edges that will be allocated in the submatrix **B** in **DM**. The steps for this are:

1. Create a backward shifted version of the data matrix $\mathbf{D}_t$ and call it $\mathbf{D}_{t-1}$. Then concatenate both data matrices to create a new $\mathbf{D} = [\mathbf{D}_t \ \mathbf{D}_{t-1}]$.

2. Compute the correlation matrix for the new $\mathbf{D}$, $Corr(\mathbf{D})$.

3. Invert $Corr(\mathbf{D})$ and then normalize the resultant matrix to obtain ones in its diagonal, the result will be the time-lagged partial correlation matrix $PCorr$.

4. Use a threshold to define the zeros and ones of the connexion matrix.

5. Build the dynamic connexion matrix as defined in Eq. 5.87.

6. Using the submatrix **B**, define all the edges pointing from $t-1$ to t, see for instance Figure 4.3. Place these edges in a list called *fixed-edges*.

7. Using submatrices $\mathbf{A}_t$ and $\mathbf{A}_{t-1}$ define all the variable edges that still do not have a direction and place them in a list called *var-edges*.

Hence, the output of this stage besides the matrix **DM** are the vectors *fixed-edges* and *var-edges*.

Rule 2: Convergent structure analysis

Rule 2 looks for convergent structures. Because convergent structures are easily inferred using the available data, it is possible to find if an undirected structure is a convergent structure or not. This stage is explained in the following steps:

1. Identify three node paths of the shape $a_{t-1} \rightarrow b_t - c_t$, see Fig. 4.3 for a reference. Note that the edges pointing from a_{t-1} to b_t were fixed in Section 5.8.2.

2. Evaluate and score the two possible combination for the paths found in step 1, they can be either $a_{t-1} \rightarrow b_t \rightarrow c_t$ or $a_{t-1} \rightarrow b_t \leftarrow c_t$.

3. Choose the option with the highest score and fix its edges. Place the fixed edges in the *fixededges* list and delete them from *varedges* including the delayed mirror edges $b_{t-1} - c_{t-1}$. Also, update the dynamic matrix **DM**.

4. Identify three node paths of the shape $a \rightarrow b - c$ regardless of the lag index.

5. Do the same as in steps 2 and 3.

6. Identify three node paths of the shape $a - b - c$ regardless of the lag index, and evaluate two possibilities; convergent structure and any other.

7. If the convergent network evaluated in step 6 results in the highest score: Place the fixed edges in the *fixededges* list and delete them from *varedges* including

the delayed mirror edges $b_{t-1} - c_{t-1}$. Also, update the connection matrix **DM**. Otherwise, leave the edges as undirected.

The next rule also looks for network patterns, but now it is more difficult because it involves one node more and possible cycles must be avoided.

Rule 3: 4-node acyclic paths

This stage searches all the acyclic four-node loops inside the total unfolded network in $\mathbf{A}_t$, $\mathbf{A}_{t-1}$ and even using the fixed edges in **B** as bridges to build four-node loops. We can summarize this stage with the following steps.

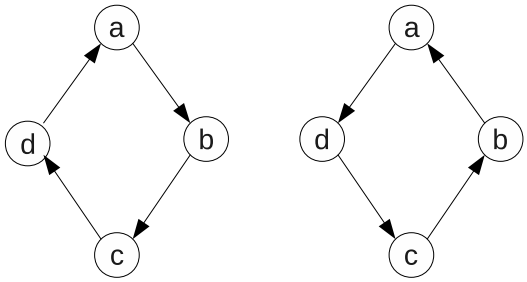

Figure 5.10: Forbidden four-node cycles in BNs.

1. Identify 4-node loops in the network.

2. Using the lists *fixededges* and *varedges* identify the directed and undirected edges of the loops found in the previous step.

3. For each loop iterate all possible edges that form a full directed loop avoiding cycles as the ones shown in Fig. 5.10.

4. If from all the possible networks found in step 3, there is only one with the highest score, fix it and place the new directed edges in *fixededges* and delete them from *vardedges* including their mirror edges at slice t or $t-1$. Otherwise, leave the undirected edges as found previously.

5. Repeat steps 3 and 4 for all the 4-node loops in the network.

Rule 4: Scoring remaining edges

This is the simplest of all the algorithm's rules. It consists of iterating all the network combinations using the remaining variable edges if any and scoring the totality of the network using the probability model.

The network searching as it is described here is the worst of all searching techniques, in fact few authors would call it a technique. However, because very few non-fixed edges are expected at this point, the computation of this stage is fast. Furthermore, the algorithm does not look for the skeleton of the network, the undirected network was established in Rule 1. Here, only the direction of the remaining edges is searched and this is more efficient. The steps for this stage are as follows:

1. Identify all the remaining variable edges in the unfolded BN.

2. Iterate all the possible combinations (avoiding cycles) and score each iteration.

3. If only one network with the highest score is found, fix all the remaining edges and place them in the vector *fixed-edges*.

4. If more than one network is found with the highest score, group them as the most probable solution.

5. End of the algorithm.

This DBN rule based algorithm was tested in Peraza and Halliday (2010a) for a simulated MVAR system and also on real time series from the Saccharomyces Cerevisiae (yeast) cell cycle datasets using BIC as score function. The next section covers a very successful area in network structure searching where networks are sampled directly from the posterior probability distribution.

5.9 Monte Carlo methods

Network structure inference using Bayesian networks can be done by sampling networks from the posterior probability distribution

$$p(S_h|D) = \frac{p(D|S_h)p(S_h)}{p(D)} \ , \tag{5.88}$$

and averaging the sampled network to obtain a representative mean of the network system. Also the population mode can be computed and taken as the estimated network. The problem here is that almost always it is impossible to sample networks from Eq. 5.88 since the denominator (the normalizing factor) is not known and the prior distribution $p(S_h)$ is most of the time also unknown and has to be proposed based on previous experiences.

However, network sampling from the posterior can be done in an indirect way using Monte Carlo methods.

5.9.1 Markov chain Monte Carlo

A Markov chain is a system where a future sample or output depends only on its predecessor. This kind of systems are also called in Signal Processing as Markov processes (León-Garcia , 1994). Markov chain Monte Carlo is a sampling method that behaves as a Markov process and it is based on a Monte Carlo method for computation of integrals. As explained in Walsh (2004), assume that it is desired to compute the following integral,

$$\int_a^b h(x)dx \tag{5.89}$$

and that the function $h(x)$ can be factorized in two functions $f(x)$ and $p(x)$, the latter being a probability distribution. This can be expressed as

$$\int_a^b h(x)dx = \int_a^b f(x)p(x)dx \ . \tag{5.90}$$

Since $p(x)$ is a probability distribution the previous expression becomes an expected value with respect to $p(x)$. This means that

$$\int_a^b f(x)p(x)dx = E_{p(x)}\left[f(x)\right] \approx \frac{1}{M}\sum_{i=1}^{M} f(x_i) \ , \tag{5.91}$$

where M is a very large number of samples x_i which were drawn from $p(x)$. Now imagine that $h(x)$ can not be factorized, in this case there is a slight change

applied by proposing another known probability distribution $q(x)$ and adding it to the integral as follows

$$\int_a^b f(x)\frac{p(x)}{q(x)}q(x)dx = E_{q(x)}\left[f(x)\left(\frac{p(x)}{q(x)}\right)\right] \approx \frac{1}{M}\sum_{i=1}^{M} f(x_i)\left(\frac{p(x_i)}{q(x_i)}\right) \ , \qquad (5.92)$$

The previous equation is called in the literature importance sampling (Gamerman and Lopes , 2006). Notice that although $h(x)$ is shown factorized it is not strictly required.

Now the concept of Markov chains needs to be explained more deeply. Markov chains are very well explained in Walsh (2004), but since this concept is needed to understand the Metropolis-Hastings algorithm in Section 5.9.2, it is briefly explained here.

As previously mentioned a Markov chain is a system where future outcomes only depend on its last predecessor. This can be represented using conditional probabilities as

$$p(X_{t+1} = s_k | X_t = s_g, X_{t-1} = s_h, ..., X_{t-2} = s_j,) = p(X_{t+1} = s_k | X_t = s_g) \ , \qquad (5.93)$$

where the outputs s are different states of the Markov chain. What Eq. 5.93 says is "the probability of jumping to state s_k if currently the system is in state s_g". This probability depends only on s_g and the information or knowledge of previous steps is lost. Now represent the probability of the current state as

$$\pi_j(t+1) = p(X_{t+1} = s_j) \qquad (5.94)$$

which can also be computed with

$$\begin{aligned} \pi_j(t+1) &= \sum_k p(X_{t+1} = s_j | X_t = s_k) p(X_t = s_k) \\ &= \sum_k p(X_{t+1} = s_j | X_t = s_k) \pi_k(t) \\ &= \sum_k p(k \to j) \pi_k(t) \end{aligned} \qquad (5.95)$$

which in matrix representation for the Markov chain state can be written as

$$\mathbf{\Pi}(t+1) = \mathbf{\Pi}(t)\mathbf{P} \ , \qquad (5.96)$$

where $\mathbf{P}$ is the transition matrix. This is a square matrix whose elements indicate the probability of going from column state a to row state b. Hence all rows in this matrix sum to 1. An interesting property of Markov chains is their stationarity. As can be seen, Eq. 5.96 is an iterable equation where $\mathbf{\Pi}$ is not known in advance but it can be proposed at the beginning of the iteration and after some "burning time" it will reach stationarity or stability, which implies that

$$\mathbf{\Pi}^* = \mathbf{\Pi}^* \mathbf{P} \ , \tag{5.97}$$

where $\mathbf{\Pi}^*$ shows the probability of being in any state at any moment of time.

5.9.2 Metropolis algorithm

The Metropolis algorithm is a sampling method proposed by Metropolis using symmetrical distributions (Metropolis and Ulam , 1949; Metropolis et al. , 1953). Symmetrical distribution refers that the probability of going from step a to step b is the same as going from b to a in the Markov chain. This is represented as $p(a \rightarrow b) = p(b \rightarrow a)$. The easiest choice although not the more efficient is to choose the uniform distribution where all transition probabilities are equal.

For the case of Bayesian networks which are the interest of this thesis the Metropolis algorithm is as follows:

1. Propose an initial hypothetical network S_h ,
2. Using the previous network, modify it applying one the three basic network changes (see Section 5.9.4) to create a new S_h^+ ,
3. evaluate the posterior probability with both networks using

$$\alpha = \frac{p(S_h^+|\mathbf{D})}{p(S_h|\mathbf{D})} \ , \tag{5.98}$$

4. If $\alpha < 1$, accept the proposed network S_h^+ with probability α, otherwise accept S_h^+ ,
5. Go to step 2 and repeat until stationarity is reached ,

6. All accepted networks will be networks sampled from $p(S_h|D)$.

The Metropolis algorithm can be summarized as

$$\alpha = \min\left(1, \frac{p(S_h^+|\mathbf{D})}{p(S_h|\mathbf{D})}\right) \ . \tag{5.99}$$

The Metropolis algorithm allows to sample networks from the posterior distribution. Notice that with this algorithm it is not needed to compute the normalizing factor which is the probability of the datasets $p(\mathbf{D})$ because this factor is cancelled. From the network samples, it is possible to compute for instance the sample mode and use it as the network estimator. However, there are modifications of the original Metropolis algorithm that are able to reach the maxima of the network probability distribution and sample networks around this point, such as simulated annealing. This is the optimization method applied in this thesis for network inference.

5.9.3 Simulated annealing

Simulated annealing (Kirkpatrick et al. , 1983; Walsh , 2004) is implemented in the same way than Metropolis algorithm, but the acceptance probability is defined by

$$\beta = \min\left[1, \ \left(\frac{p(S_h^+|\mathbf{D}}{p(S_h|\mathbf{D})}\right)^{1/T(t)}\right] \ . \tag{5.100}$$

The function $T(t)$ is called the cooling schedule which cools down the network acceptance probability β in Eq. 5.100 and helps to sample networks closer to the network distribution maxima (Walsh , 2004). The cooling schedule function is defined as

$$T(t) = \max\left[T_0\left(\frac{T_f}{T_0}\right)^{t/b}, T_f\right] \ , \tag{5.101}$$

where T_0 is the initial temperature, T_f is the final temperature and b is the cooling time, which is the time it takes to reach T_f. The proposed network S_h^+ is created by changing the previous network S_h using one of three basic network changes, see Section 5.9.4.

Notice that as in the Metropolis algorithm the denominators in Eq 5.60 are cancelled, leading to

$$\alpha = \frac{p(S_h^+|\mathbf{D})}{p(S_h|\mathbf{D})} = \frac{p(\mathbf{D}|S_h^+)p(S_h^+)}{p(\mathbf{D}|S_h)p(S_h)} \tag{5.102}$$

At this point there are two approaches that can be followed, one is to say that for large networks S_h^+ and S_h are similar or almost equal, differing in only one edge. This means that $p(S_h) \approx p(S_h^+)$ and the previous ratio can be further simplified to

$$\alpha = \frac{p(S_h^+|\mathbf{D})}{p(S_h|\mathbf{D})} = \frac{p(\mathbf{D}|S_h^+)}{p(\mathbf{D}|S_h)} \ . \tag{5.103}$$

This previous step is commonly assumed because in principle the prior distribution $p(S_h)$ is not known. See for instance Rajapakse and Zhou (2007).

The second approach is to apply our previous knowledge of what is expected to find as a network, and this is a sparse network. Hence, it is possible to use a prior distribution for the Bayesian ratio that gives high probabilities to sparsely connected networks and low probabilities to highly connected ones. This approach is followed in this thesis for the FBNs networks by using the FBIC score which introduces a punishment factor for network complexity. The FBNs acceptance ratio using simulated annealing is finally defined by

$$\beta = \min\left[1, \ \left(\frac{p(\mathbf{F}|S_h^+)p(\mathbf{\Theta}|S_h^+)p(S_h^+)}{p(\mathbf{F}|S_h)p(\mathbf{\Theta}|S_h)p(S_h)}\right)^{1/T(t)}\right] \ . \tag{5.104}$$

The first proposed network S_h has to be chosen previously by the researcher, which can be the empty network for instance. The following network proposals S_h^+ must be chosen using one of the three basic network changes; edge addition, edge deletion, and edge reversal.

5.9.4 Proposing a network change

In order to propose new networks for the simulated annealing algorithm, the new network S_h^+ must be created by applying to the current network S_h one of the three basic network changes. These are edge addition, edge deletion, and edge reversal. These basic network edge modifications apply a small change to the current network. Here it can be appreciated why the Metropolis algorithm produces a Markov chain. Every network in the universe of networks is one state in the Markov chain, and when

a new network is proposed what is evaluated is the probability of jumping to the next state. The networks that receive more jumps when the chain reaches stationarity, are the networks that describe the best the studied system. Figure 5.11 shows the three network changes. In this thesis the edge changes applied for Metropolis or simulated annealing algorithm are chosen randomly, each one with probability 1/3.

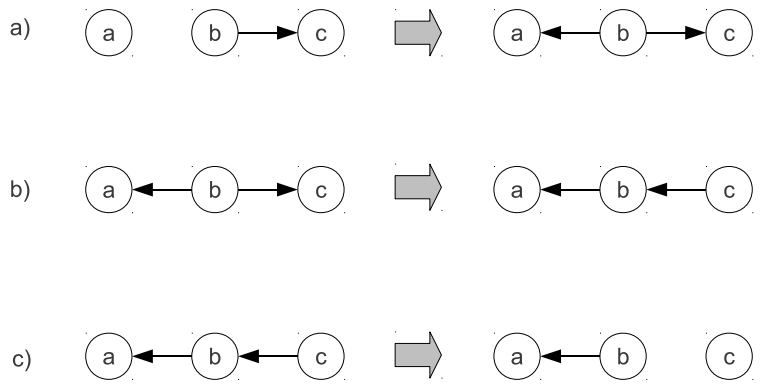

Figure 5.11: Three basic network changes. a) Edge addition, b) Edge reversal, c) Edge deletion.

Chapter summary

In this chapter an introduction to Bayesian networks for structure searching was given. This included basic theory on network evaluation using the posterior distribution from Bayes' theorem. Also, network score metrics were reviewed such as BDe, BIC, and FBIC which can be used in rule-base algorithms or stochastic ones. This chapter also introduced the main contribution of this thesis which are Fourier Bayesian networks or FBNs. FBNs use the complex valued information embedded in the time series to find network structures and edges' causality. In order to do this, FBNs uses the Fourier transform to decompose the time series in two matrices, one for the frequency power $\mathbf{F}$ and another for the frequency phase $\mathbf{\Theta}$. Using these two matrices FBNs are able to factorize the likelihood distribution $p(\mathbf{F}, \mathbf{\Theta}|S_h)$ in Bayes' theorem in two likelihoods, one for the power and other for the phase. FBNs have many good properties against standard BNs or DBNs. FBNs are nonparametric,

this means that FBN does not rely on the network unfolding like DBNs. FBNs do not need to unfold the network and can be focused on a specific band of frequencies λ for the network searching thanks to their Fourier nature. These properties will be studied in the next chapter where FBNs are tested through simulations and compared with BNs and DBNs using Gaussian models.

Chapter 6

Simulations

This chapter explains through simulations the proposed method of Fourier Bayesian networks or FBNs. FBNs performance is tested using two network models, a linear one using MVAR processes and a non-linear model using a variant of the MVAR where the exponential function e is used as non-linear operator. FBNs are based on the Fourier transform whose complex coefficients are used to make inferences about the network power and the network phase likelihood functions.

All simulations in this chapter are implemented in Python (version 2.6.5). The 2-dimensional graphs are created using the Python module Matplotlib (version 1.3.0) and the 3-dimensional ones with the brain cortex meshes are created using Python-Mayavi2 module Enthought.mayavi (version 3.3.0).

6.1 Spectral and phase likelihood

This section shows in action the spectral and phase likelihoods presented in Eq. 5.74 and Eq. 5.79. The spectral likelihood which is based on the linear model can be seen as a spectral regression of Fourier coefficients. From this perspective Eq. 5.74 is not different from any other regression in time domain. However, since the Fourier coefficients are used for the regression and these represent the amplitudes of orthogonal sines and cosines, the cross-phase among the child node and its parents is completely ignored. In other words, for the spectral likelihood in Eq. 5.74 only power fitting is important at every frequency point.

How Eq. 5.74 works can be explained with the following network example. The network shown in Figure 6.1 (Baccala , 2001) is a 5-node network whose behaviour is described by a set of MVAR equations in Eq. 6.1. It is possible to infer this network structure by studying the interactions among the time series recorded from the network nodes (sources). Figure 6.2 shows the coherence matrix for the five sources. Each element of this matrix shows the coherence $R_{row,column}$. As can be seen, there is a strong coherence between neighbouring nodes and weak coherence between nodes that are far away within the network.

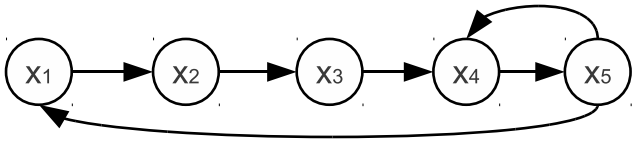

Figure 6.1: Network example for spectral error and phase estimation. The equations that describe the behaviour of this system are shown in Eq. 6.1.

$$
\begin{aligned}
x_1(n) &= 0.33x_1(n-1) + 0.33x_5(n-1) + 0.33\eta_1 \\
x_2(n) &= 0.5x_1(n-1) + 0.2\eta_2 \\
x_3(n) &= 0.5x_2(n-1) + 0.2\eta_3 \\
x_4(n) &= 0.25x_3(n-1) + 0.25x_4(n-1) + 0.25x_5(n-1) + 0.2\eta_4 \\
x_5(n) &= 0.33x_4(n-1) + 0.33x_5(n-1) + 0.2\eta_5
\end{aligned}
\tag{6.1}
$$

In order to perform network structure inference, the spectral likelihood function is used in a network searching algorithm to evaluate all convergent structures that obey the network conditional independence and the acyclic path condition. For instance, suppose that it is of interest to find the parents of node x_4. This can be achieved by applying the spectral error function and see the resultant error spectrum. The parents that give the lowest error spectrum will be the more likely parents for node x_4.

Figure 6.3 shows the evaluation of error spectrum in Eq. 5.71 for the parent nodes for x_4. The evaluated convergent networks for this example are x_4 itself, $(x_4 \leftarrow x_3)$,

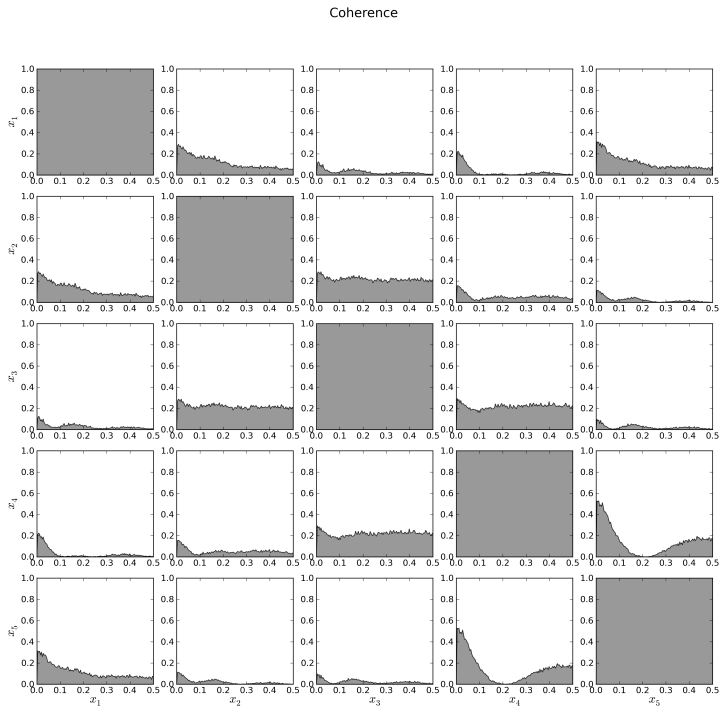

Figure 6.2: Coherence matrix of the network system in Eq. 6.1. Each row and column represents a node of the network shown in Figure 6.1.

$(x_4 \leftarrow [x_3, x_1])$, $(x_4 \leftarrow [x_3, x_5])$, $(x_4 \leftarrow [x_3, x_5, x_2])$, and $(x_4 \leftarrow [x_3, x_5, x_2, x_1])$. Notice how the error spectrum power decreases when the parent node x_1 is added to the network $(x_4 \leftarrow x_3)$ to form $(x_4 \leftarrow [x_3, x_1])$. Notice also that the error spectrum decreases even more with the network $(x_4 \leftarrow [x_3, x_5])$, where x_3 and x_5 are neighbours of x_4 and make x_4 independent of the rest of the network. This can be confirmed by noticing that when more network nodes are added into the vector of parents, the error spectrum does not decrease more as can be seen in the last two error spectra for the convergent networks $(x_4 \leftarrow [x_3, x_5, x_2])$, and $(x_4 \leftarrow [x_3, x_5, x_2, x_1])$. This means that information coming from x_1 and x_2 is irrelevant once x_3 and x_5 are given and that x_4 is conditional independent from the rest of the network given nodes x_3 and x_5. The same could be proven for the rest of the nodes in the network. Basically all the network structure searching algorithms use this principle to find the best network that produces the lowest regression error, with some constraints such as limiting the complexity of the network represented by the number of edges or cyclic path avoidance.

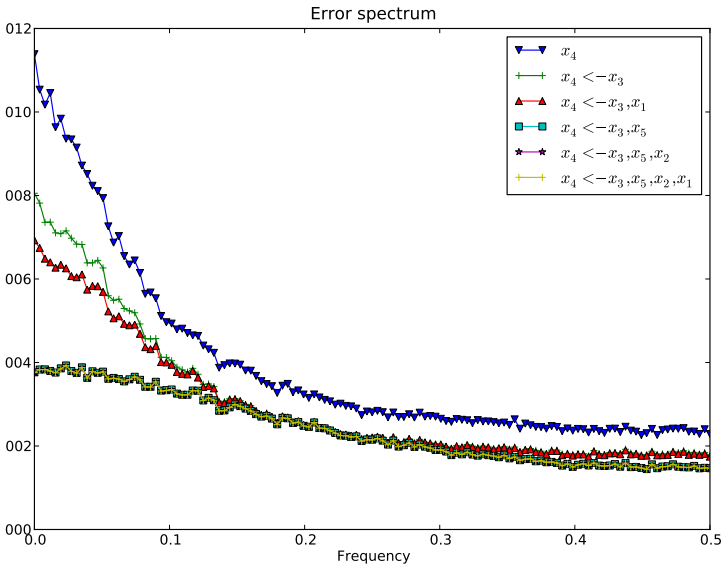

Figure 6.3: The error spectrum is shown for networks formed by node x_4 and different combinations of parents. It is possible to see how the addition of a new source node in the parent vector decreases the error spectrum until the two neighbouring nodes of x_4 are included, these are x_3 and x_5. Once these nodes are given, any information coming from other sources in the network is irrelevant. This shows that x_4 is conditional independent from the rest of the network given x_3 and x_5. Colour image can be seen in Figure B.13.

Figure 6.4 shows the cross-phase obtained from coherency in Figure 6.2. Since the delay in this system is on average constant among network nodes, most of the phase slopes are constant through all frequencies, meaning that we can fit the phase with a straight line. According to the Fourier linear model, a negative slope indicates that the row node causes the column node in the cross-phase matrix in Figure 6.4, or $row \rightarrow column$. All cross phases in Figure 6.4 show consistent slopes, with exception of the phase between $x_4 \leftrightarrow x_5$ which have a bidirectional interaction (cyclic path) that can not be modelled by a straight line.

The phase likelihood in Eq. 5.79 uses the cross-phase slope to infer time delay and map this information in a probability score, see Eq. 5.85. Figure 6.5 shows the phase spectrum between x_4 and its possible parents. A negative slope indicates that the causality model is correct.

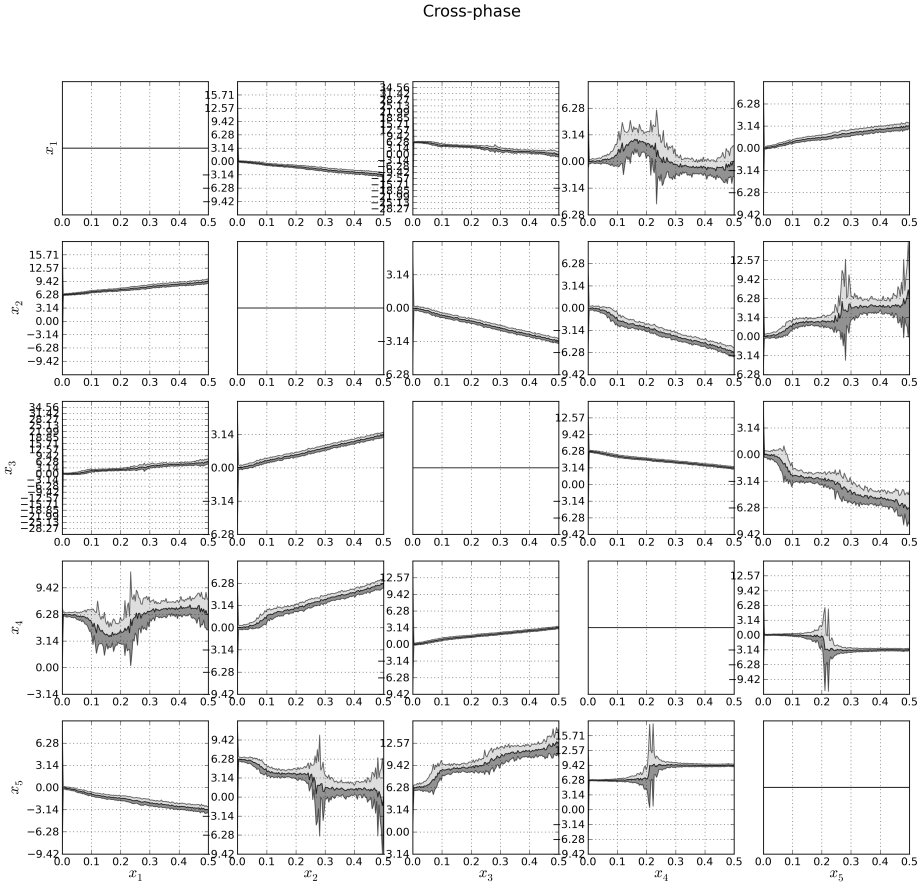

Figure 6.4: Cross-phase matrix for the system shown in Figure 6.1. The phase slope indicates causality for the network system. A negative slope indicates that a row node is a parent of a column node.

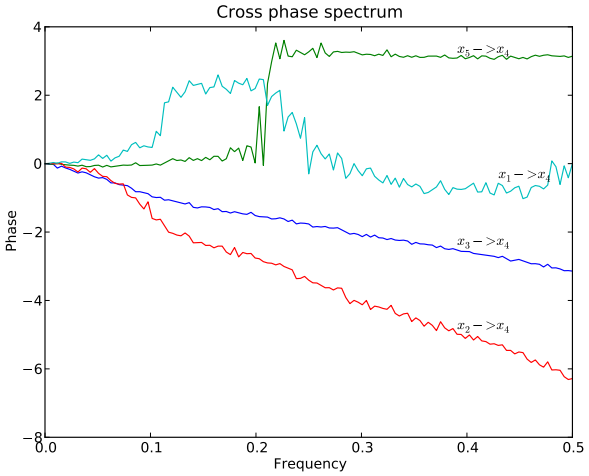

Figure 6.5: Phase spectra between the child node x_4 and the rest of the nodes in the network as parents. A negative slope indicates that the edge direction or model is correct.

6.2 Network models

For all experiments in this chapter, two network models are implemented. A MVAR(d) and a non-linear model NL(d), where d indicates delayed interaction among sources. All simulations use $d = 1$ and $d = 2$ which are sufficient to show the properties of FBNs for network structure inference. Two kind of experiments were implemented: first the inference of synthetic networks generated by the MVAR and NL models using FBNs and also DBNs for comparison purposes. The second set of experiments consist of the same set up but instead of using time series generated by the synthetic networks only, a simulated MEG recording environment is included. This means that the synthesized time series modulate electric dipoles placed in a brain mesh to generate fluctuating magnetic fields that are acquired by MEG SQUIDs. Then, brain imaging beamforming is applied in order to infer the current activity of the original brain electric dipoles and the network structure with this information. The experiments using the MEG recording simulated environment are designed to test how the inference of brain networks works after MEG beamforming.

In addition to network inference from the defined network models and the simulated MEG recording environment, different network sizes are also tested. In this thesis project, it is of interest to see how network inference using MEG beamforming behaves for large networks. Networks of 100 nodes are used to test the FBNs behaviour. Nevertheless, smaller networks of 10 nodes are also studied in order to see the network inference performance by keeping the analysis as simple as possible.

6.2.1 Linear case: Multivariate autoregressive model

The MVAR networks are implemented using the following set of equations in matrix notation;

$$\mathbf{x}(n) = \mathbf{A}\mathbf{x}(n-d) + \mathbf{g}(n) \ , \tag{6.2}$$

where $\mathbf{x}$ is the vector of generated time series, $\mathbf{g}$ is a vector of independent Gaussian noise sources, d is the variable time delay, and $\mathbf{A}$ is the MVAR coefficient matrix, the coefficients that define the edges and their strength. The coefficient matrix $\mathbf{A}$ is chosen by three steps. First a small-world network is created with 4

nearest neighbour edges for each node and random reconnecting edge probability of 0.25, see Section 1.5.1 for details. Second, the undirected matrix for the small world network is directed by taking the upper triangular part of this matrix. Finally the directed matrix is weighted where every weight value is chosen randomly with uniform distribution $-1 < a_{ij} < 1$ while keeping the MVAR system stable.

To generate small-world networks randomly, the Watts and Strogatz (Watts and Strogatz , 1998) method implemented in Networkx[1] python module is used. Figure 6.6 shows the 10-node small-world network for the MVAR experiments. This network is shown in two network layouts, a spring layout at the left and a circular layout at the right side of Figure 6.6. The undirected connectivity matrix is given directly by the random network generator function in the Networkx python module.

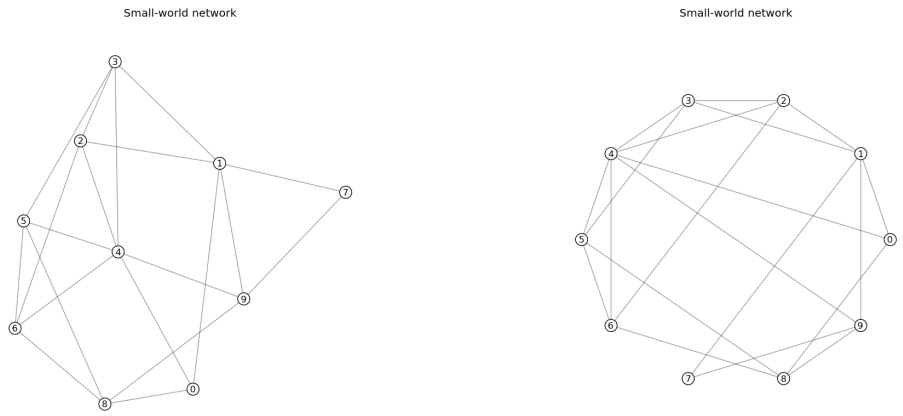

Figure 6.6: 10-node MVAR(d) network structure for simulations. Both networks show the same network structure but in two different layouts; spring and circular. Using this structure FBNs are test for network connectivity inference.

Figure 6.7 shows the directed connectivity matrix for the network in Figure 6.6. Since static BNs do not allow cyclic paths, the undirected networks were directed by zeroing out the lower triangular part of the undirected matrix. This procedure generates automatically a directed graph whose edges will be oriented in a clockwise fashion in a circular layout. For instance, for the network in Figure 6.6 the nodes x_1,x_4, and x_8 will point to x_0. Notice that with this arrangement x_0 will always be the lowest node in the network hierarchy.

[1]http://networkx.lanl.gov/index.html

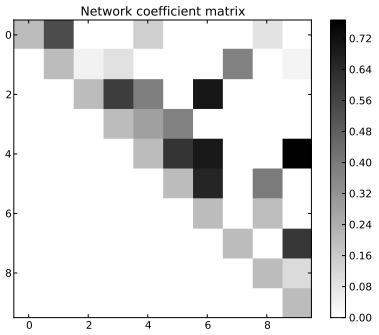

Figure 6.7: 10-node MVAR(d) coefficient matrix. This matrix represents the connectivity structure of the network in Figure 6.6 and also the edge weights.

Using the weighted and directed connectivity matrix $\mathbf{A}$ it is possible then to generate time series using the MVAR model. Figure 6.8 shows the correlation matrix of 10 time series generated by the 10-node MVAR(1) network. Notice that the interaction among the sources does not allow to infer the real shape of the network using simple correlation. But if the time-lagged partial correlation matrix (see Eq. 4.11) is computed, the network structure is recovered from the time series as seen in Figure 6.9.

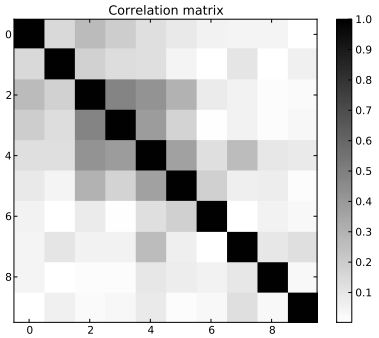

Figure 6.8: 10-node MVAR(1) correlation matrix. This matrix was estimated using time-series generated by the coefficient matrix $\mathbf{A}$ in Figure 6.7 and Eq. 6.2.

In the same manner as the 10-node MVAR network explained previously, a 100-node MVAR network was designed. This network is shown in Figure 6.10 in a spring and circular layout. Also, its connectivity matrix was directed by taking the upper triangular part and weighted randomly with a uniform distribution as can be seen in Figures 6.10 and 6.11.

Figure 6.12 shows the correlation matrix for the MVAR-100-node system. Again

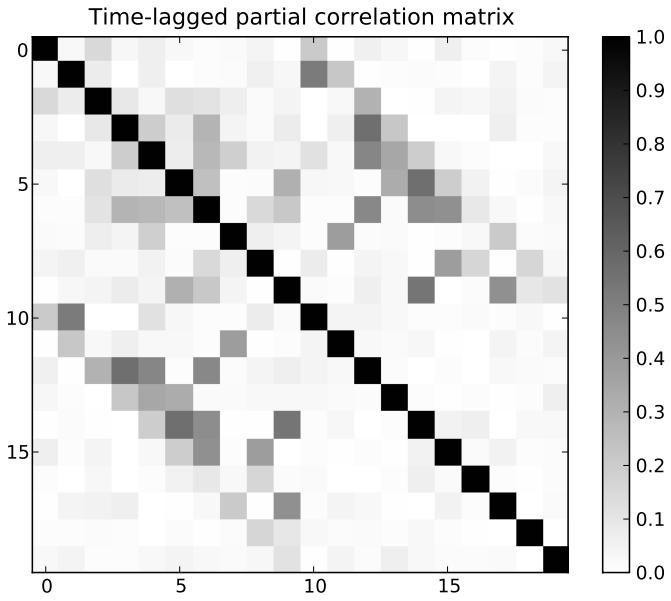

Figure 6.9: 10-node MVAR(1) time-lagged partial correlation matrix. This matrix was estimated using time series generated by Eq. 6.2 and the matrix in Figure 6.7 as coefficient matrix $\mathbf{A}$.

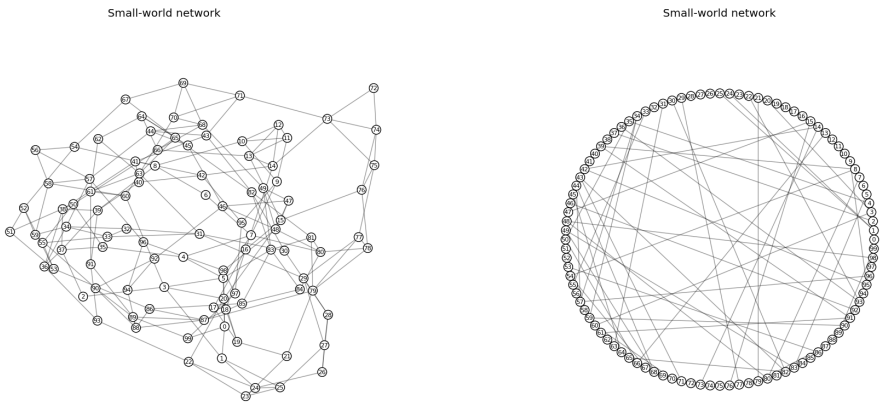

Figure 6.10: 100-node MVAR(d) network for simulations. This network uses the same connectivity model shown in Eq. 6.2 and it is used to test FBNs performance on networks with large number of nodes.

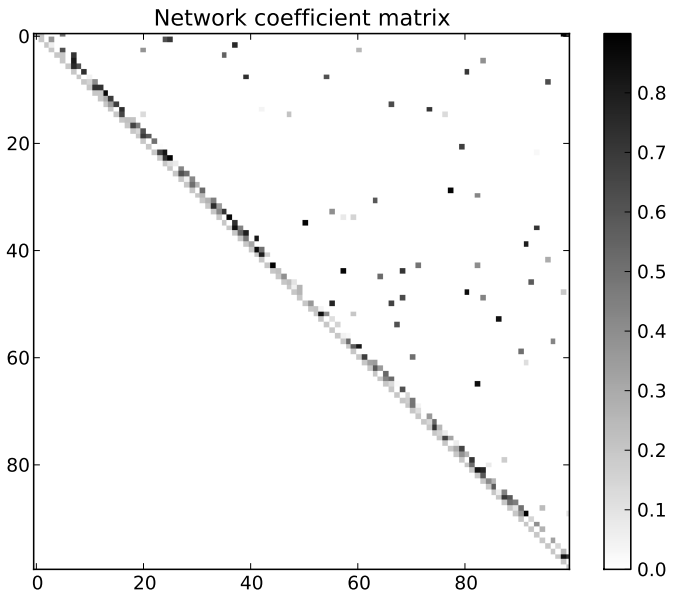

Figure 6.11: 100-node MVAR(d) coefficient matrix. This is the matrix $\mathbf{A}$ for Eq. 6.2 to synthesize time series from the MVAR(d) model.

as the 10-node network case, it is not possible to infer the network structure from the correlation matrix. However, it is already known that this is a linear system whose sources have interaction of one sample lag, and this knowledge allows to apply the time-lagged partial correlation matrix, shown in Figure 6.13. The structure can be inferred in the submatrix $\mathbf{B}$ of this matrix, which can be located at the rows $100 - 199$ and columns $0 - 99$.

6.2.2 Non-linear model

The non-linear simulations are performed with a modification of the original MVAR(d) model where the exponential function e is used as a non-linear operator. The synthesized networks using the non-linear model NL(d) are defined by the equation

$$\mathbf{x}(n) = \mathbf{A}\exp(\mathbf{x}(n-d)) + \mathbf{g}(n)\ , \tag{6.3}$$

where the coefficient matrix $\mathbf{A}$ is generated as in the linear case in Section 6.2.1 but with uniform distribution $0 < a_{ij} < 2.0$, and $\mathbf{g}(n)$ is a vector of Gaussian noise sources. Figure 6.14 shows the structure of the non-linear system for this simulation in both layouts, spring and circular.

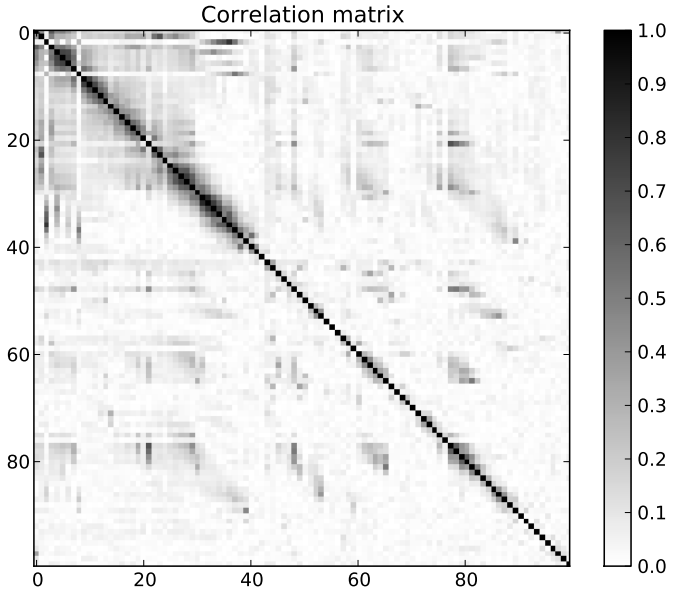

Figure 6.12: 100-node MVAR(1) correlated matrix. This matrix is estimated using time series synthesized from the MVAR(1) model in Eq. 6.2.

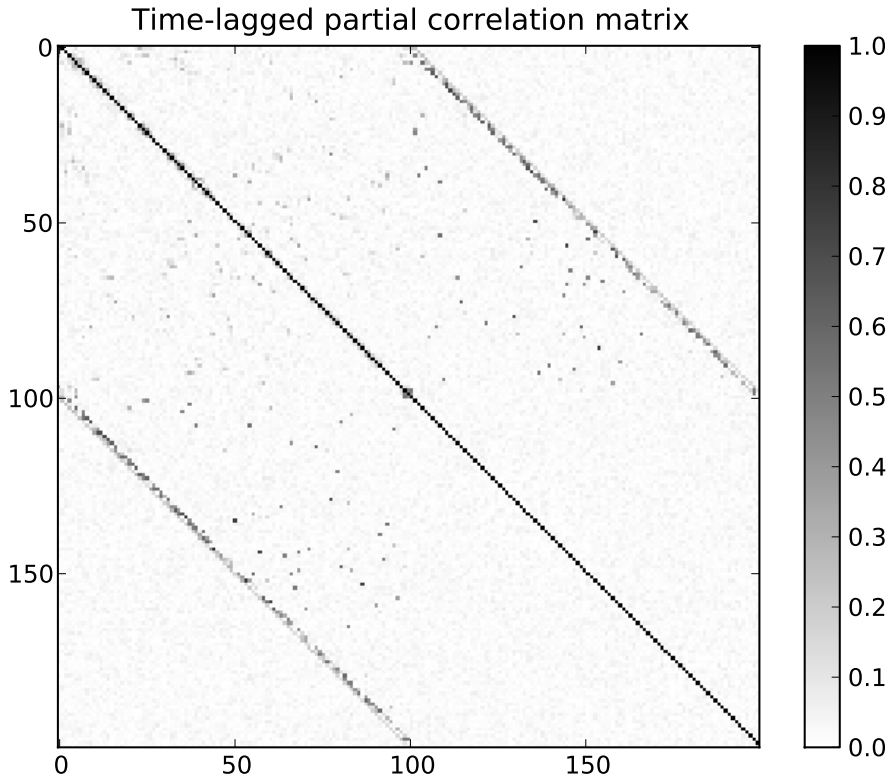

Figure 6.13: 100-node MVAR(1) time-lagged partial correlation matrix estimated from the MVAR network system in Figure 6.10.

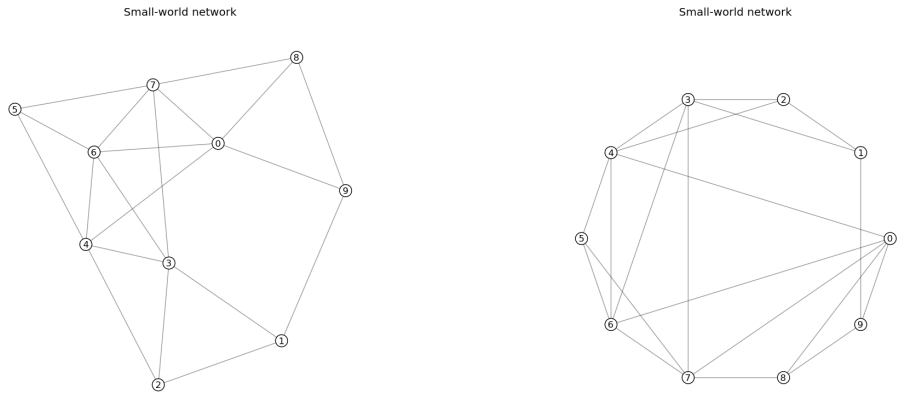

Figure 6.14: 10-node NL(d) network for the non-linear simulations. Both networks represent the same structure shown in two different layouts; spring and circular layout.

Figure 6.15 shows the coefficient matrix **A** for the network in Figure 6.14. Using this matrix and the NL model in Eq. 6.3 time series can be generated. The correlation matrix of these time series are shown in Figure 6.16. As in the MVAR case, the structure of the network can not be inferred by simply analysing the correlation matrix. However, it is possible to compute the time-lagged partial correlation matrix to infer the structure. The time-lagged correlation matrix is shown in Figure 6.17.

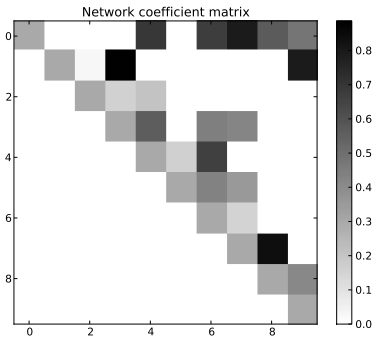

Figure 6.15: 10-node NL(1) coefficient matrix for the non-linear experiments. This matrix and its coefficients define the connectivity structure and edge weights of the non-linear network.

In order to see the behaviour of FBNs in larger non-linear systems, a 100-node network is also implemented. Its structure is shown in Figure 6.18 and its weighted connectivity matrix **A** is shown in Figure 6.19. By this point the correlation matrix and the partial correlation matrix are omitted since their results are similar to the previous cases, and do not offer more insight to the comprehension of the NL(d)

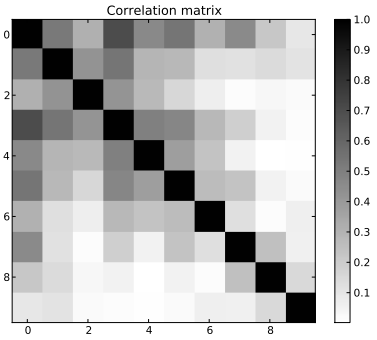

Figure 6.16: 10 node NL(1) correlation matrix estimated from time series generated using matrix **A** in Figure 6.16 and the NL(1) model in Eq. 6.3.

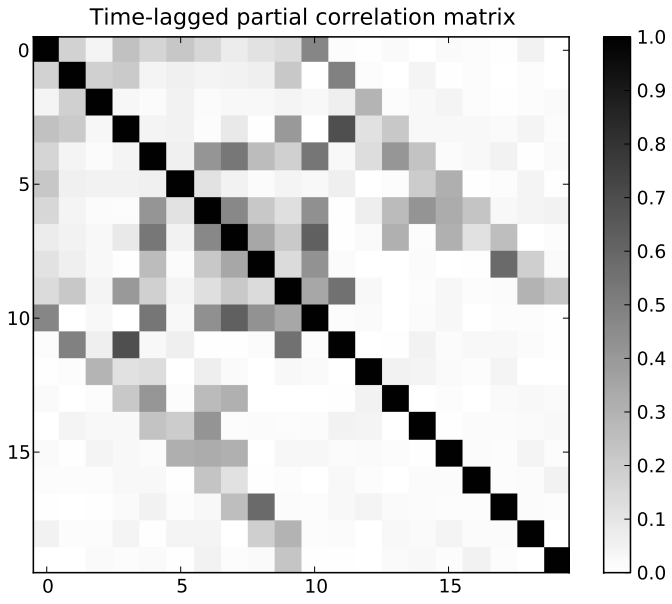

Figure 6.17: 10-node NL(1) time-lagged partial correlation matrix.

system.

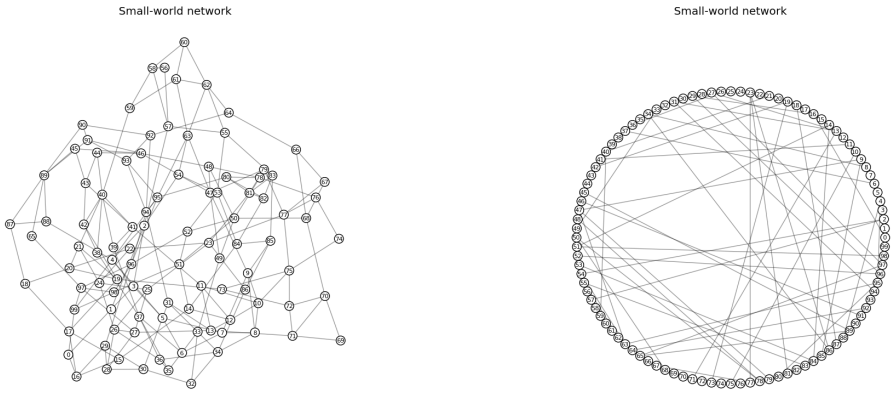

Figure 6.18: 100-node NL(d) network for simulations.

6.3 Simulated brain networks

The main interest of this project is to design a connectivity technique for MEG studies and for this reason it is of paramount importance to test how FBNs behave in MEG studies. This is difficult to perform on real datasets. Using real datasets will not allow to compare the performance of FBNs inferring networks because every network connectivity technique highlights different properties of the system under study depending on the model or assumptions followed. Hence, FBNs can not be compared using other connectivity techniques, unless the real brain network were known in advance.

In order to test FBNs' performance in MEG studies, a simulated MEG recording environment is implemented. This MEG environment consists of a brain mesh with the shape of the brain cortex, simulated electric dipoles, and SQUIDs sensors which have the same sensor layout of the 4D Neuroimaging Magnes 3600 with 248 sensors. The cortex is a VTK mesh/polyhedron of 263924 triangular faces for the left hemisphere and 265694 faces for the right hemisphere and it was taken from the 3DSlicer[2] software and its tutorial files.

[2]http://www.slicer.org/

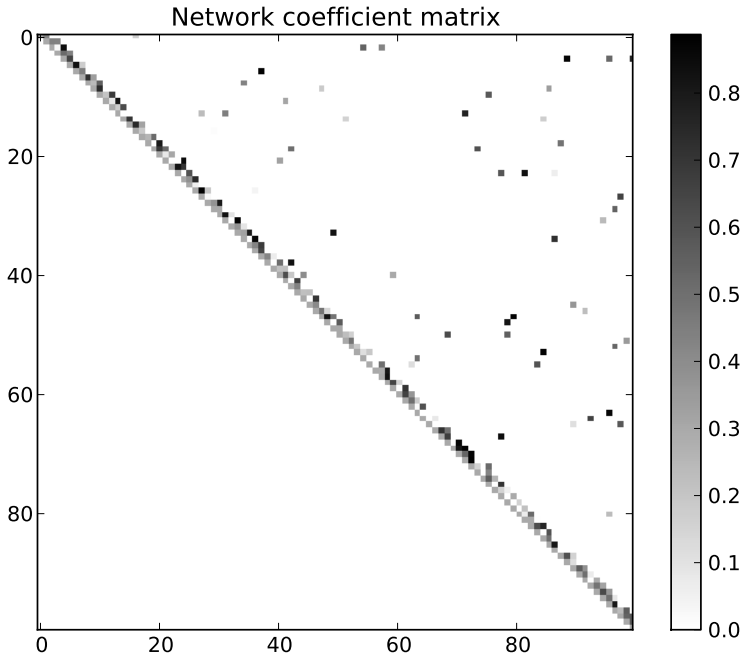

Figure 6.19: 100-node NL(d) coefficient matrix $\mathbf{A}$.

A straight forward way to set up simulations is assuming that every triangular face in the cortex polyhedron is a potential place for cortical sources whose electric dipoles are orthogonal to the faces of the polyhedron. Nevertheless, using such amount of positions increases the computation for the beamforming analysis. Recall from Section 2.5.2 that a new spatial filter must be estimated for each position within the tessellated brain, which in this case will be of 529618 spatial filters. For this reason the original brain mesh was decimated in order to have fewer faces and in consequence fewer spatial filters to compute. Using Mayavi2[3] software, the brain cortex mesh was decimated from its original 529618 faces to 25416 triangular faces for the right hemisphere and 25376 faces for the left hemisphere. This reduces the number of estimated spatial filters to 50792.

Having the network systems defined in Sections 6.2.1 and 6.2.2, N number of faces from the brain mesh are chosen randomly where N is the number of sources to simulate (10 and 100 nodes). The brain network connectivity is defined by the original network systems defined in Sections 6.2.1 and 6.2.2, but the spatial locations of the nodes are chosen randomly. Once the source positions are designated, their

[3]http://code.enthought.com/projects/mayavi/

positions are fixed. The chosen brain network configurations are used for all the remaining experiments in this chapter.

Figure 6.20 shows the 10-node and 100-node network for the MVAR network systems. The thickness of the network edges is proportional to the edge weights.

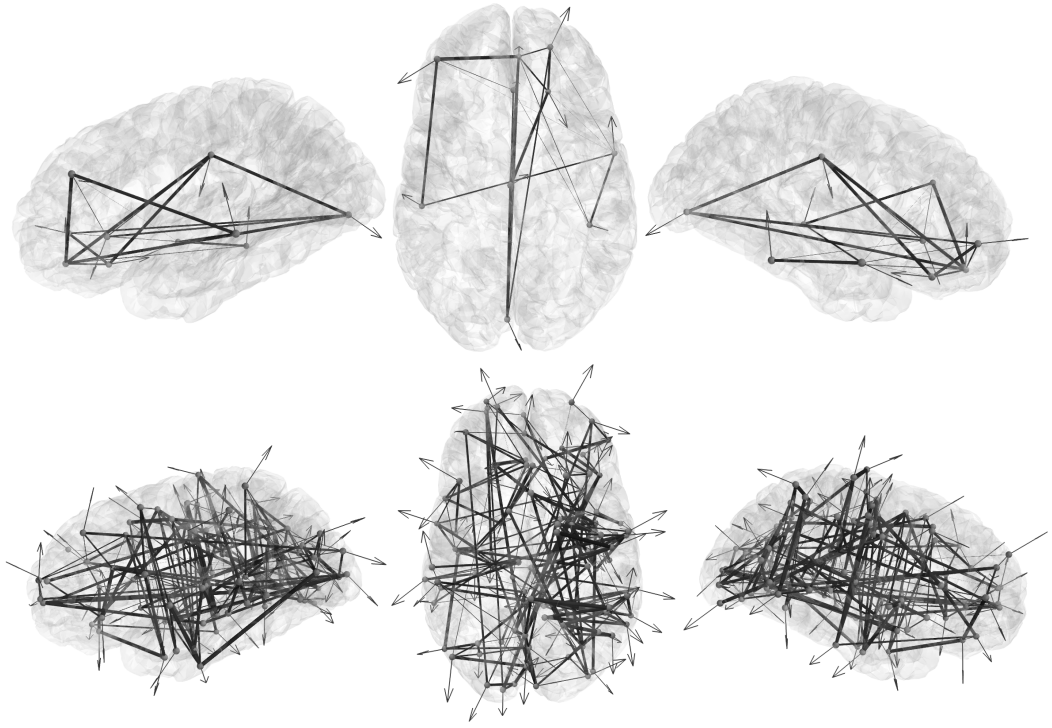

Figure 6.20: Generated network structures in the brain cortex mesh for the network structure inference experiments using MVAR(d) model. The upper row shows the 10-node network and the 100-node network is shown in bottom row. Colour image can be seen in Figure B.14

The NL(d) cases are also implemented. The reason for implementing a NL case is mainly because it is important to study the FBNs' performance when the model is not linear. FBNs are based on the Fourier transform, hence there are some assumptions automatically taken such as stationarity and linearity. FBNs assume that the interaction among brain regions is linear which is not true for the brain as in all phenomena in nature. Nevertheless, linear techniques are still being used in Neuroscience and Neuroimaging and much of the accumulated knowledge about the brain today has been achieved by linear techniques.

The simulation set up presented in this section can be summarized as follows: First, network inference performance from synthesized networks using linear MVAR(d)

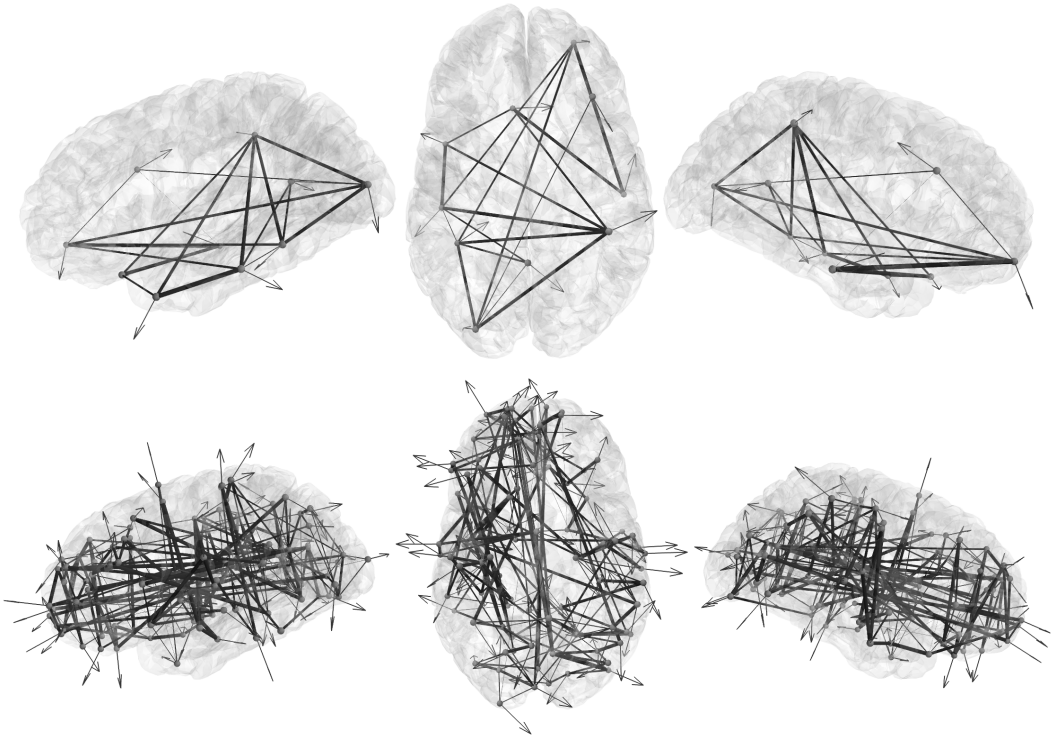

Figure 6.21: Generated network structures in the brain cortex mesh for network structure inference experiments using the NL(d) model. The upper row shows the 10-node network and the 100-node network is shown at the bottom row. Colour image can be seen in Figure B.15

and non-linear NL(d) models with two different time delays $d = 1$ and $d = 2$ is studied. These simulations show the performance of FBNs assuming full access to the signal generating sources. Nevertheless, in MEG studies it is not possible to have access to the sources, that is why the MEG recording simulated environment is implemented. Using the MEG implementation, FBNs are tested after the MEG has recorded the brain's magnetic fields and the LCMV beamformer has been applied, where beamforming has the function of recovering the activity of the original sources. The next section explains how the simulated MEG recording environment is implemented for brain source acquisition and also localisation of brain sources.

6.4 MEG and Beamforming

This section describes the MEG recording environment designed for the simulations. The MEG implemented here is composed of 248 sensors that are able to record magnetic fields that crosses the magnetometers or SQUIDs, these are shown in Figure 6.22 as small spheres surrounding the brain mesh. The layout of the SQUIDS is the same of the 4D Neuroimaging Magnes 3600 located at the York Neuroimaging Centre[4] (YNiC). As explained in Section 2.3, only the magnetic field components parallel to the SQUIDS or orthogonal to the magnetometers' loops are recorded, see Section 2.5.2 for details.

Using the MVAR(d) and NL(d) models and networks defined in Section 6.2 and Section 6.3, it is possible to generate time series that modulate electric dipoles placed in the brain cortex mesh and acquire their magnetic fields.

It is important to notice that in real experiments, the number of sources and their position is unknown. Hence, in a real brain imaging using MEG, the researcher will have to answer first how many sources there are and their position in the brain volume before attempting brain activity extraction using the virtual electrodes. There are several techniques and protocols to achieve this such as designed paradigms to compare resting state and a cognitive task of interest in order to find significant statistical differences between both conditions. The places with higher significance

[4]https://www.ynic.york.ac.uk/

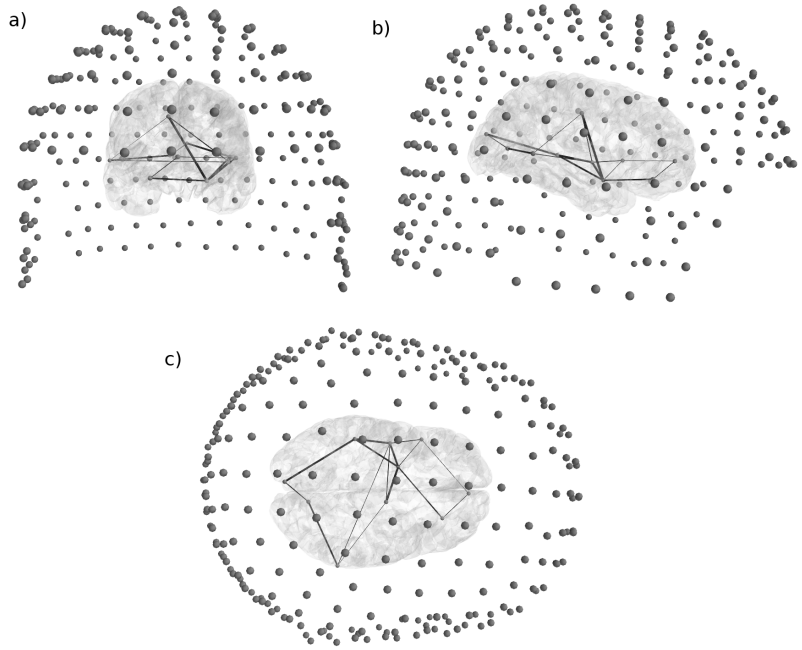

Figure 6.22: Simulated MEG acquisition environment. The figure shows the 248 SQUIDs as small spheres surrounding the brain mesh. The layout of the SQUIDs is the same of the real 4D Neuroimaging Magnes 3600 magnetoencephalographer.

are considered true activated sources for the particular cognitive task tested. Another popular technique is independent component analysis, which tries to find the original independent sources within all sources in the brain. Then areas of the brain correlated with significant independent components are taken as true source activity.

Since the experiments are done with simulated networks where every node is a brain source, the positions of all nodes are already known. Hence for the matters of this thesis, source inference and techniques to find them will not be discussed or tested. For network structure inference simulations, the MEG virtual electrodes are used at those places where it is known in advance the existence of a brain source. In the next section, beamforming simulations are shown for the two brain networks, linear MVAR and the non-linear NL.

6.4.1 Independent sources

The LCMV beamforming is originally designed to find the position and power of independent or at least uncorrelated sources. According to the beamforming deriva-

tion, its performance decreases if two or more brain sources are highly correlated. This makes beamforming "fuse" the correlated sources in a single one, because from the beamforming perspective, two or more sources with almost equal activity must have a common single driving source. For this reason, it is interesting to see first how the beamforming method implemented here works for independent sources. Figure 6.23 shows the LCMV beamforming results where the brain cortex mesh faces have been coloured coded according to the power obtained from the NAI index in Eq. 2.45 (red=highest and blue-lowest).

The network structure used for this simulation is the one for the 10-node MVAR(d) network, and the sources were modelled as independent Gaussian noise that represent cortical activity. By comparing the beamforming maps in Figure 6.23 with the linear structure in Figure 6.20 it can be seen that for the 10-node network the areas with larger NAI index correspond to the areas where the network sources were placed, indicating that the original sources can be localized.

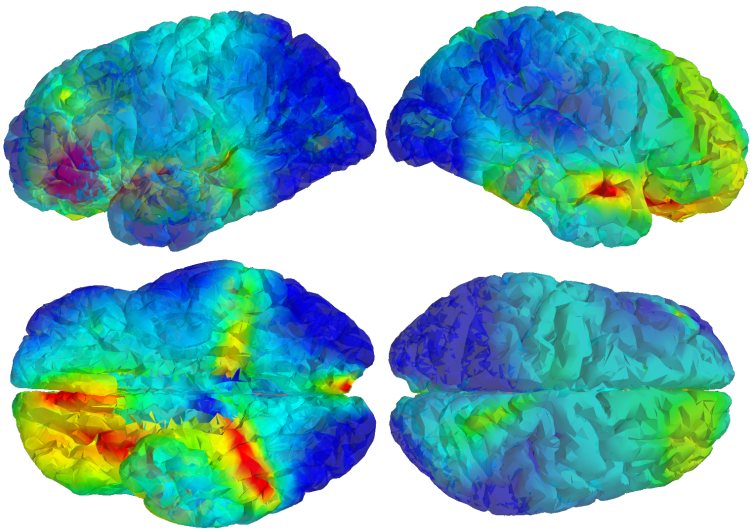

Figure 6.23: Beamforming maps from independent sources. The current dipoles were modulated by independent Gaussian noise sources and the source arrangement is the one of the 10-node linear network in Figure 6.20. The image was programmed and created using Python-Mayavi2. Colour image can be seen in Figure B.16.

Even when the sources can not be easily seen from the NAI map in Figure 6.23, it is still possible to apply the virtual electrodes since the positions of the original sources is known in advance. This approach is used in this thesis to infer networks

from the MEG environment.

6.4.2 Linear networks

Figure 6.25 shows the same network structures as in Figure 6.23 but now using the MVAR(1) model in Eq. 6.2 and its respective weighted connectivity matrix. As can be seen, the 10-node beamforming NAI cortical map is very similar to the one in Figure 6.23 where the NAI largest index areas coincide with the network node positions at some degree.

Figure 6.24 shows correlation values between the extracted brain activity using beamforming on those regions where it is known in advance the existence of network sources and the original source activity. The correlation matrix was computed using the vector $[Beamforming\ results\ ;\ Original\ signals]$ and it is a 20×20 correlation matrix which can be segmented as

$$C = \begin{bmatrix} \mathbf{B} & \mathbf{A} \\ \mathbf{A} & \mathbf{O} \end{bmatrix} , \tag{6.4}$$

where the matrix $\mathbf{B}$ is as 10×10 correlation matrix of the beamforming extracted sources using the virtual electrodes, $\mathbf{O}$ is also a 10×10 correlation matrix but in this case represents the correlation of the original source activity. The matrix of interest for this experiment is $\mathbf{A}$ which is the cross matrix between the beamforming extracted signals and the original sources. As can be seen in Figure 6.24, the largest values in $\mathbf{B}$ are in its diagonal, indicating that the source extraction was performed correctly, although there is one source that was not inferred correctly and showed low correlation. Notice also that the cross correlations among the sources in $\mathbf{B}$ are decreased compared with the original source cross correlations in matrix $\mathbf{O}$.

Figure 6.26 shows the correlation matrix between the beamforming extracted signals and the original sources for the 100-node MVAR(1) brain network. As in the previous case, this matrix can be divided in four submatrices $\mathbf{B}$, $\mathbf{O}$ and two $\mathbf{A}$ as shown in Eq. 6.4. The matrix of interest for this experiment is matrix $\mathbf{A}$ which is the cross correlation between the beamforming extracted signals and the original sources. As can be seen the signal extraction was poor with only some high

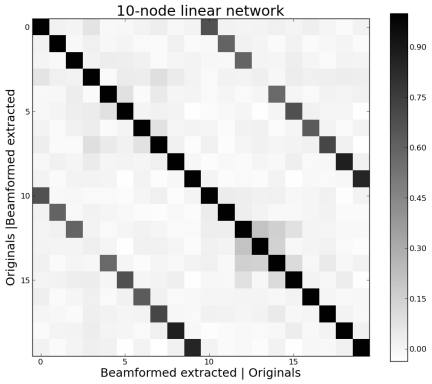

Figure 6.24: Beamforming vs. Originals correlation matrix for the 10-node MVAR(1) system. The correlation matrix was computed by concatenating the original sources with the extracted ones by beamforming.

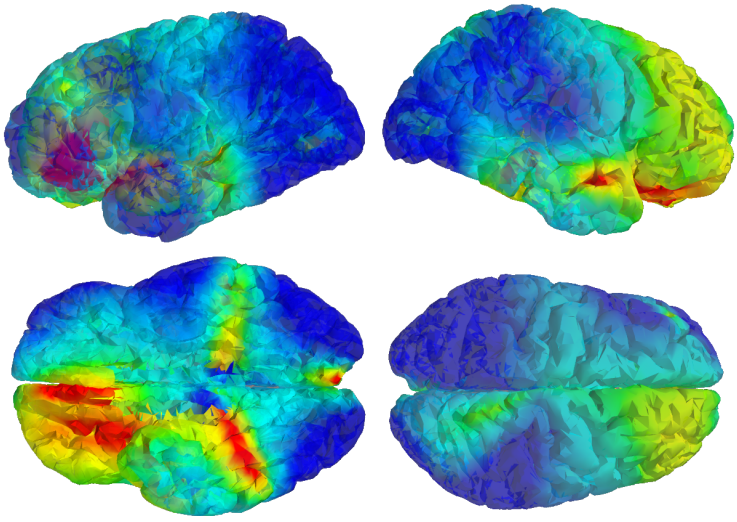

Figure 6.25: Beamforming maps using the 10-node MVAR(1) brain network shown in Section 6.20. The image was programmed and created using Python-Mayavi2. Colour image can be seen in Figure B.17.

correlation values in the diagonal. Figure 6.27 shows the NAI map for the 100-node MVAR(1) network. The high density and number of the network sources affect their extraction.

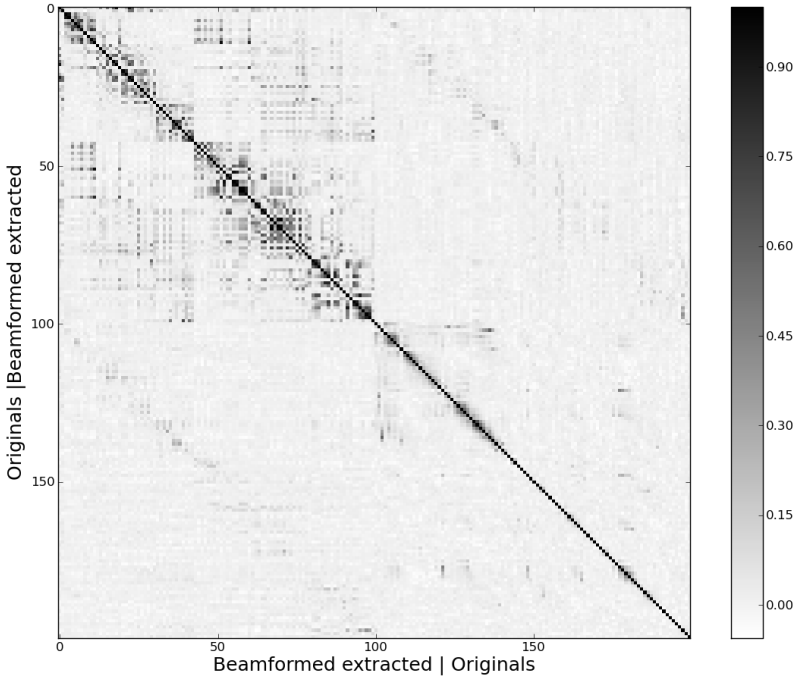

Figure 6.26: Beamforming vs. Originals correlation matrix for the 100-node MVAR(1) system. As in the 10-node case, this matrix was created by concatenating the original sources with the beamforming extracted ones.

6.4.3 Non-linear networks

For the NL network systems similar results were obtained. The 10-node NL(1) network showed localised activity in the NAI map which coincides in some degree with the original network structure. Figure 6.28 shows the correlation matrix between the beamforming extracted sources and the original sources for the 10-node NL(1) network. As can be seen in submatrix **A**, the extraction was successful with exception of one source.

Figure 6.30 shows the correlation matrix but now for the 100-node NL(1) network. Matrix **A** which has the cross correlation values between the beamforming extracted sources and the original ones, shows that beamforming has a poor source

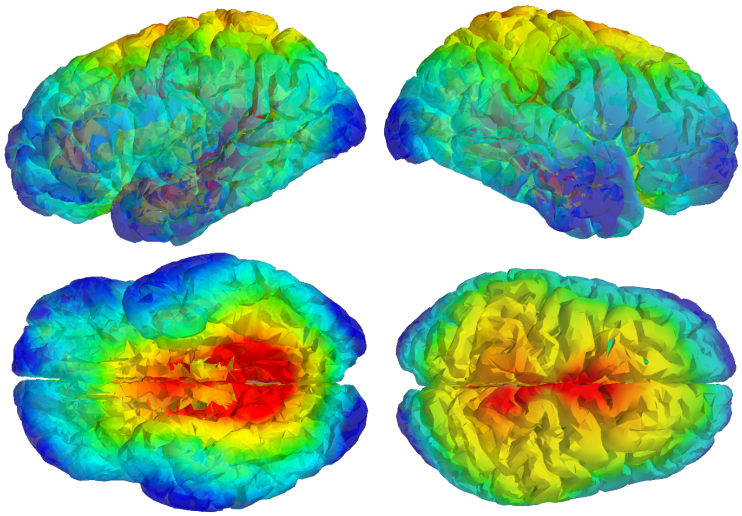

Figure 6.27: Beamforming map using the 100-node MVAR(1) network model shown in Figure 6.20. Colour image can be seen in Figure B.18.

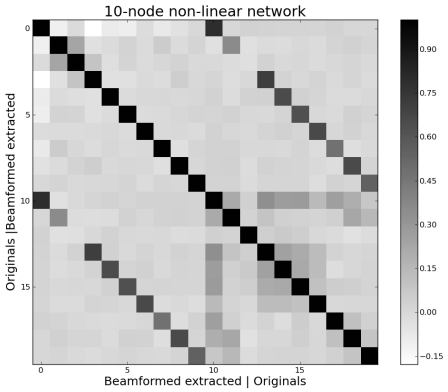

Figure 6.28: Beamforming vs. original source correlation matrix for the 10-node NL(1) network system.

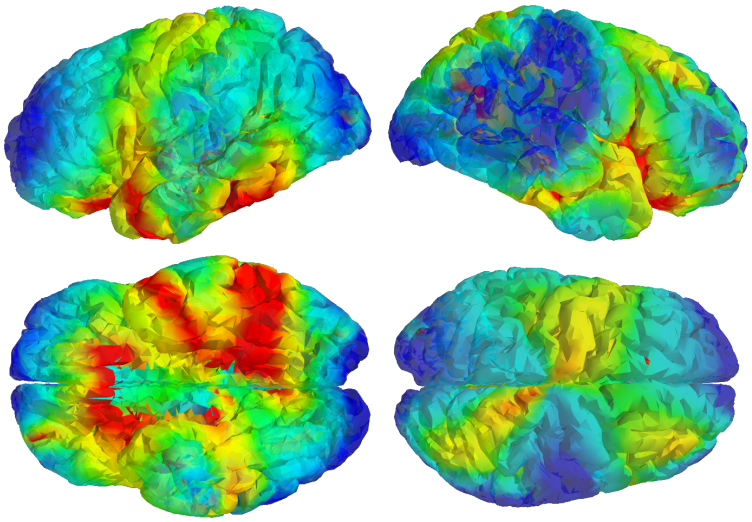

Figure 6.29: Beamforming maps using the 10-node NL(1) network. The original network structure system is depicted in Figure 6.21. Colour image can be seen in Figure B.19.

extraction, since the diagonal elements of **A** are low value correlation coefficients.

The NAI map for the 100-node network case is shown in Figure 6.31. As in the linear case, the large number of sources produces generalized activity through the entire brain cortex and not single isolated sources from the original brain network can be seen from the NAI map.

The beamforming results using the 10-node networks for both cases, linear and non-linear, showed to be successful. The correlation between the extracted sources by beamforming with the original sources is high. For the 100-node networks, this was not the case as shown by the low correlation values in matrix **A**. Low correlation indicates that network structure inference using beamforming for the 100-node network is difficult.

6.5 Bayesian networks

Using the networks designed in previous sections, now it is possible to run experiments for network inference for both the network systems alone and also using the MEG recording environment. The FBN algorithm for structure searching will be tested here using the four network systems designed and for comparison purposes DBNs using Gaussian linear model will also be tested. It is not the intention of this

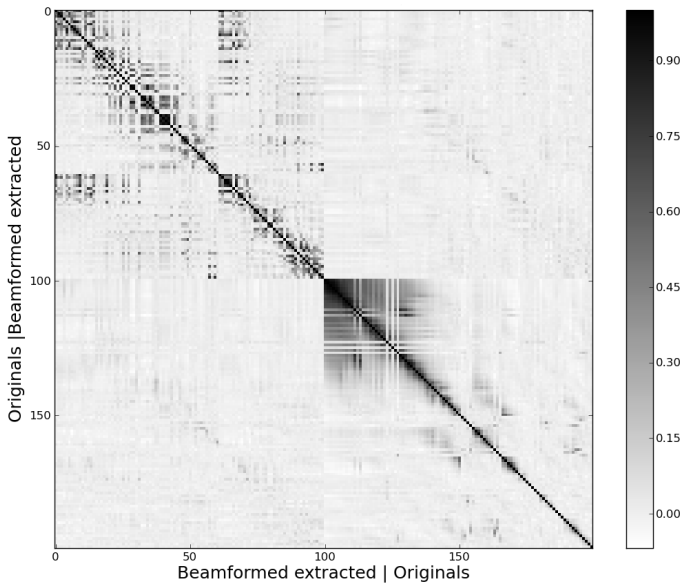

Figure 6.30: Beamforming vs. original network correlation, 100-node NL(1) system. The original network structure system is depicted in Figure 6.21.

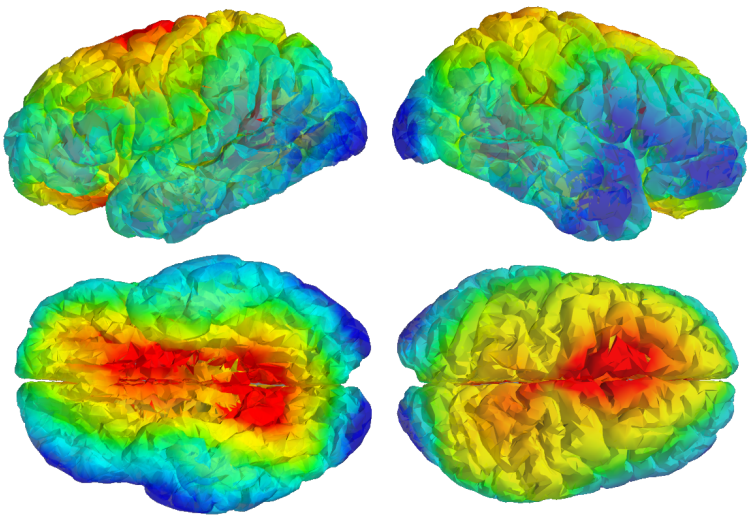

Figure 6.31: Beamforming map for the 100-node NL(1) network. The original network structure systems are depicted in Figure 6.21. Colour image can be seen in Figure B.20.

section to make FBNs and DBNs compete. The reason for applying DBNs is to have a background that tells us how good or bad is the FBNs' performance.

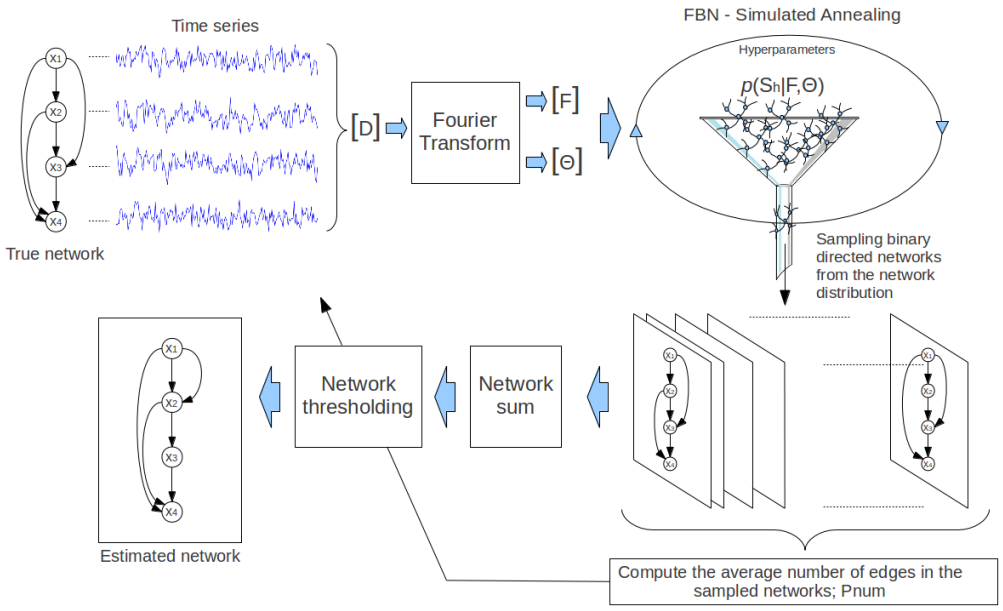

Figure 6.32: Fourier Bayesian network (FBN) simulation and experiment scheme. The figure shows the steps followed for most of the simulations in this section. First a network model is chosen from which times series are synthesized. Then, these time series represented as the data matrix **D** are Fourier transformed to obtain the power matrix **F** and the phase matrix **Θ**. Both matrices are fed into the FBNs using the simulated annealing algorithm as optimization method in order to sample networks from the network distribution close the distribution maxima. For these experiments 100 network are sampled. The network matrices are then added up to create a matrix that summarizes the final estimated network from which a "champion" network is chosen using a network edge threshold.

The results in this section start with single experiments where only one run of the algorithm is plotted. Then statistical results obtained by several repetitions of the algorithm are given. The statistical results give an idea of the average performance of FBNs for the analysed networks. A diagram explaining the experiments in this section is shown in Figure 6.32.

Figure 6.33 shows the coherence matrix for the 10-node MVAR(1) network in Figure 6.6. From the figure, it can be seen that x_3 shows high coherence values with sources x_4, x_5, x_6, and x_7. Nevertheless, although coherence allows to infer relation or connectivity among the universe of sources, it does not give any information about causality of these connections. This can be assessed by analysing the phase

information of coherency, see Section 5.6.3.

For all coherence and phase matrices in rest of thesis, the length of the data segments is $M = 20 \times\ 256$, where 256 samples are the periodogram's time window with 75% overlapping.

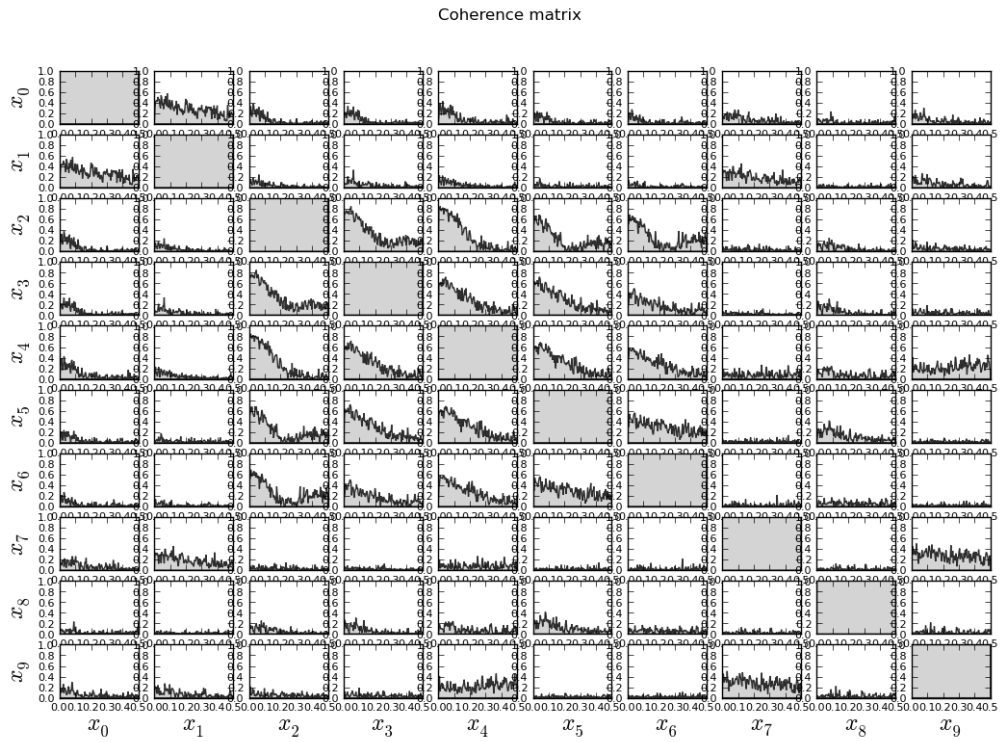

Figure 6.33: 10-node MVAR(1) coherence matrix. The coherence matrix was computed using time series generated from the 10-node MVAR(1) system. The time series length was of 20×256 samples. This length will be constant for all simulations in this section. Each row and column index of the matrix represents the network source it belongs to.

Figure 6.34 shows the cross phase spectrum (wrapped frequency phase) obtained by taking the argument of coherency. As explained in Section 5.6.3, a negative phase slope indicates that the row source in the matrix causes or leads the column source. For instance, Figure 6.34 shows that x_1 causes x_0. This result was expected. Recall from Section 6.2 that causality was created by taking the upper triangular part of the weighted connectivity matrix. Hence, if a connection exist between two sources in the analysed network, causality will always be defined from the node with higher index to the node with lower index. This causality definition will be helpful for analysing the simulation results in this section. Using this phase slope it is possible

to infer the time delay between the two sources, see Section 5.6.3 for more details.

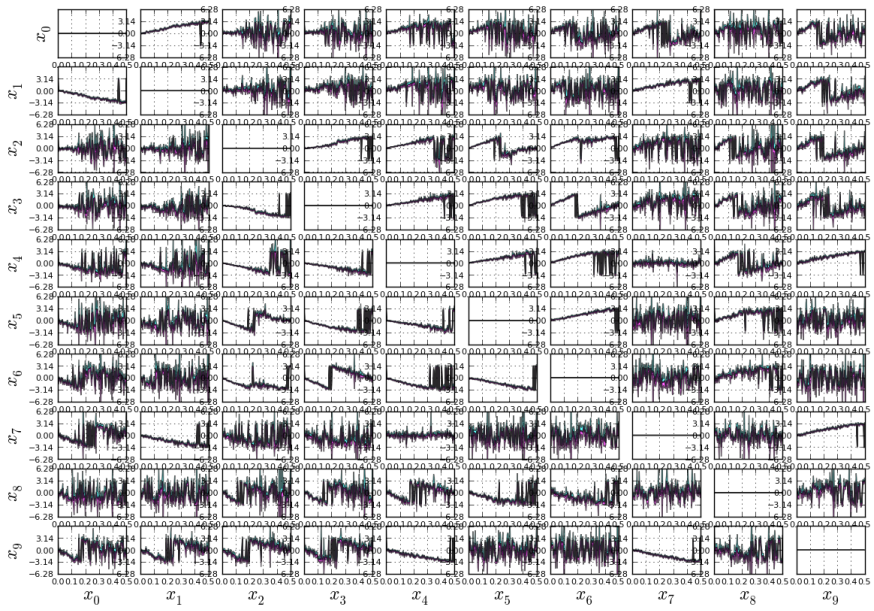

Figure 6.34: 10-node MVAR(1) cross-phase matrix. The cross-phase information shows the causality interaction among the network sources. This causality can be estimated by the phase slope. If this slope is negative, it means that the source represented by the matrix row causes the source represented by the matrix column.

In order to infer the 10-node MVAR(1) network, the FBN algorithm explained in Section 5.6 was run using times series obtained from this network system. The convergence curves for the FBN algorithm are shown in Figure 6.35. Figure 6.35-top-left shows the score values $|\log(p(\mathbf{F}|S_h)p(\mathbf{\Theta}|S_h)p(S_h))|$ which is the numerator of the Metropolis rule in Eq. 5.104 for the simulated annealing. In this algorithm all networks are evaluated and then the Metropolis rule decides if the new network is taken as a true sample of the network distribution or discarded. The accepted network scores are shown in Figure 6.35-top-right. The flat segments in this curve are the algorithm iterations when the proposed networks were rejected by Metropolis, and the curve's steps indicate when the proposed network was accepted. Hence a "noisy" curve in this case means a good performance for the Metropolis algorithm. The third graph, Figure 6.35-bottom shows the number of edges of the estimated network. From this graph it can be seen how this number converges to a constant value with an acceptable variance.

The hyperparameters for the simulated annealing algorithm are given in Table 6.1 for the 10-node MVAR(1) and MVAR(2) network systems. The hyperparameters define the simulated annealing algorithm's behaviour which is in charge of sampling networks from the network distribution given the data matrices, $p(S_h|\mathbf{F}, \mathbf{\Theta})$. The hyperparameters' function and their interpretation are explained in Appendix A.

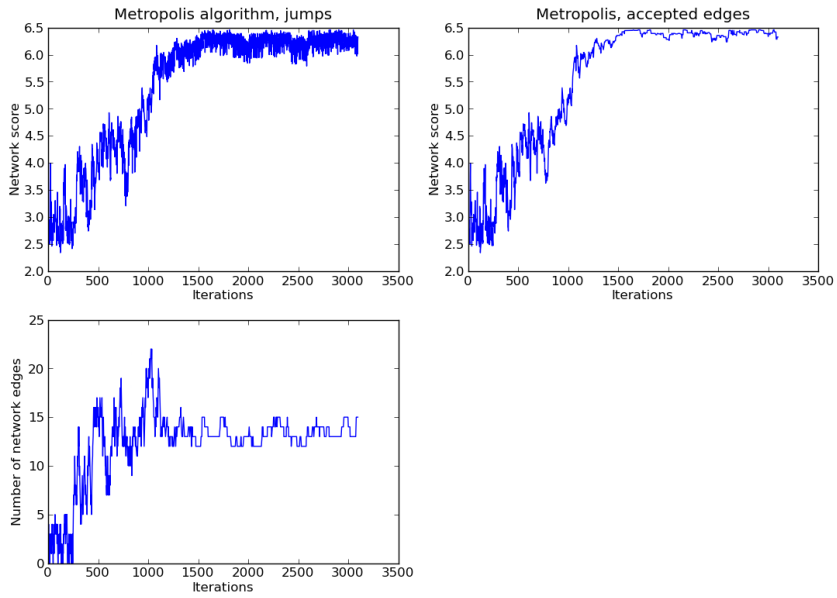

Figure 6.35: 10-node MVAR(1) simulated annealing performance curve. The top-left curve shows the score $|\log(p(\mathbf{F}|S_h)p(\mathbf{\Theta}|S_h)p(S_h))|$ for the sampled networks. The top-right curve shows the score of accepted network by simulated annealing. The bottom curve shows the number of edges of the accepted network.

Table 6.1: Hyperparameters for simulated annealing: 10-node network systems, MVAR and NL.

numruns:	10000		initemp:	2.0
coolruns:	1500		fintemp:	0.02
burnedruns:	8000		alpha:	0.02
			beta:	0.04

The last iterations of the simulated annealing algorithm (burnedruns) are used for network inference where the first 100 accepted networks are saved. Then these 100 sampled networks are added up and thresholded to obtain a champion network. Figure 6.36-left shows the estimated average network for the 10-node MVAR(1) system, here the threshold *Pnum* is of 13 edges, while the true weighted network for this system is shown at the right with all its weighted edges where the width of

the arrows are proportial to their weight. Notice that the FBN method was able to infer correctly the network system with exception of the weak edges which were ignored by the algorithm.

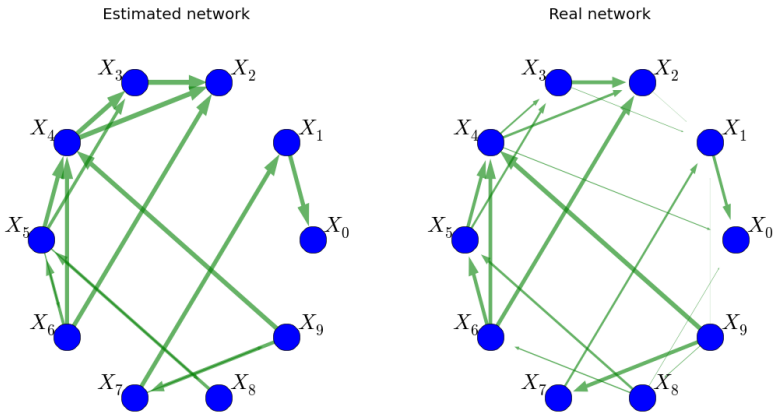

Figure 6.36: 10-node MVAR(1) estimated network. The estimated network is shown at the left and the true weighted network is shown at the right. Notice that the FBN method was able to infer correctly the network with exception of the weak edges which were ignored. The edge threshold for this simulation was $Pnum = 13$.

The results for the 10-node MVAR(2) system are shown in Figure 6.37, 6.38, 6.39, and 6.40 and the simulated annealing algorithm uses the hyperparameters shown in Table 6.1. Figure 6.37 and 6.38 shows the cross coherence and cross phase matrices for this network system. Notice how the coherence spectrum changed with respect to Figure 6.33 and the phase slopes have higher values than the ones in Figure 6.34.

The performance curves are shown in Figure 6.39. Notice that the average edge of the sampled networks when the algorithm reaches stability is 12 edges. The flat segments in this curve represent the iterations where the Metropolis rule rejects the proposed networks.

Figure 6.40 shows the estimated network at the left of the figure, while the true weighted network is shown at the right. Similarly to the MVAR(1) case, all strong edges from the true network were found correctly while the weak ones were ignored by the FBN method. Recall that in Figure 6.40-right the width of the arrows are proportional of the edge strength. The estimated network is created by adding up 100 sampled networks which were taken after the simulated annealing algorithm reached stability and the threshold applied is $Pnum = 13$.

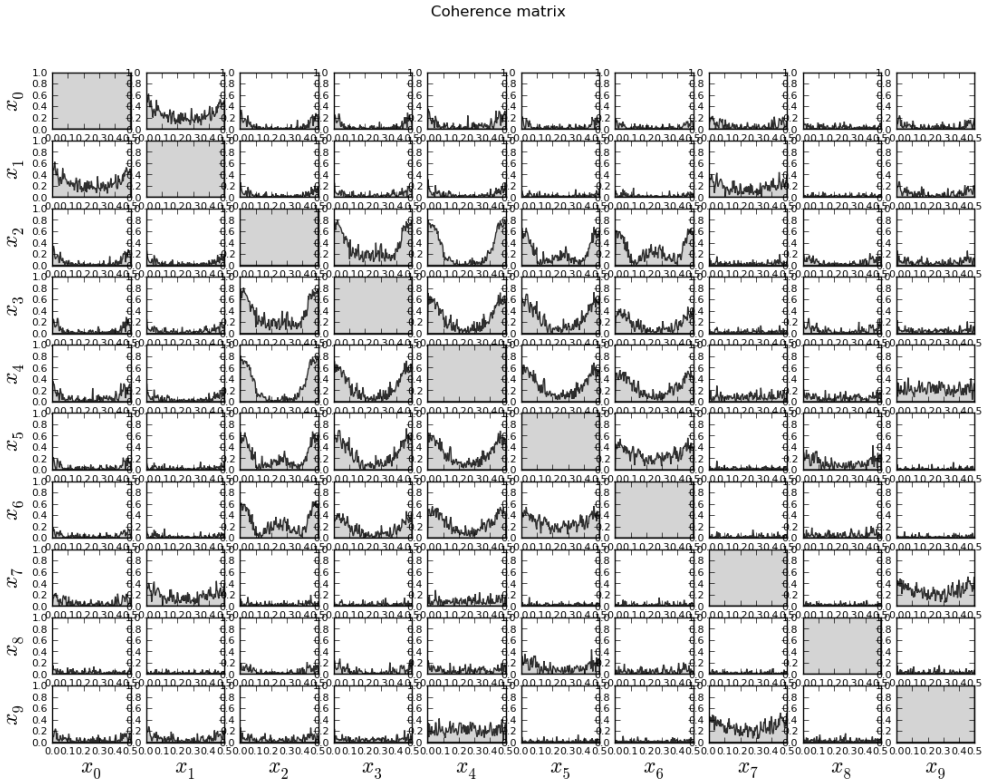

Figure 6.37: 10-node MVAR(2) Cross-coherence matrix from the network system in Figure 6.6. Notice the difference with the coherence matrix in Figure 6.33.

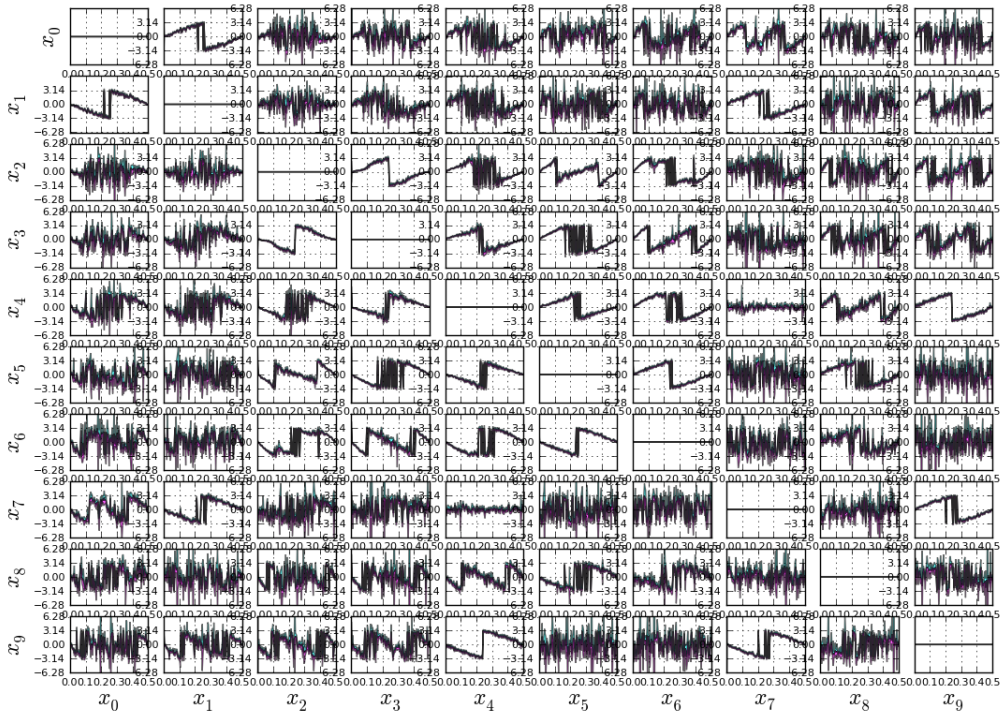

Figure 6.38: 10-node MVAR(2) cross-phase matrix from the network system in Figure 6.6.

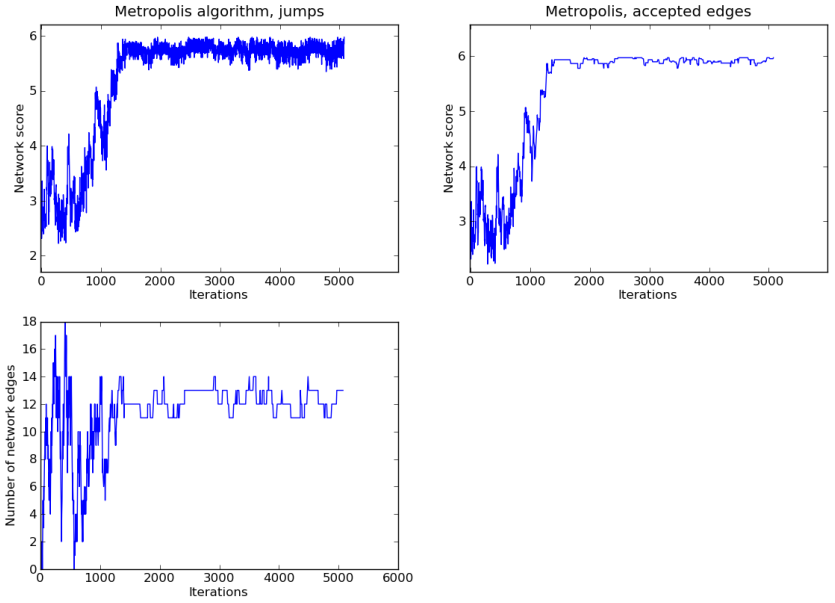

Figure 6.39: 10-node MVAR(2) simulated annealing performance curves. Top-left) Network score values $|\log(p(\mathbf{F}|S_h)p(\mathbf{\Theta}|S_h)p(S_h))|$ per algorithm iterations. Top-right) Network score values of the Metropolis accepted networks. Bottom) Number of edges of the networks accepted by simulated annealing.

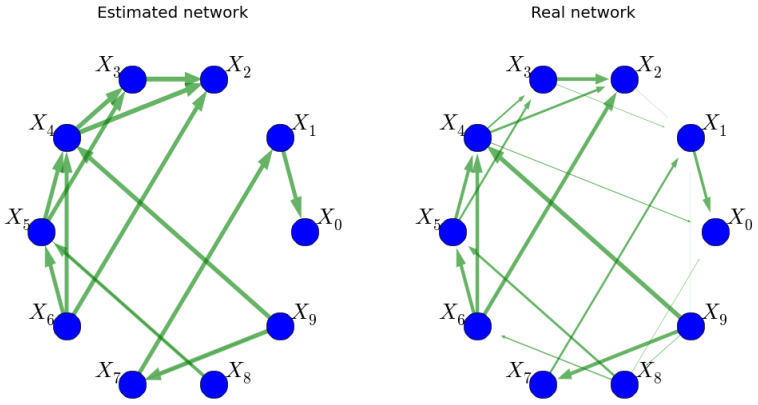

Figure 6.40: 10-node MVAR(2) estimated networks. Left) The estimated network using the FBN method. Right) True weighted MVAR(2) network. Here the network edge threshold was $Pnum = 13$ as in Figure 6.36.

The same experiments using the FBN method with the same hyperparameters were implemented to infer the nonlinear 10-node networks, NL(1) and NL(2). For the 10-node NL(1) network system, Figure 6.41 shows the coherence matrix and Figure 6.42 shows the cross phase among network nodes. Most of the coherence power is concentrated at high frequencies as can be seen in Figure 6.41.

Figure 6.43 shows the performance curves for the simulated annealing algorithm during the network sampling for the 10-node NL(1) system. Figure 6.43-top-left shows the network scores. The accepted network scores are shown at the top-right of the same figure. Figure 6.43-bottom shows the number of edges of the accepted networks which are approximately 16 edges. The estimated network is created by adding up the last 100 Metropolis accepted networks after the algorithm reached stability is shown in Figure 6.44-left. The original weighted 10-node network is shown at the right of the same figure. Notice again that the weakest edges were ignored by the FBN method.

The same set of experiments were performed for network NL(2). The results for a single run of the FBN algorithm using simulated annealing as the network distribution sampling method are also presented. As in the previous results, Figure 6.45 shows the coherence matrix for the 10-node NL(2) network, and Figure 6.46 shows the cross-phase matrix for the same system. Notice how the phase slopes in

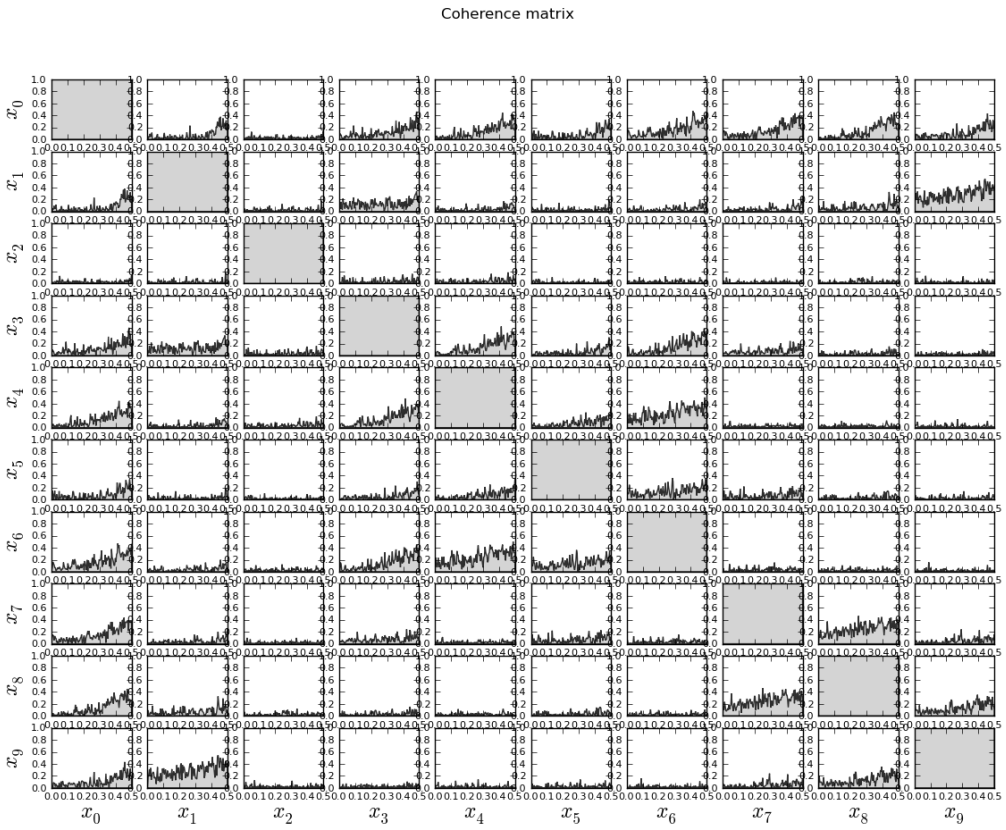

Figure 6.41: 10-node NL(1) network coherence matrix. The coherence matrix was computed from time series synthesized using the network system in Figure 6.14.

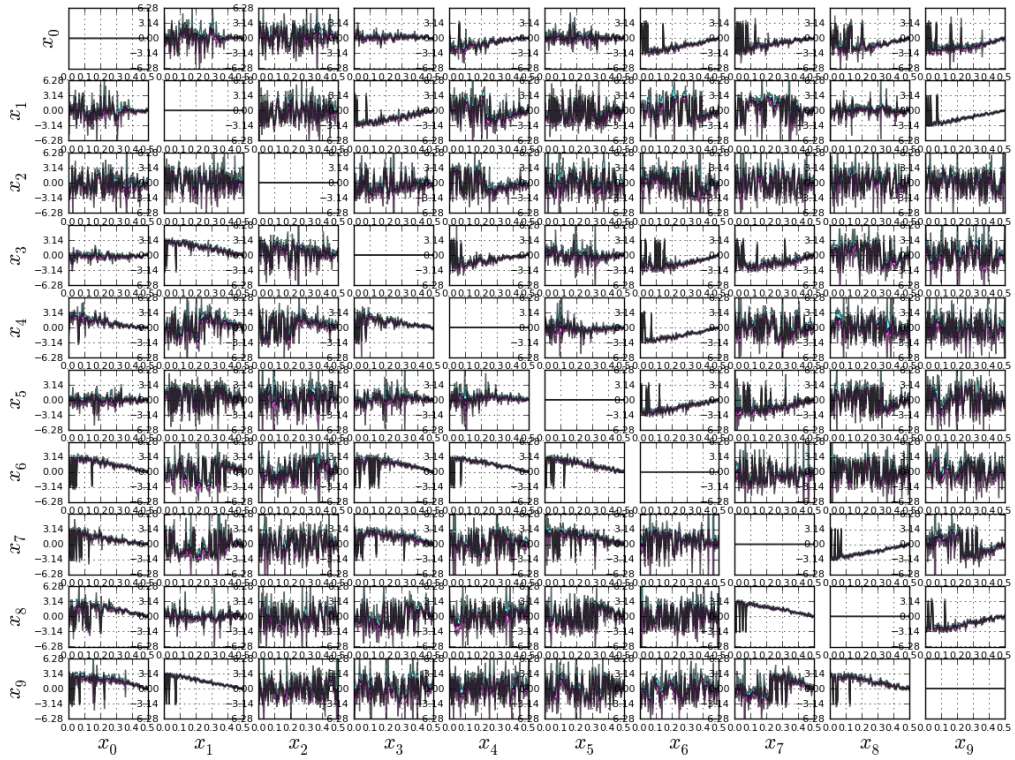

Figure 6.42: 10-node NL(1) network cross phase matrix, wrapped phase. The 10-node NL network is depicted in Figure 6.14.

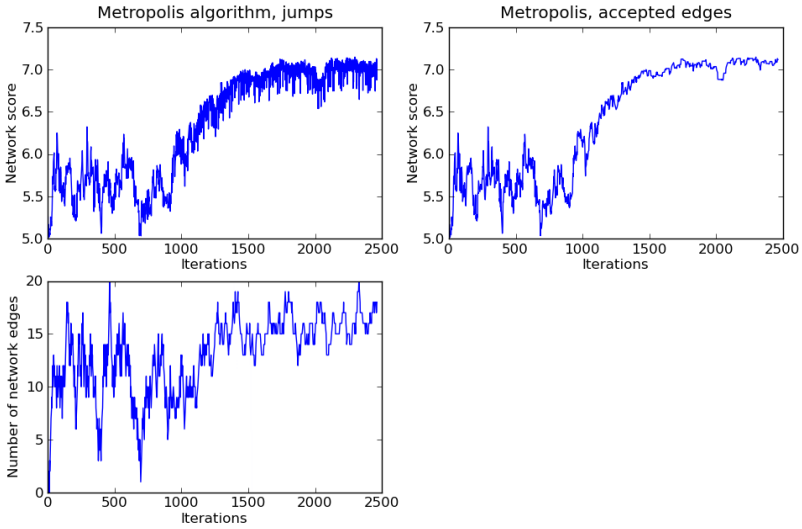

Figure 6.43: 10-node NL(1) simulated annealing performance curves. The top-left curve shows the network score value $|\log(p(\mathbf{F}|S_h)p(\mathbf{\Theta}|S_h)p(S_h))|$ for all proposed networks during the algorithm's run. Top-right curve shows the networks accepted by the simulated annealing. The number of edges of the accepted networks are shown in the bottom curve.

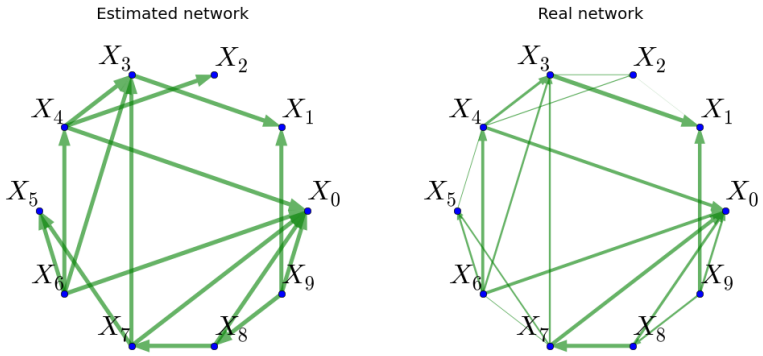

Figure 6.44: 10-node NL(1) estimated network. left) The estimated 10-node network using 100 sampled networks. right) True weighted nonlinear network system originally depicted in Figure 6.14. The threshold $Pnum = 16$ for the estimated structure.

Figure 6.46 are higher for the NL(2) when compared to the ones in Figure 6.42 for the NL(1) network, and recall that the slope is a indicator of causality. A negative slope indicates that the source represented by the matrix rows causes or influences the sources at the matrix columns.

As expected after observing the previous experiments, the performance curves for the simulated annealing algorithm were similar to the NL(1) and both MVAR network estimation cases. The performance curves for the 10-node NL(2) are shown in Figure 6.47. The network scores of all the proposed networks are shown at the top-left of this figure, while the top-right curve shows only the scores of the Metropolis accepted networks. The flat segments of this curve are iterations where the proposed networks were rejected by the Metropolis rule. The bottom curve in Figure 6.47 shows the number of edges of the accepted networks which is on average 15. Recall that the network structure is the same as the NL(1) case analysed previously.

Figure 6.48-left shows the estimated network using the FBN method. The original network is shown at the right of the same figure for comparison purposes. As in all previous cases the weak edges in the true weighted network in Figure 6.48 were ignored by the FBN method and only strong ones were obtained as the true network.

The four 10-node networks, MVAR(1), MVAR(2), NL(1) and NL(2), show that the FBN algorithm is able to infer the structure of these systems correctly, and more

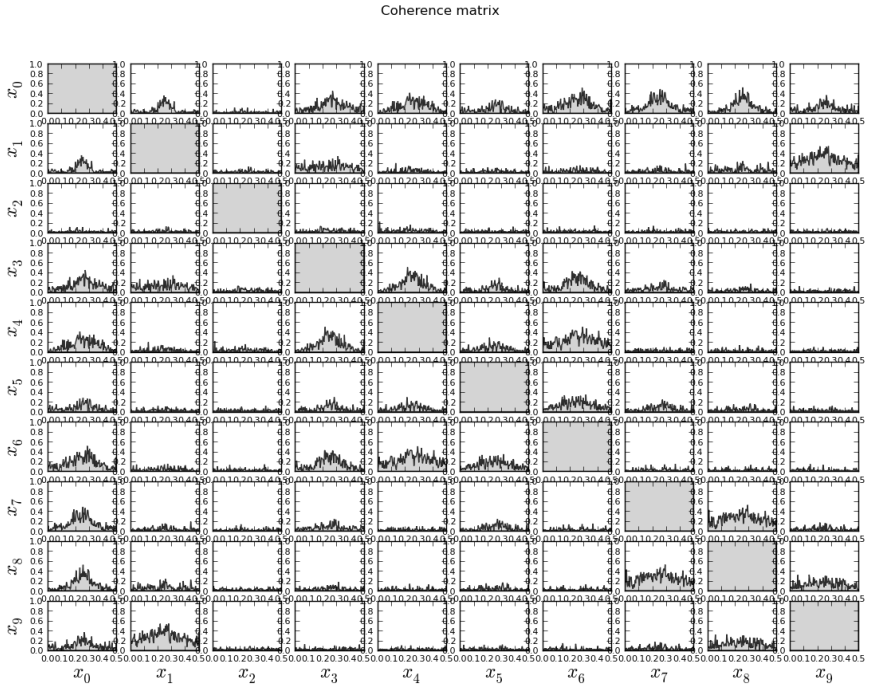

Figure 6.45: 10-node NL(2) coherence matrix. The coherence matrix was computed from time series synthesized by the network system shown in Figure 6.14.

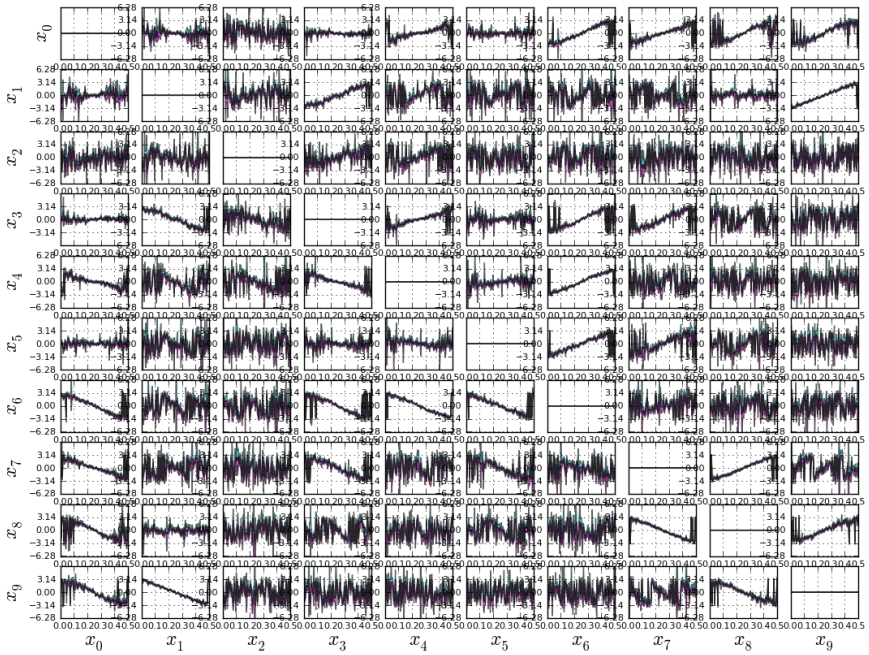

Figure 6.46: 10-node NL(2) cross phase matrix.

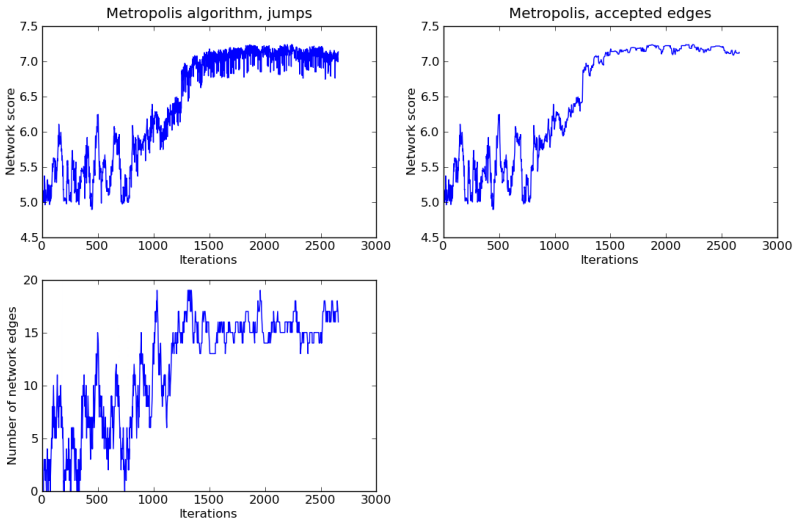

Figure 6.47: 10-node NL(2) simulated annealing performance curves. Top-left curve shows the network score values $|\log(p(\mathbf{F}|S_h)p(\boldsymbol{\Theta}|S_h)p(S_h))|$ for proposed networks before accepted by the Metropolis ratio. Top-right figure shows the accepted network scores. Bottom picture shows the number of edges for the accepted scores.

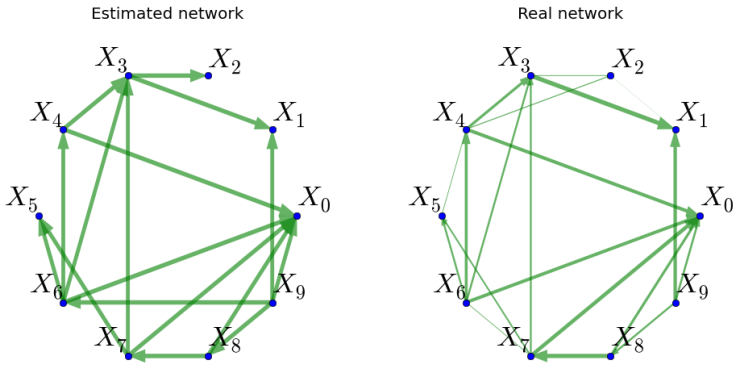

Figure 6.48: 10-node NL(2) estimated network. The estimated network is shown at the left of the figure. Right) The true weighted network for the 10-node NL system. Here the network edge threshold was $Pnum = 17$.

importantly that its performance is not affected by the system's time delay d. For the FBNs it is possible to infer NL(1) and NL(2) with the same successful result. This non-parametric property of FBNs is inherited by the Fourier transform.

The 10-node networks show that non-parametric property of FBNs, but their performance inferring large networks is still to be tested. In general, Bayesian networks is the preferred method to do this kind of task. Inference of large networks is the reason why BNs and DBNs have become so popular in fields like genetic networks or protein networks. The next section presents simulations of FBNs for network inference but now for the 100-node networks defined in Section 6.2.1 and Section 6.2.2.

6.5.1 FBNs: Inferring large networks

In this section the FBN method is used to infer the 100-node networks defined in Sections 6.2.1 and 6.2.2. In order to test the accuracy of the FBN algorithm inferring the network system, some matrices and performance measures must be defined first. Recalling the model equations for the linear and non-linear system in Eq. 6.2 and Eq. 6.3, the matrix that describes the causality and weights of the network system is defined by

$$\mathbf{C}_w = (\mathbf{U} - \mathbf{I}). * \mathbf{A} \ , \tag{6.5}$$

where $.*$ stands for element per element multiplication, $\mathbf{A}$ is the coefficient matrix in Eq. 6.2 and Eq. 6.3, $\mathbf{U}$ is a matrix of ones, and $\mathbf{I}$ is the unitary matrix. Notice that in $\mathbf{C}_w$ only the diagonal of $\mathbf{A}$ was zeroed out. The estimated matrix E_w is obtained by adding up 100 sampled directed binary networks from the network probability distribution using the simulation annealing algorithm, see Figure 6.32.

$\mathbf{E}_w$ and $\mathbf{C}_w$ define the connectivity of the estimated network and the real network system. Nevertheless comparison of both matrices can not be done directly due to the difference in their number of network edges and weights. For this reason it is necessary to threshold both matrices in order match the average degree of the networks; the real and the estimated networks. We define this threshold as $Pnum$ which represents the number of accepted edges in the inferred network.

Using $Pnum$ it is possible define binary directed matrices for the real and estimated networks $\mathbf{C}_{bd} = \lfloor \mathbf{C}_w \rfloor_{Pnum}$ and $\mathbf{E}_{bd} = \lfloor \mathbf{E}_{bd} \rfloor_{Pnum}$, where the floor brackets indicates thresholding and $Pnum$ is the number of edges for the threshold. The binary undirected versions of these matrices can be created easily as

$$\mathbf{E}_{bu} = \mathbf{E}_{bd} + \mathbf{E}_{bd}^T \ , \tag{6.6}$$

and similarly for $\mathbf{C}_{bu}$. Using these matrices the network inference performance measures are defined. As said previously, these measures test the accuracy of the FBN algorithm and they are:

Structural correct edges(SC): SC measures the number of edges that were found correctly regardless of causality. SC focuses on the network structure using the undirected edges and it is defined mathematically as

$$SC = \frac{1}{2} \sum_i \sum_j E_{bu}(i,j) C_{bu}(i,j) \tag{6.7}$$

where $E_{bu}(i,j)$ and $C_{bu}(i,j)$ are the (i,j) elements of matrices $\mathbf{C}_{bu}$ and $\mathbf{E}_{bu}$ respectively. The weighted version of WSC, which uses the original weights of the connectivity matrix $\mathbf{C}_w$, is defined as

$$WSC = \sum_i \sum_j E_{bu}(i,j) C_{bu}(i,j) C_w(i,j) \ . \tag{6.8}$$

Correct directed edges (CD): From the total structural similarities (SC) in both networks, CD will measure the number of edges whose inferred causality match to the ones in the real network. This measure is defined by

$$CD = \sum_i \sum_j E_{bd}(i,j) C_{bd}(i,j) \ , \tag{6.9}$$

and its weighted version WCD is defined by

$$WCD = \sum_i \sum_j E_{bd}(i,j) C_{bd}(i,j) C_w(i,j) \ . \tag{6.10}$$

Wrong directed edges (WD): From the structural similarities (SC) in both networks, WD measures the number of edges whose causality is contrary to the ones in the real network. WD is defined by

$$WD = \sum_i \sum_j E_{bd}(j,i) C_{bd}(i,j) \ . \tag{6.11}$$

Its weighted version (WWD) is defined by

$$WWD = \sum_i \sum_j E_{bd}(j,i) C_{bd}(i,j) C_w(i,j) \ . \tag{6.12}$$

Structural errors (SE): SE measures all the edge differences between both networks either it is an extra inferred edge or a missing edge. This measure is defined by

$$SE = \frac{1}{2} \sum_i \sum_j \left| E_{bu}(i,j) - C_{bu}(i,j) \right| . \tag{6.13}$$

The weighted version WSE can only be defined for the missing edges, edges that are present in the real network but not in the inferred one since for the extra inferred edges there are not weights in $\mathbf{C}_w$. Taking into account this previous fact, WSE is defined as

$$WSE = \sum_i \sum_j \left| E_{bu}(i,j) - C_{bu}(i,j) \right| C_w(i,j) \ . \tag{6.14}$$

The four defined performance measures and their weighted versions are used to test the performance of the FBN algorithm for the remaining of this chapter.

Table 6.2 shows the hyperparameters used to estimate the 100-MVAR(1) network. Notice that the large number of nodes in this experiment requires more algorithm iterations in order to reach stability. For this experiment a total of 200000 iterations were used with a cooling time of 70000.

Table 6.2: Simulated annealing hyperparameters for the 100-node MVAR(1) system experiments.

numruns:	200000		initemp:	2.0
coolruns:	70000		fintemp:	0.06
burnedruns:	8000		alpha:	0.02
			beta:	0.04

The large number of nodes makes it impossible to show figures with coherence and cross-phase matrices. In their place the estimated connectivity matrix is given. Figure 6.49 shows the estimated binary directed matrix $\mathbf{E}_{bd}$ and the thresholded original matrix $\mathbf{C}_{bd}$ for the 100-node MVAR(1) network at the left and right of the same figure respectively. Notice that some of the inferred edges have contrary direction. This can be noticed by the elements at the lower triangular part of the

estimated connectivity matrix which should be full of zeros as it is for the true network.

For the connectivity matrices in Figure 6.49 the average number of edges of the sampled networks obtained from the simulated annealing algorithm was of 133. Hence, the original network matrix C_w was thresholded until the 133 strongest directed edges remained while the rest were zeroed out and similarly for $\mathbf{E}_w$. Here it is important to mention that even though the networks sampled by the simulated annealing algorithm are binary directed and acyclic matrices, there is no guarantee that the network E_{bd} will be acyclic since it is created by adding up sampled networks, as shown in Figure 6.32.

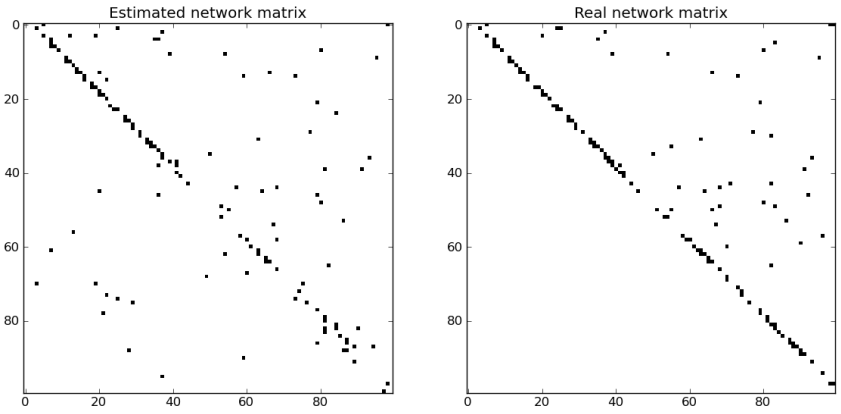

Figure 6.49: 100-node MVAR(1) estimated connectivity matrix shown at the left of the figure and the original connectivity is shown at the right side. Notice that some edges were estimated with backward causality which can be seen in the lower triangular part of the estimated matrix.

The simulated annealing performance curves are shown in Figure 6.51. Notice that the average number of edges of the sampled network is around the 133 edges, where the algorithm reached stability.

Table 6.3 shows the summarized results for this experiments. The true network system represented by $\mathbf{C}_w$ has 200 weighted edges. The estimated network E_{db} has 133, 67 less edges than the original system. Recall the 10-node network experiments in the previous section where the FBN method tends to ignore all low weighted edges and keeps only the stronger ones, hence an estimated network with fewer edges was also expected here. From the 133 estimated edges 99 undirected edges are estimated

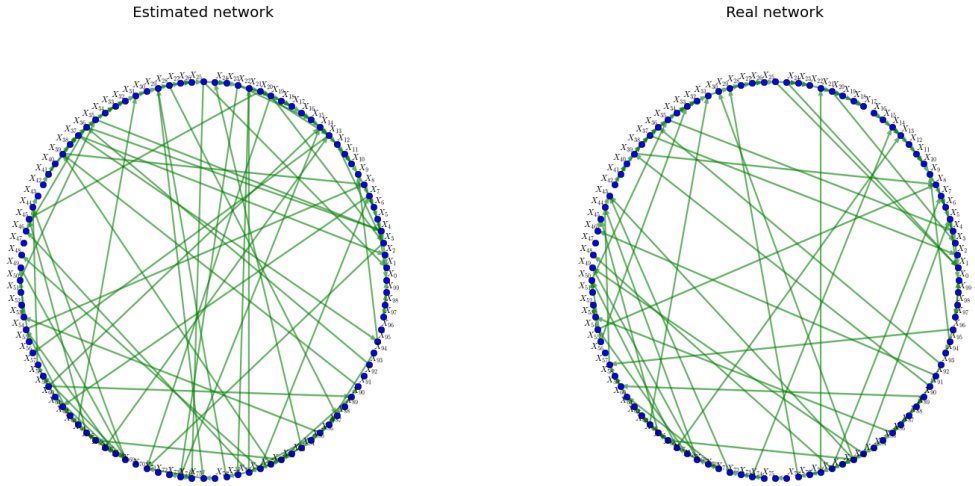

Figure 6.50: 100-node MVAR(1) estimated network using the FBN algorithm. The estimated network is shown at the left and a network edge threshold $Pnum = 133$ edges was used. The true network is shown at the right side of the figure.

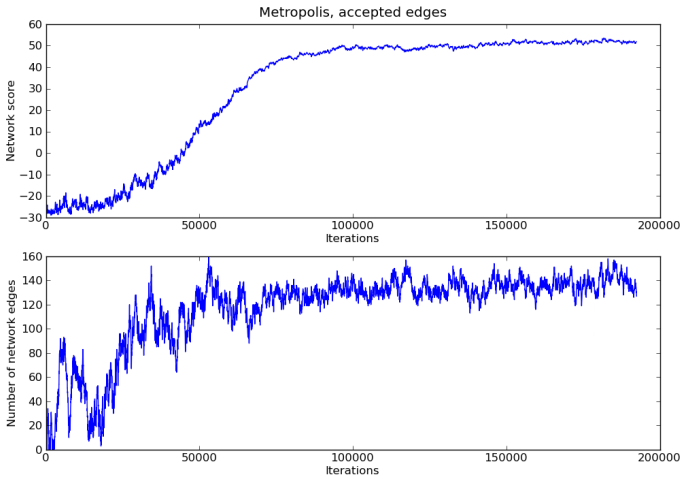

Figure 6.51: 100-node MVAR(1) simulated annealing performance curves. The top curve shows the score values $|\log(p(\mathbf{F}|S_h)p(\mathbf{\Theta}|S_h)p(S_h))|$ of the accepted network vs. number of iteration of the algorithm. The bottom curve shows the number of edges in the accepted networks.

correctly, meaning that the structure of the network was inferred correctly in 99 of the 133 edges, ignoring causality. From these 99 correct undirected edges, 89 were inferred with the correct direction and the remaining 10 edges with contrary direction as can be seen in WD. Finally, 34 edges were estimated incorrectly either they were missed by the algorithm or were spurious found edges.

Since the network structure inference favours strong connections, it is also important to analyse the weighted version of the performance measures. Using the edge weights in matrix $\mathbf{C}_w$ the weighted version of the performance measures are computed. The total network weight is of 80.8238 which was computed by summing 133 weights in $\mathbf{C}_w$ that match the edges in $\mathbf{C}_{bd}$. WSC is 65.1132, meaning that 80.56% of the original structure was inferred correctly, and WCD also shows that 72.41% of the original edges were inferred with the correct direction.

Table 6.3: Single run results: 100-node MVAR(1) network.

Edges in $\mathbf{E}$ (Pnum):	133	Edges in $\lfloor \mathbf{C}_{bd} \rfloor$:	133	
Number of edges in $\mathbf{C}_w$	200	Real network weight:	80.8238	
SC:	99	WSC	65.1132	80.56%
CD:	89	WCD:	58.5304	72.41%
WD:	10	WWD:	6.5828	8.14%
SE:	34	WSE:	15.7105	19.43%

The same network inference experiment was also run for the MVAR(2) network. The hyperparameters used are shown in Table 6.4 and the estimated network in Figure 6.52. The structure of the original system is exactly the same as for the MVAR(1) system, the only thing that has changed is the model order of the MVAR equations. At first glance both networks in Figure 6.52 look very similar.

Table 6.4: Simulated annealing hyperparameters for the 100-node network MVAR(2) system experiments.

numruns:	150000		initemp:	2.0
coolruns:	50000		fintemp:	0.04
burnedruns:	8000		alpha:	0.02
			beta:	0.04

Figure 6.53 shows the weighted connectivity matrices for both networks, the estimated and true network. As it happened in the MVAR(1) case some of the estimated edges have inverse causality, the direction of the edges was estimated

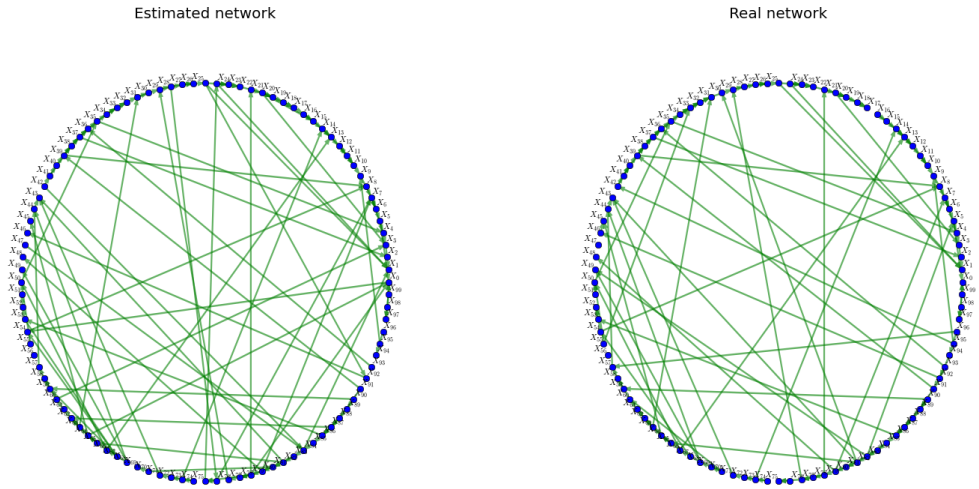

Figure 6.52: 100-node MVAR(2) estimated network using the FBN algorithm. The estimated network is shown at the left side of the figure and the true network system is at the right side. The estimated network edges were thresholded using a $Pnum = 141$.

backwards. This can be seen easily by inspecting the matrix elements at the lower triangular part of the connectivity matrix for the estimated network, which as shown in Figure 6.53-right should be full of zeros.

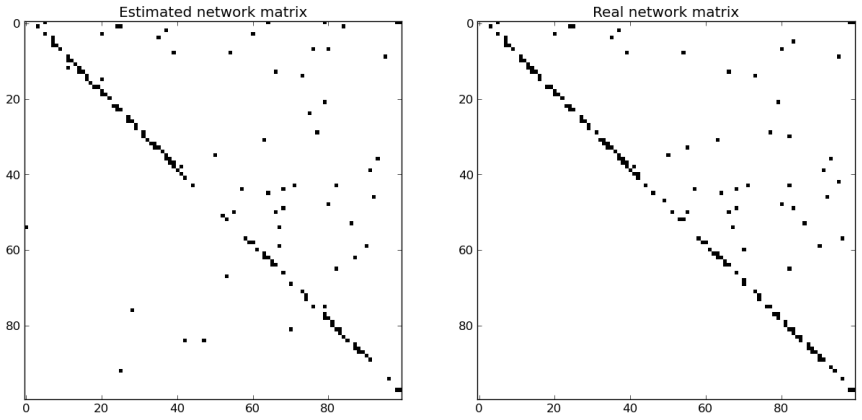

Figure 6.53: 100-node MVAR(2) estimated connectivity matrix using the FBN method. The elements at the lower triangular part of the estimated connectivity matrix indicates edges with backwards causality.

The simulated annealing performance curves for this experiment are shown in Figure 6.54 where the number of edges of the sampled networks is around 141, being the edge number average of the last 100 networks from which matrix $\mathbf{E}_w$

is computed. From the 141 edges, 119 were estimated correctly, representing this number the 90.04% of the original structure and all of them were inferred with correct causality. Only 22 edges were missing or spurious, representing the first ones the 9.95% of the original network structure.

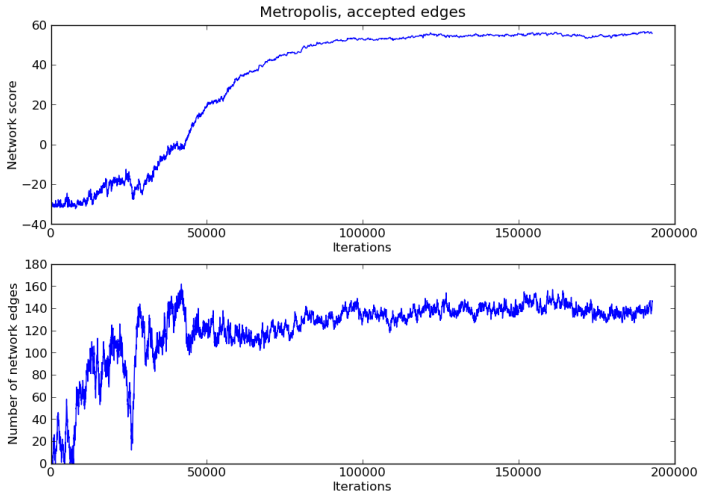

Figure 6.54: 100-node MVAR(2) simulated annealing performance curves. The top curve shows the score values $|\log(p(\mathbf{F}|S_h)p(\mathbf{\Theta}|S_h)p(S_h))|$ of the accepted networks during the simulated annealing algorithm. The bottom curve shows the number of edges of the accepted networks versus the number of iterations for the simulated annealing algorithm.

Table 6.5: Single run results: 100-node MVAR(2) network estimation using FBNs.

Edges in $\mathbf{E}_{bd}$:	141	Edges in $\lfloor \mathbf{C}_{bd} \rfloor$:	141	
Number of edges in $\mathbf{C}_w$	200	Real network weight:	83.0968	
SC:	119	WSC:	74.8225	90.04%
CD:	119	WCD::	74.8225	90.04%
WD:	0	WWD:	0.0	0.0%
SE:	22	WSE:	8.2743	9.95%

Nonlinear large networks

Experiments are also run for the non-linear cases NL(1) and NL(2). For this set of experiments the hyperparameters used are shown in Table 6.6. Figure 6.55 shows the performance curves for the NL(1) network inference. In this case the average number of edges in the sampled network was of 108 edges.

Figure 6.56 shows the inferred (left) and real (right) connectivity matrices for the studied network after thresholding. Notice that as in previous experiments

Table 6.6: Simulated annealing hyperparameters for 100-node network NL(1) experiments.

numruns:	200000		initemp:	2.0
coolruns:	70000		fintemp:	0.015
burnedruns:	8000		alpha:	0.01
			beta:	0.04

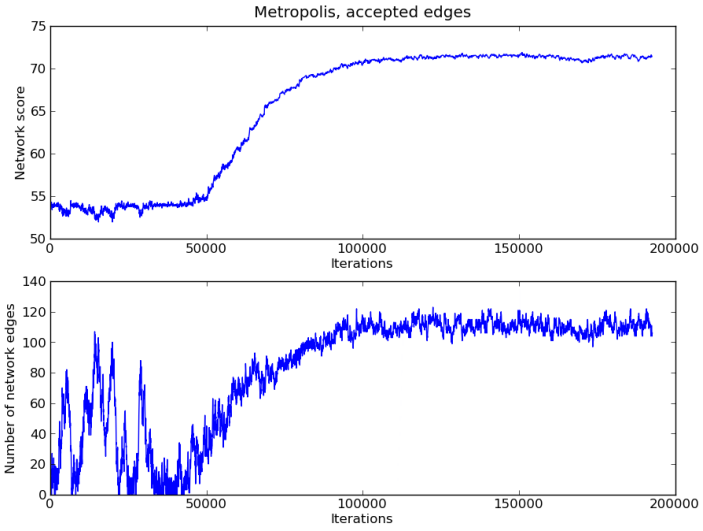

Figure 6.55: 100-node NL(1) simulated annealing performance curves. As in the previous cases, the top curve shows the score $|\log(p(\mathbf{F}|S_h)p(\mathbf{\Theta}|S_h)p(S_h))|$ of accepted networks which were sampled by the simulated annealing algorithm. The bottom curve shows the number of edges of the accepted networks versus algorithm iterations.

some edges in the inferred network were found with reverse causality. These can be seen in the lower triangular part of the estimated connectivity matrix. The same information is shown in circular network layout in Figure 6.57 where the similarities between the two networks can be noticed better.

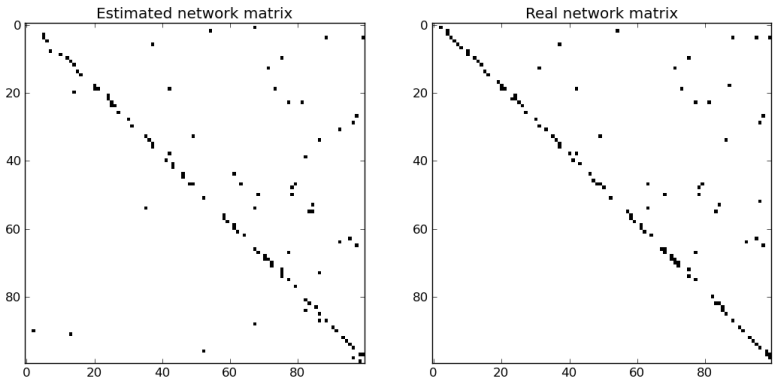

Figure 6.56: 100-node NL(1) estimated (left) and true (right) connectivity matrices. The elements in the lower triangular part of the estimated connectivity matrix indicate estimated edges whose causality are backwards.

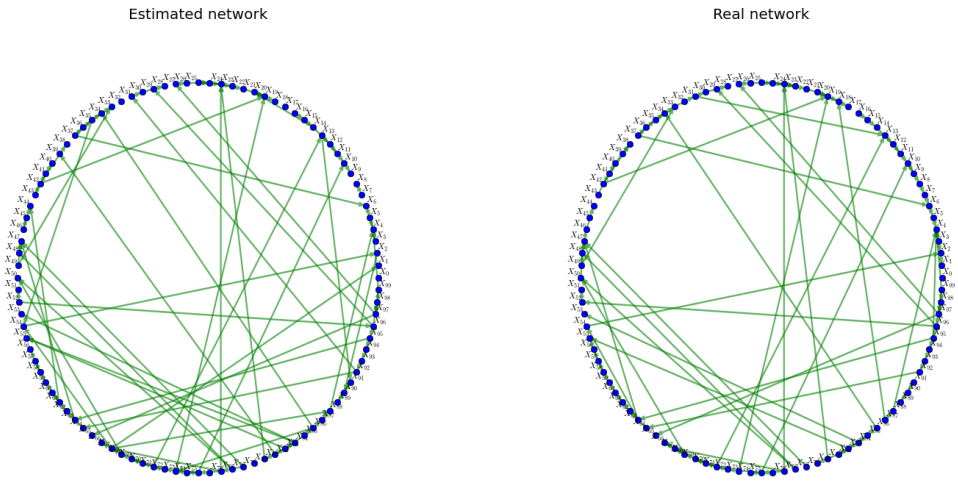

Figure 6.57: 100-node NL(1) estimated (left) and true (right) networks in a circular layout. The network edge threshold for the estimated network was $Pnum = 108$. The true network in $\mathbf{C}_w$ has a total of 117 weighted edges.

The performance measures are shown in Table 6.7. The average number of edges in the sampled networks was 108 and this number was chosen as the edge threshold for network comparison. In this case 84.84% of the network structure was

inferred correctly ignoring of course the causality of the edges. If causality is taken into account, then 80.27% of the real structure was inferred correctly. The network structural errors in this experiment represents the 15.35% of the original thresholded structure.

Table 6.7: Single run results: 100-node NL(1) network estimation using FBNs.

Edges in $\mathbf{E}_{bd}$:	108	Edges in $\lfloor \mathbf{C}_{bd} \rfloor$:	108	
Number of edges in $\mathbf{C}_w$	117	Real network weight:	72.1025	
SC:	88	WSC:	61.0305	84.64%
CD:	83	WCD:	57.8833	80.27%
WD:	5	WWD:	3.1471	4.364%
SE:	20	WSE:	11.0719	15.35%

For the 100-node NL(2) network experiment, the same hyperparameters given in Table 6.6 were used in the simulated annealing algorithm. As in the previous cases, performance curves are shown where the average number of edges in the sampled network was of 117 edges. The inferred connectivity matrix is shown in Figure 6.59-left where the edges with contrary causality appear in the lower triangular part of this matrix. These matrices are also shown in circular network layouts in Figure 6.60 where similarities can be appreciated but also some long range spurious edges in the inferred network.

Performance results are given in Table 6.8. The structure was recovered with 85.51% accuracy and all edge directions were also found correctly. Only the 14.48% of the real thresholded structure were found with errors.

Table 6.8: Single run results: 100-node NL(2) network.

Edges in $\mathbf{E}_{bd}$:	117	Edges in $\lfloor \mathbf{C}_{bd} \rfloor$:	117	
Number of edges in $\mathbf{C}_w$	117	Real network weight:	75.99	
SC:	96	WSC:	64.9832	85.51%
CD:	96	WCD:	64.9832	85.51%
WD:	0	WWD:	0.0	0.0 %
SE:	21	WSE:	11.0087	14.48%

In this section FBNs for network structure inference were tested using practically eight network systems. These systems were created by changing the model (MVAR and NL), time delay d (1 and 2), and their network sizes (10 and 100 nodes). Due to the difficulty of choosing a fair network using these models, the network edge weights and the network structures were chosen randomly. The first set of experiments using

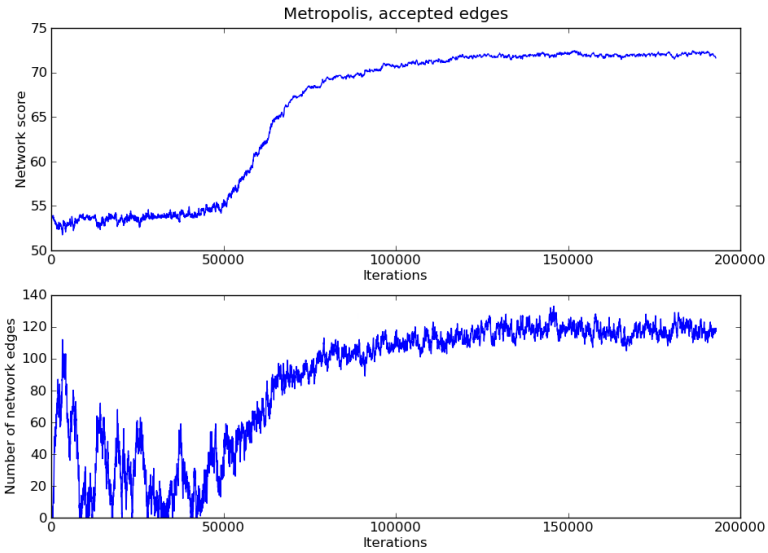

Figure 6.58: 100-node NL(2) simulated annealing performance curves. The top curve shows the score values $|\log(p(\mathbf{F}|S_h)p(\mathbf{\Theta}|S_h)p(S_h))|$ of the accepted sampled networks. The bottom curve shows the number of edges of the accepted sampled networks.

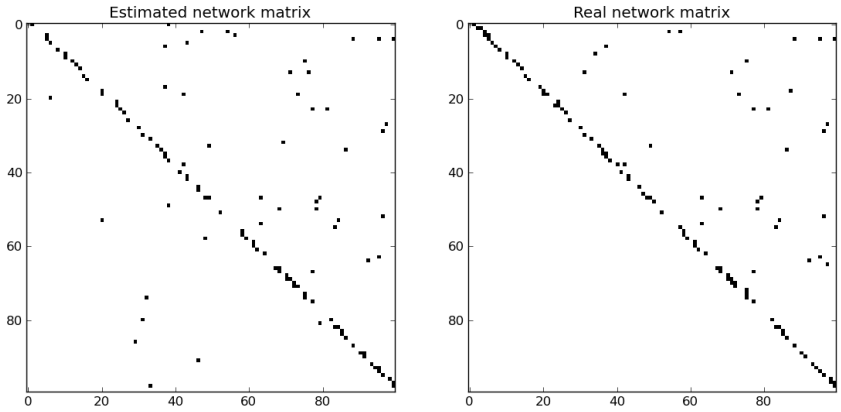

Figure 6.59: 100-node NL(2) estimated (left) and true (right) connectivity matrices.

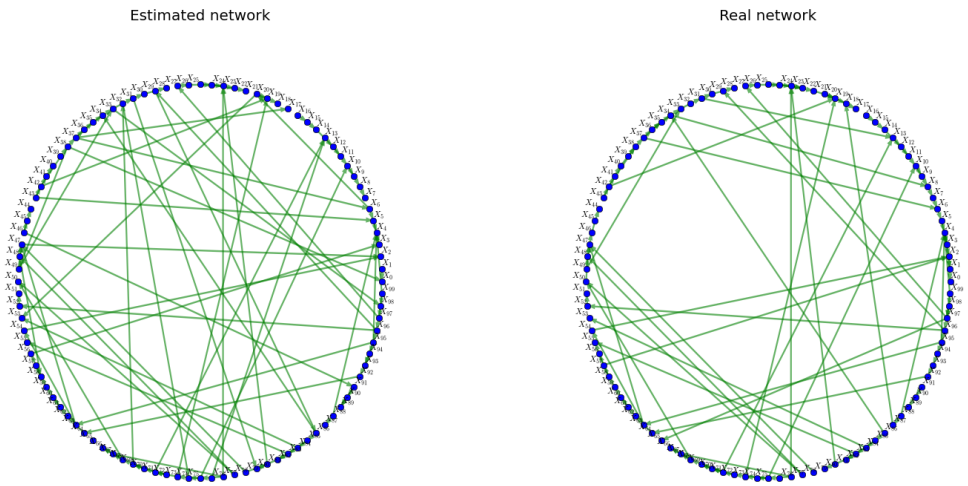

Figure 6.60: 100-node NL(2) estimated (left) and real (right) networks in a circular layout. The network edge threshold was $Pnum = 117$. The true matrix has a total of 117 weighted edges.

the 10-node size networks (MVAR and NL) were implemented with the intention of showing the algorithm's performance in a network that can be eye inspected. All these experiments were successful for the FBN method. The second set using large networks were implemented in order to see how the method behaves when inferring large structures. Although 100 is not strictly large it is large enough for the application in this thesis. Again, FBNs behaved relatively well inferring these network systems for the linear and non-linear cases.

A pending issue is to compare FBNs with the DBN algorithm for network structure inference. This is the focus of the following section.

6.5.2 Comparing FBNs and DBNs

In the previous sections, the behaviour and performance of the FBN method was shown for two different networks with different parameters: equation model, time delay, and network size. This section shows performance comparison of the proposed method in this thesis, FBNs, with the well known DBNs for structure inference explained in Section 5.3. DBNs work in the time domain and causality is inferred by network unfolding. This means that a second delayed set of time series is reproduced and in consequence the number of nodes is doubled. A DBN can be unfolded several

times depending on the needs of the analysed system, although this is something that is not normally known in advance. However, for many of the applications in the current literature one network unfolding might be sufficient for DBNs.

The one delay unfolding is very practical in brain network inference using fMRI data and DBNs have given successful results in this field. Nevertheless, fMRI is a brain imaging technique that although it has an unparalleled spatial resolution it suffers from poor time resolution, in the order of seconds. Network unfolding using one time delay for this brain imaging technique is not a matter of choice but a need that results from the low sampling rate and short time series length.

In Electrophysiology the sampling rate can be high, 500Hz or more, and the delayed interactions are variable among the neuronal groups. Hence, if DBNs are applied in Electrophysiology, several unfoldings should be performed or a different representation of the data sets should be used to feed into the DBN algorithm, such as the time series envelope as in Smith et al. (2006).

In this experiment the advantage of FBNs over DBNs when the model order of the analysed system is unknown is demonstrated. Experiments for MVAR(1) and MVAR(2) networks are shown only. Suppose then, that the model order of the system whose time series were recorded is unknown and in consequence a DBN algorithm using one time delay (one network unfolding) is implemented as a first option.

Using the previously defined 10-node networks for the MVAR model, time series are generated for each of the network nodes. Then, the FBN and DBN algorithms are implemented to infer the network structures. This structure will be the best network that explains the available datasets.

Figure 6.61-left shows the inferred 10-node MVAR(1) network using FBNs similarly to the previous experiments shown in Section 6.5, but here a $Pnum = 15$ was chosen as network degree threshold, this means that only the 15 most sampled edges were kept for the final estimated network. By simple observation we can see that the structure is well recovered.

Figure 6.62 shows the same experiment as in Figure 6.61 but here a standard DBN algorithm is implemented. Notice that a similar result was obtained. DBNs

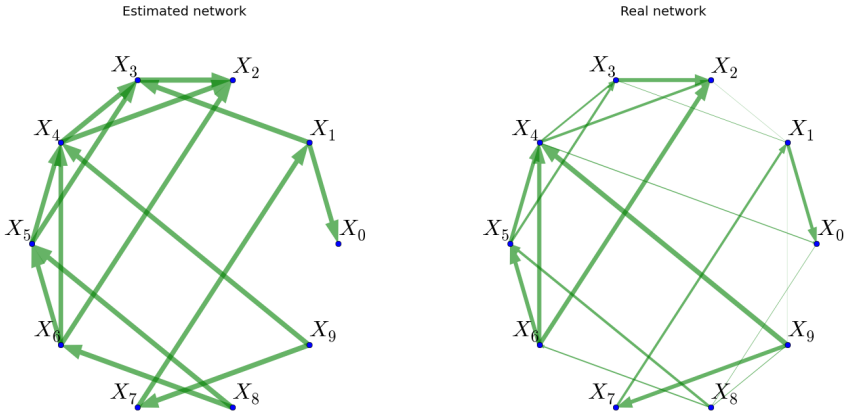

Figure 6.61: 10-node MVAR(1) estimated (left) and true (right) networks. The network was estimated using the FBN method and the network edge threshold was $Pnum = 15$.

were able to infer correctly the structure, but recall here that the delay of the system is 1, MVAR(1), which is exactly the expected by the DBNs which use one time delay for the network unfolding.

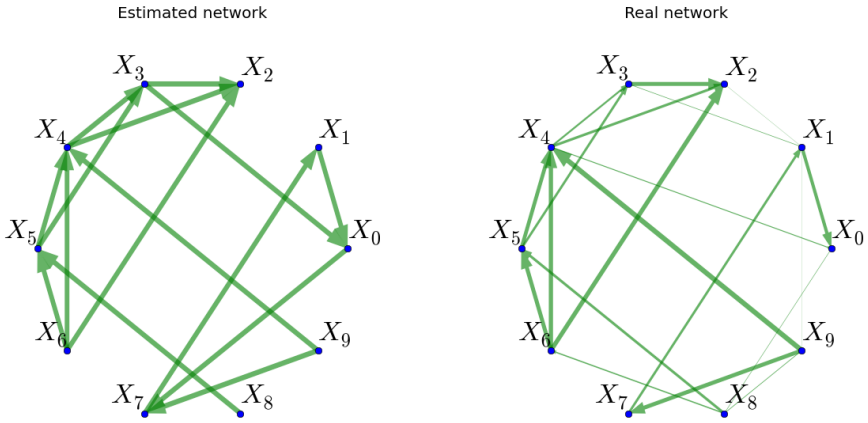

Figure 6.62: 10-node MVAR(1) estimated (left) and true (right) networks. The network was estimated using DBNs and the network edge threshold was $Pnum = 15$.

The results for the MVAR(2) are shown in Figure 6.63 for the the FBN method. Here we can see by visual inspection that the proposed method was able to infer most of the real structure. On the other hand, Figure 6.64-left shows the inferred network using the standard DBNs and it can be seen that it is completely different from the original structure shown at the right side of the same figure. The poor performance of the DBNs are due to the change of d, which does not fit the delay

from which the DBN method was designed.

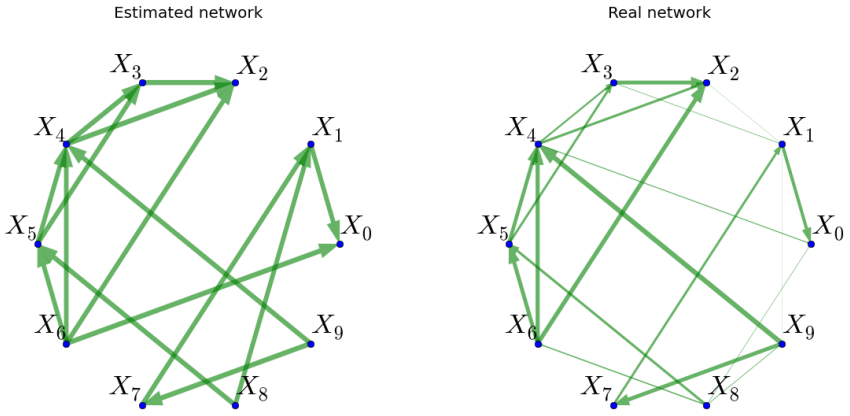

Figure 6.63: 10-node MVAR(2) estimated (left) and true (right) networks. The network was estimated using the FBN algorithm proposed in this thesis and the network edge threshold $Pnum = 15$ was used.

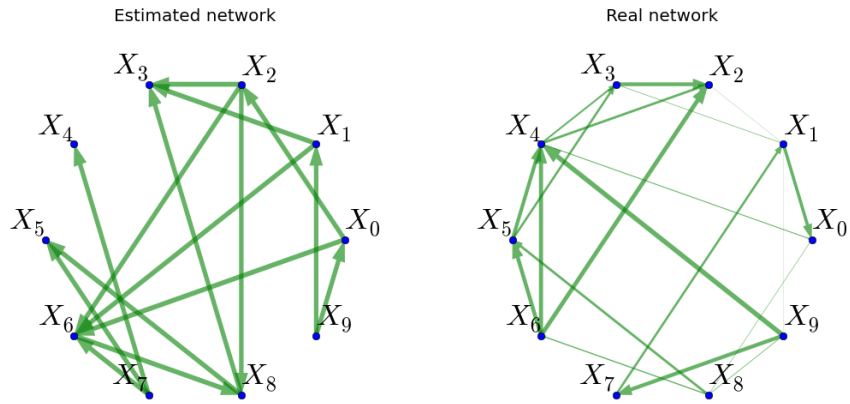

Figure 6.64: 10-node MVAR(2) estimated (left) and true (right) networks in a circular layout. The Figure shows the advantage of FBNs over DBNs. FBNs are non-parametric, meaning that this method does not rely on the delay variable d. On the other hand DBNs relies heavily on this parameter and when it is changed it produces wrong networks as shown in this figure.

In order to have a broader perspective of DBNs and FBNs performances, 500 runs of the previous experiments were implemented. With these network results the eight defined performance measures in Section 6.5.1 were computed. Figure 6.65 shows the structural correct edge (SC) and the weighted SC (WSC) measures in box plots for the FBN and DBN methods. Recall that the box's boundaries are defined by the lower quartile and the upper quartile, the central line is the median

of the population which is also the second quartile. Also the whiskers indicate the population range and the "+" markers are samples that are considered outliers.

The SC measure estimates how good the algorithm was in extracting the original structure regardless of the causality of the inferred edges. From Figure 6.65 we can see that FBNs had a good performance inferring the structure. DBNs also had a good performance for the structure inference for the MVAR(1) case. This did not occur for the MVAR(2) where the parameter d of the network equations was changed. These results show that DBNs are highly affected by the delay of the model and that FBNs are robust against this parameter. The weighted version WSC shows similar results as can be observed in Figure 6.65-right.

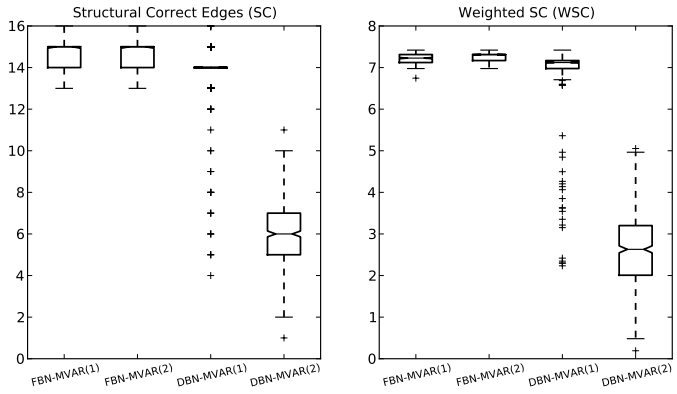

Figure 6.65: Structural correct (SC) and weighted SC (WSC) results for the 500 experiments. The box limits define the lower quartile and the upper quartile. The thickest line in each box is the median of the population for each case.

Figure 6.66 shows the correct directed edge (CD) measure and its weighted version (WCD). CD is similar to SC but here the direction of the edges matters. If an edge has a contrary direction it is an error. The panel at the right shows box plots for SC, which counts for the directed binary inferred networks. For the FBN-MVAR(1) the average of correct directed edges found was 14 with the lower quartile in 13. In general both FBN and the DBN-MVAR(1) had a similar performance. The worst behaviour was shown by DBN-MVAR(2) which found around 3 edges with the correct direction for the inferred networks.

Figure 6.67 shows the wrong directed (WD) edge measures. From the structural correct edges measured by SC, WD counts those ones whose direction was inferred

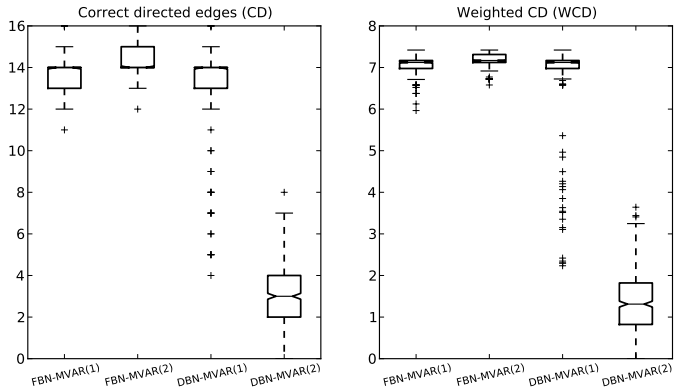

Figure 6.66: Correct directed (CD) and Weighted CD (WCD) results for the 500 experiments shown in box-plots. Here the best performance was achieved by the FBN-MVAR(2) case.

incorrectly. The median for the FBN cases is 0 which is a good result although a better one was obtained by DBN-MVAR(1). The worst case again is DBN-MVAR(2) where the range for these results reaches 7 edges. WWD also shows similar results to the binary ones.

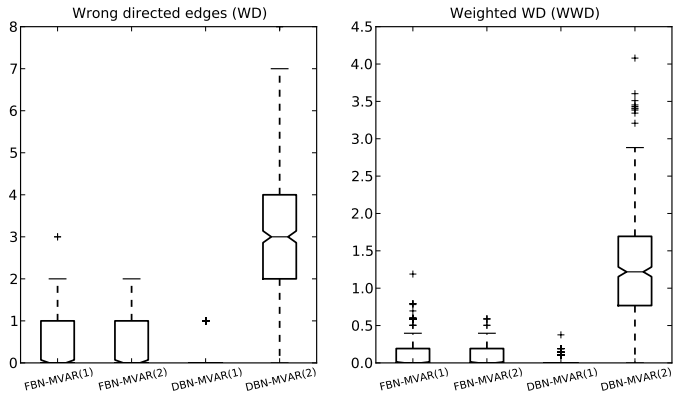

Figure 6.67: Wrong directed (WD) and weighted (WD) WWD results for the 500 experiments. Best performance was achieved by DBN-MVAR(1).

Structural error (SE) counts the number of edges in the inferred network that do not match the real network (here their causality is not important by definition). The best results were obtained by FBNs for both delays $d = 1$ and $d = 2$. The worst result was obtained by DBN-MVAR(2). Similar results are shown for the weighted version of this measure, meaning that the error edges or extra edges inferred by the FBN method do not have a high weight in the network.

This section showed through a series of experiments the performance of the

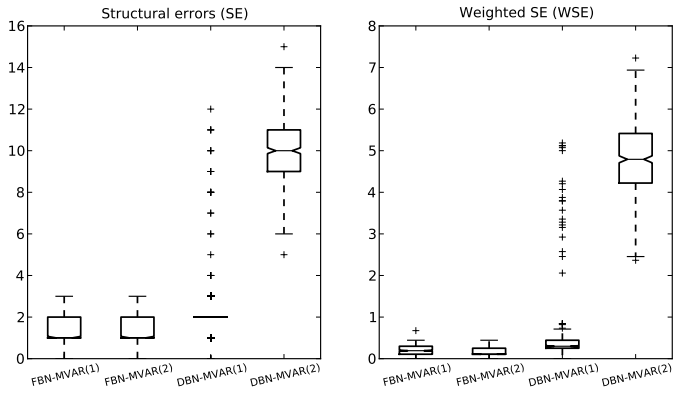

Figure 6.68: Structural error (SE) and weighted SE (WSE) results for the 500 experiments.

proposed method FBNs for network structure estimation. The FBN algorithm was tested using eight different network systems defined by size (10 and 100 nodes), model (MVAR and NL), and time delay (1 and 2). In all experiments, FBNs showed similar results to DBNs with exception of all $d = 2$ experiments. The performance of DBNs when $d = 2$ is dramatically decreased because the DBN was designed to search one time delay interactions in the network system. Recall that only one network unfolding was implemented with one time delay. This problem can be fixed by either delaying twice the one unfolded network or unfolding the DBN network twice. This second solution will increase by three the number of sources which also increases the complexity of the original problem. On the other hand, FBNs present the advantage of being non-parametric, a property inherited by the Fourier transform and proved by the results presented here. FBNs' performance is not heavily affected by changes in d.

6.5.3 FBNs performance at defined frequency bands

All simulations in previous sections were implemented using the entire frequency spectrum, normalized from 0.0 Hz to 0.5 Hz. But the spectral nature of FBNs also allows to study smaller sets of frequency bands for the same system using the same data sets by defining a range of frequencies for λ. In this section the experiments of the previous section (500 algorithm iterations) are repeated for the 10-node MVAR(1) network estimation using different ranges for λ. These are: $\lambda_1 =$

0.0Hz−0.1Hz, λ_2 = 0.0Hz−0.2Hz, λ_3 = 0.1Hz−0.2Hz, and λ_4 = 0.0Hz−0.5Hz is repeated again for easy comparison.

Figure 6.69 shows the SC and the WSC measures for the inference of the 10-node MVAR(1) network using FBNs in the defined frequency bands. It is possible to see that the best performance is achieved when the entire range of frequencies (λ_4) is used. The worst case is for λ_1 (0.0 to 0.1Hz) and λ_2 and λ_3 showed similar results. Notice from Figure 6.33 that most of the coherence spectrum is below 0.2 Hz which explains the good performance obtained by λ_2 and λ_3 from which we can see that $\lambda_3 = 0.1 - 0.2$Hz showed a better performance in average than $\lambda_2 = 0.0 - 0.2$Hz.

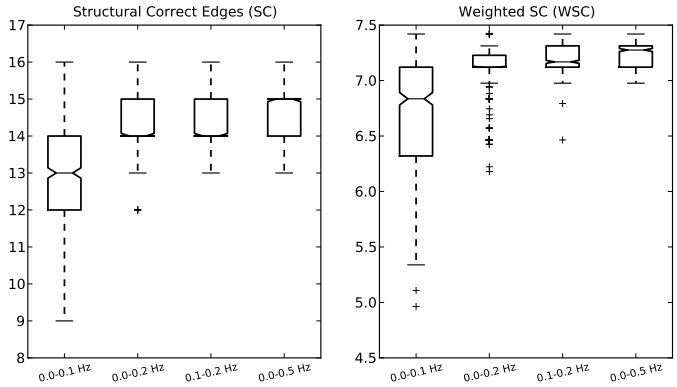

Figure 6.69: Structural correct (SC) and weighted SC (WSC) results for the 500 experiments using frequency bands λ. The box limits define the lower quartile and the upper quartile. The thickest line in each box is the median of the population for each case.

Figure 6.70 shows the correct directed edges found for the 10-node MVAR(1) network. This results are similar to ones shown in Figure 6.69 with λ_4 showing the best performance in average being followed by λ_3. Again λ_1 which covers from 0.0Hz to 0.1Hz had the worst performance inferring the correct edges and their causality.

Figure 6.71 shows the wrong directed edges found by the FBN algorithm. These are similar results to the previous measures with the exception of λ_3 for WWD which clearly had a better performance than λ_2 even though this last band covers more frequencies. Similar results to these ones are found in Figure 6.72 for the structural errors.

The use of a shorter range of frequency also helps to decrease importantly the computational load, since lower frequency elements are using making the cross spec-

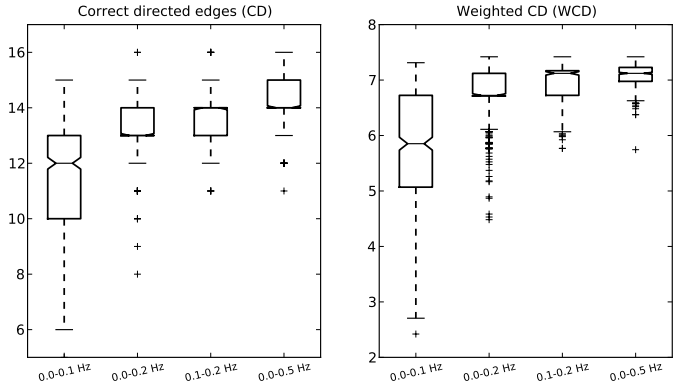

Figure 6.70: Correct directed (CD) and weighted CD (WCD) results for the 500 experiments using frequency bands λ. The box limits define the lower quartile and the upper quartile. The thickest line in each box is the median of the population for each case.

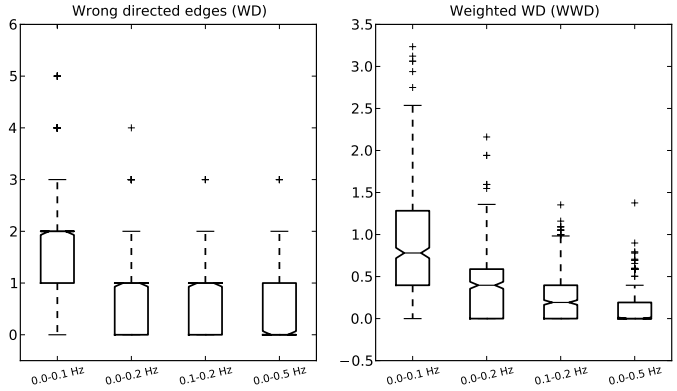

Figure 6.71: Wrong directed (WD) and weighted WD (WWD) results for the 500 experiments using frequency bands λ. The box limits define the lower quartile and the upper quartile. The thickest line in each box is the median of the population for each case.

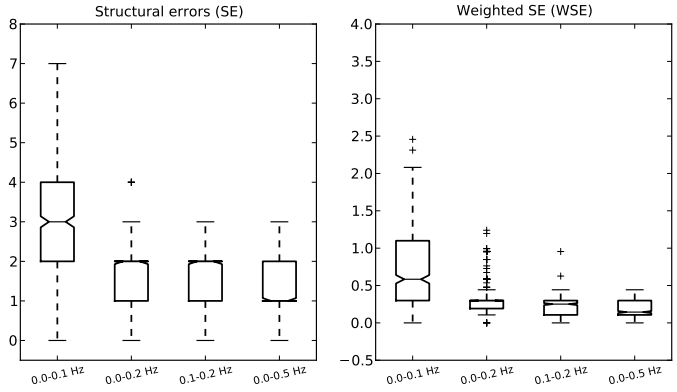

Figure 6.72: Structural error (SE) and weighted SE (WSE) results for the 500 experiments using frequency bands λ. The box limits define the lower quartile and the upper quartile. The thickest line in each box is the median of the population for each case.

tra matrices smaller. This can be seen in Figure 6.73 which shows the time used by each one of the 500 iterations in these experiments. The computer used for these experiments (and all experiments and results in this thesis) was a relatively fast one; Intel Core Duo (3.16 MHz per processor) and 3.2 GB of RAM under Ubuntu Linux. As expected using shorter frequency bands improves the processing time, but as we saw previously this has the cost of worsening the network inference performance. For instance, it can be seen that simulation using λ_4 took more than 150 seconds while λ_1 and λ_3 took 50 seconds approximately.

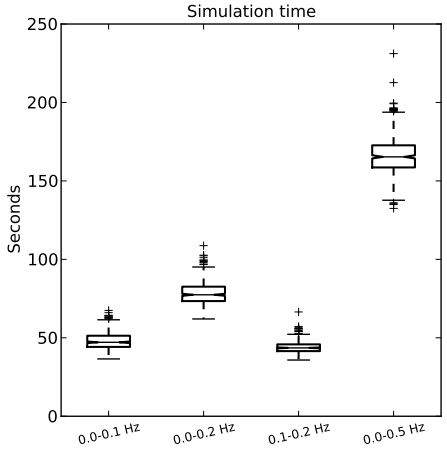

Figure 6.73: Simulation time for different frequency bands λ. $\lambda_1 = 0.0\text{Hz}-0.1\text{Hz}$, $\lambda_2 = 0.0\text{Hz}-0.2\text{Hz}$, $\lambda_3 = 0.1\text{Hz}-0.2\text{Hz}$, and $\lambda_4 = 0.0\text{Hz}-0.5\text{Hz}$.

6.5.4 MEG Beamforming and network inference

In this section, network inference experiments using LCMV beamforming are implemented. The scheme for the experiments is shown in Figure 6.74. The networks designed in Section 6.2.1 and Section 6.2.2 are fed into the MEG recording simulated environment as explained in Section 6.4. Then using the recorded signals which are in the sensor domain, these are mapped to the source domain by implementing the LCMV beamforming and its virtual electrodes. In this stage it is assumed that the spatial location of the N sources of interest is known. Using the virtual electrodes, the electrophysiological activity of these regions is extracted and their time series are fed into the FBN algorithm for network structure inference.

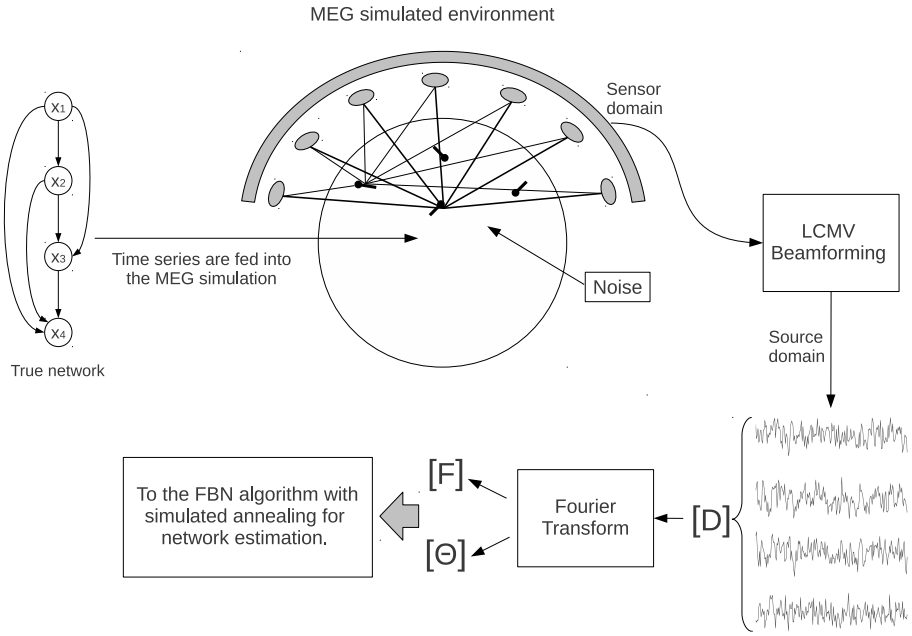

Figure 6.74: MEG simulated environment and brain network estimation scheme. An acyclic causal network is placed within the brain mesh. Then generated magnetic fields are acquired by simulated SQUIDs and the extraction of the time series is done using LCMV beamforming. For this simulation the position of the original sources is known in advance.

Figure 6.75 shows the coherence matrix of extracted signals using beamforming virtual electrodes for the 10-node MVAR(1) network system. Notice that coherence among the sources is affected by beamforming even though the exact position of the original sources is known. This issue is mainly because the LCMV beamforming assumes independence or at least decorrelation among the extracted brain sources. The decrease in coherence values for this system can be noticed better by comparing Figure 6.75 with the coherence matrix of the original network sources in Figure 6.33. The FBN parameters for these experiments are shown in Table 6.9.

Table 6.9: Simulated annealing hyperparameters for the 10-node network MVAR(1) experiments using beamforming.

numruns:	20000		initemp:	2.0
coolruns:	1500		fintemp:	0.01
burnedruns:	8000		alpha:	0.015
			beta:	0.04

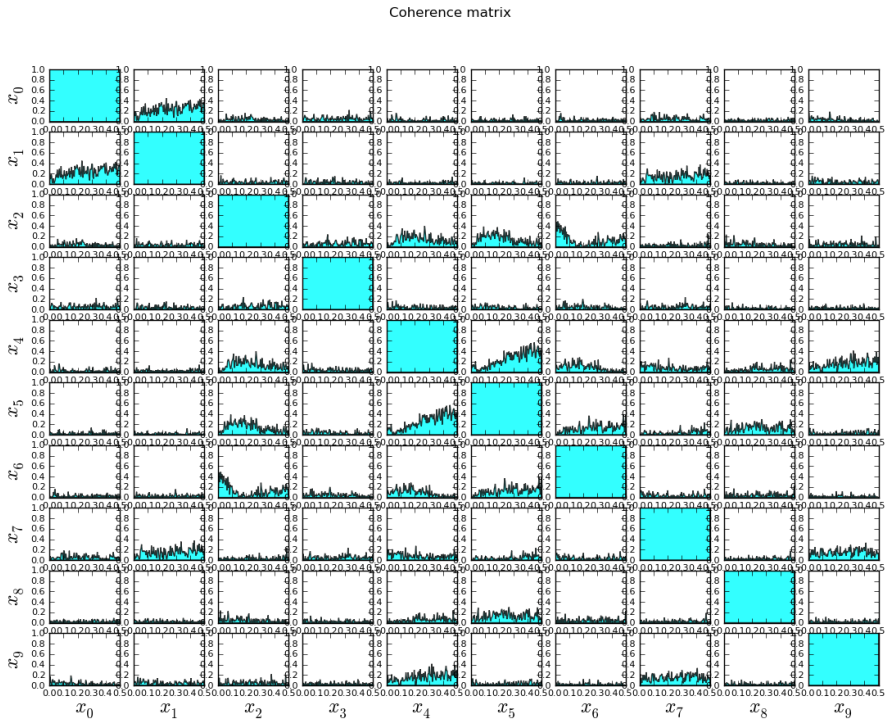

Figure 6.75: 10-node MVAR(1) network coherence matrix after beamforming. Compare with Figure 6.33 which is the coherence matrix of the original network. The coherence among the related sources is decreased due to source extraction by beamforming.

Figure 6.76 shows the cross-phase matrix of the extracted time series. An in-

teresting phenomenon here is that although coherence is highly affected by the beamforming extraction, the phase slope of the extracted time series almost remains intact. This can be seen by comparing the cross-phase matrix of the original sources in Figure 6.34.

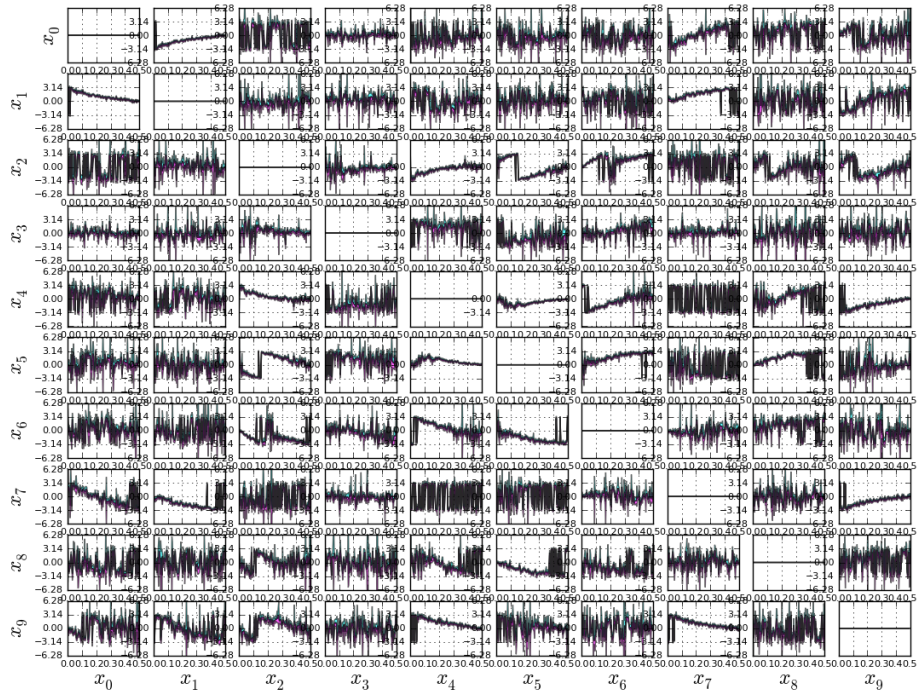

Figure 6.76: 10-node MVAR(1) network cross-phase matrix. Compare the cross-phase matrix with the one in Figure 6.34 and notice that after beamforming the phase slope is practically unaltered.

Figure 6.77 shows the inferred network using the beamformed extracted time series. The network is not accurate, it has noticeable differences when compared with the original system shown at the right of the same figure. The network inference performance measures for this single experiment are shown in Table 6.10.

Table 6.10: Single run results: 10-node MVAR(1) network.

Edges in $\mathbf{E}_{bd}$:	11	Edges in $\lfloor \mathbf{C}_{bd} \rfloor$:	11	
Number of edges in $\mathbf{C}_w$	20	Real network weight:	6.3160	
SC:	9	WSC:	5.0734	80.3261%
CD:	9	WCD:	5.0734	80.3261%
WD:	0	WWD:	0.0	0.0%
SE:	2	WSE:	1.2425	19.6722%

The results for the 10-node MVAR(2) are shown in Figures 6.78-6.80. Figure 6.78 shows the coherence matrix for this network system. As in the MVAR(1), the

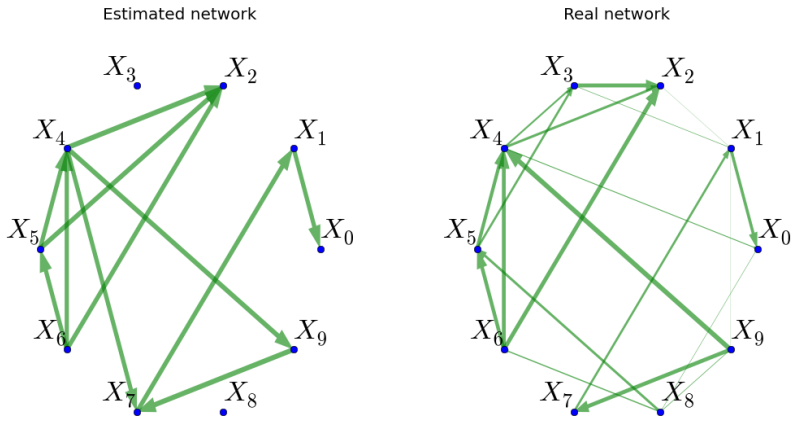

Figure 6.77: 10-node MVAR(1) inferred network structure after beamforming. It can be seen that for the algorithm it is harder to infer the original structure as shown in Figure 6.36.

coherence spectrum among the connected sources is highly decreased when compared with the coherence matrix of the original sources in Figure 6.37.

Figure 6.79 shows the cross-phase matrix. Again the phase slopes of the original sources and the ones extracted by beamforming are very similar, see the cross-phase matrix in Figure 6.38. The inferred network is shown in Figure 6.80 and the values of the network measures are shown in Table 6.11.

Table 6.11: Single run results: 10-node MVAR(2) network.

Edges in $\mathbf{E}_{bd}$:	11	Edges in $\lfloor \mathbf{C}_{bd} \rfloor$:	11	
Number of edges in $\mathbf{C}_w$	20	Real network weight:	6.3160	
SC:	9	WSC:	5.333	84.4363%
CD:	8	WCD:	4.5654	72.2830%
WD:	1	WWD:	0.7675	12.1516%
SE:	2	WSE:	0.9830	15.5636%

The NL(d) systems are also implemented in the MEG acquisition environment. For the 10-node NL(1) network, the coherence matrix of the extracted sources is shown in Figure 6.81. Again, beamforming affects seriously the coherence among the connected sources. The original coherence matrix which represents the real relation strength among the sources prior to beamforming is shown in Figure 6.41.

Figure 6.82 shows the cross-phase matrix. Comparing this matrix with the one in Figure 6.42 shows that the phase slope is preserved in some of the matrix elements. This means that for the FBN algorithm, network inference will be more difficult

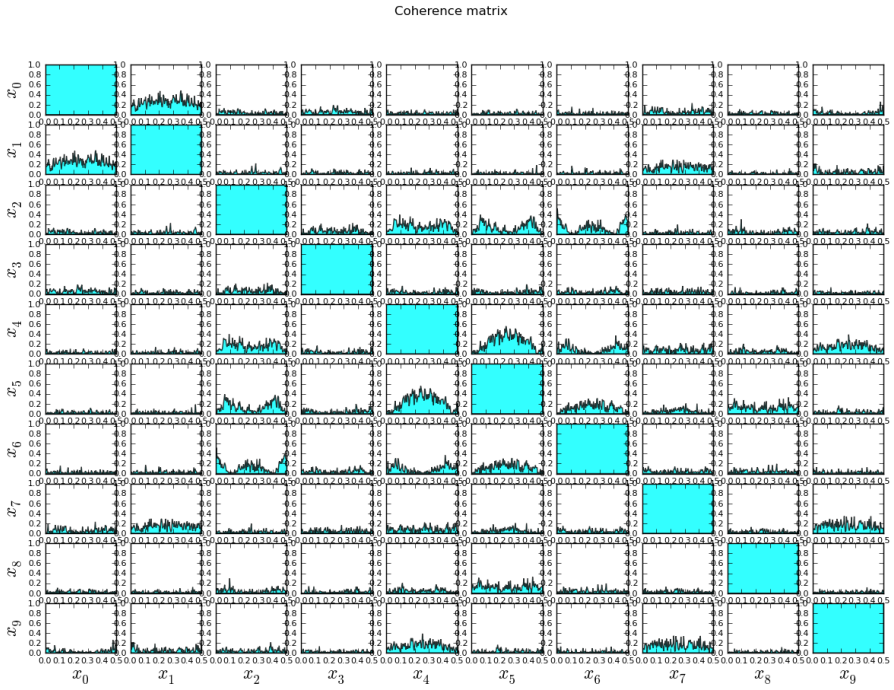

Figure 6.78: 10-node MVAR(2) network coherence matrix after beamforming. As in the previous case, the coherence spectrum among the connected sources is highly decreased after beamforming. The coherence matrix of the original network before beamforming can be seen in Figure 6.37.

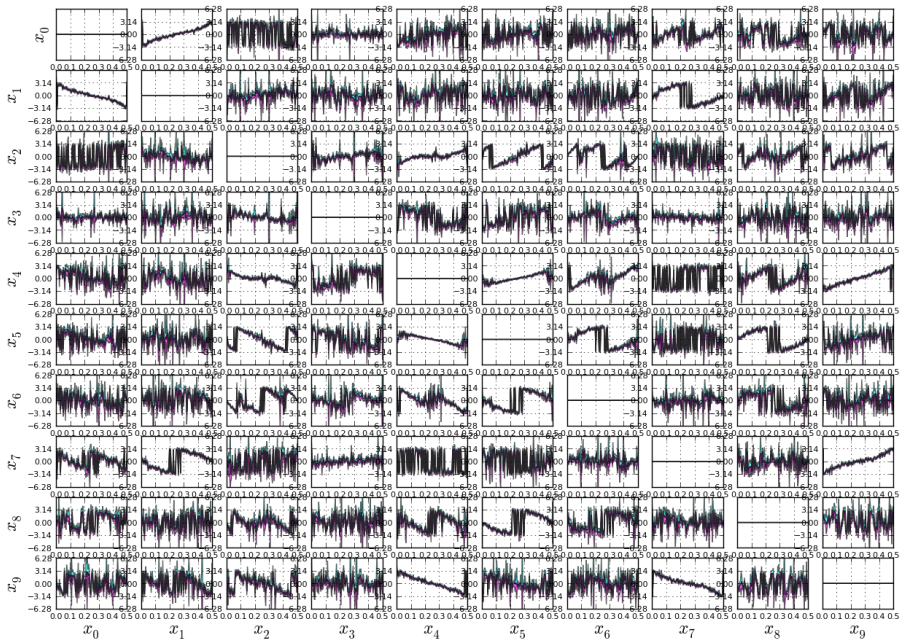

Figure 6.79: 10-node MVAR(2) network cross-phase matrix after beamforming.

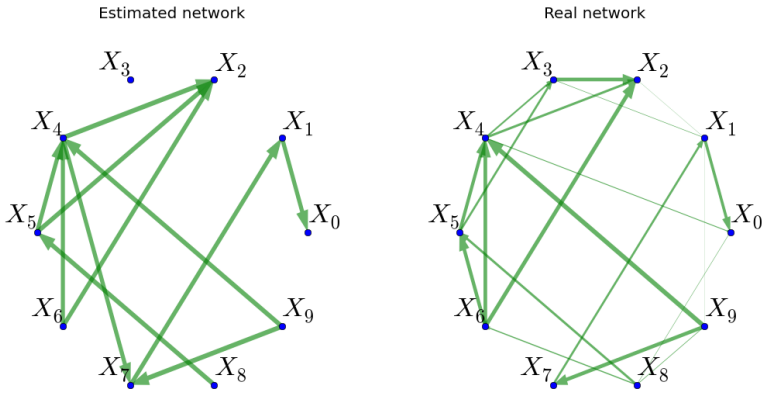

Figure 6.80: 10-node MVAR(2) inferred network structure after beamforming.

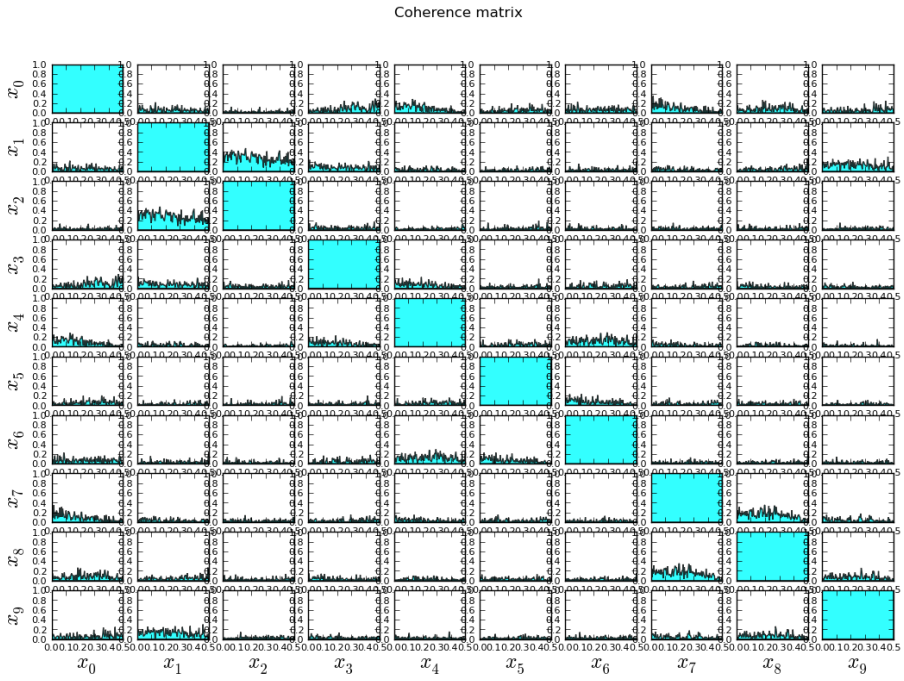

Figure 6.81: 10-node NL(1) network coherence matrix after beamforming. The coherence spectrums among the original sources can be seen in Figure 6.41.

since not only the coherence spectra are decreased but also the phase slopes among the connected sources are affected.

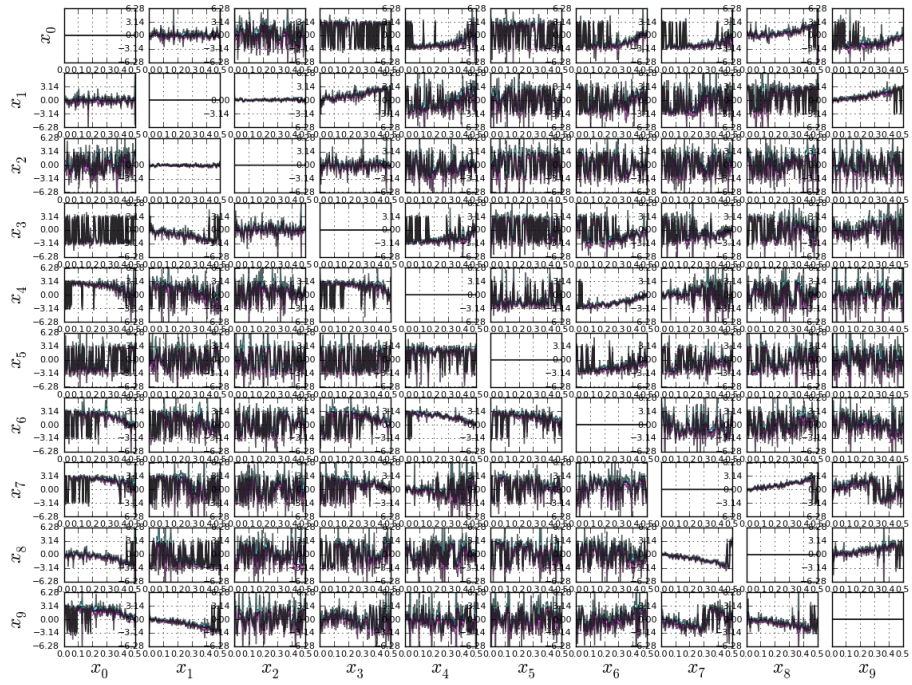

Figure 6.82: 10-node NL(1) network cross-phase matrix after beamforming.

Figure 6.83 shows the inferred network using the FBN algorithm. We can see that some of the structure was recovered with some edge errors. The network inference measures for this experiments are shown in Table 6.12, where we can see that only 68.16% of the thresholded structure was recovered correctly.

Table 6.12: Single run results: 10-node NL(1) network.

Edges in $\mathbf{E}_{bd}$:	14	Edges in $\lfloor \mathbf{C}_{bd} \rfloor$:	14	
Number of edges in $\mathbf{C}_w$	20	Real network weight:	8.6708	
SC:	8	WSC:	5.9101	68.16%
CD:	7	WCD:	5.0734	58.51%
WD:	1	WWD:	0.8367	9.64 %
SE:	6	WSE:	2.7606	31.83%

The last set of figures show the results for the 10-node NL(2) system. Figure 6.84 shows the coherence matrix, see Figure 6.45 for comparison. The cross-phase matrix is shown in Figure 6.85.

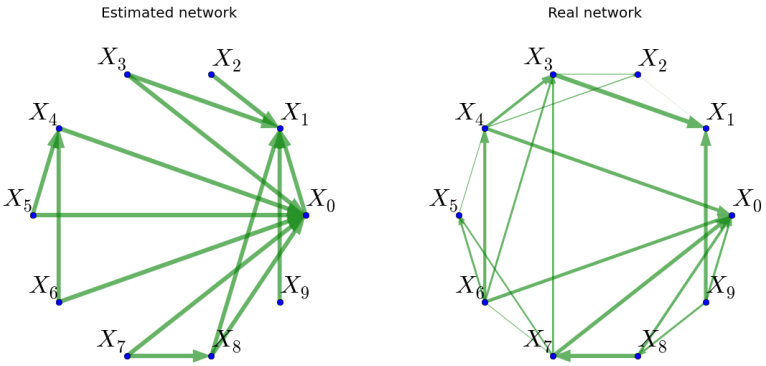

Figure 6.83: 10-node NL(1) inferred network structure estimation after beamforming.

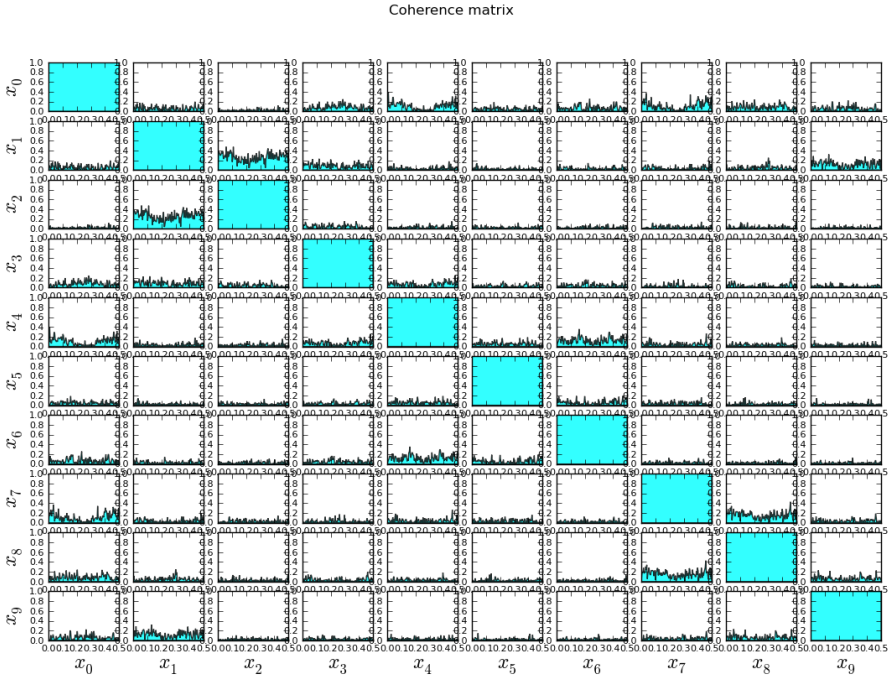

Figure 6.84: 10-node NL(2) network coherence matrix estimation after beamforming.

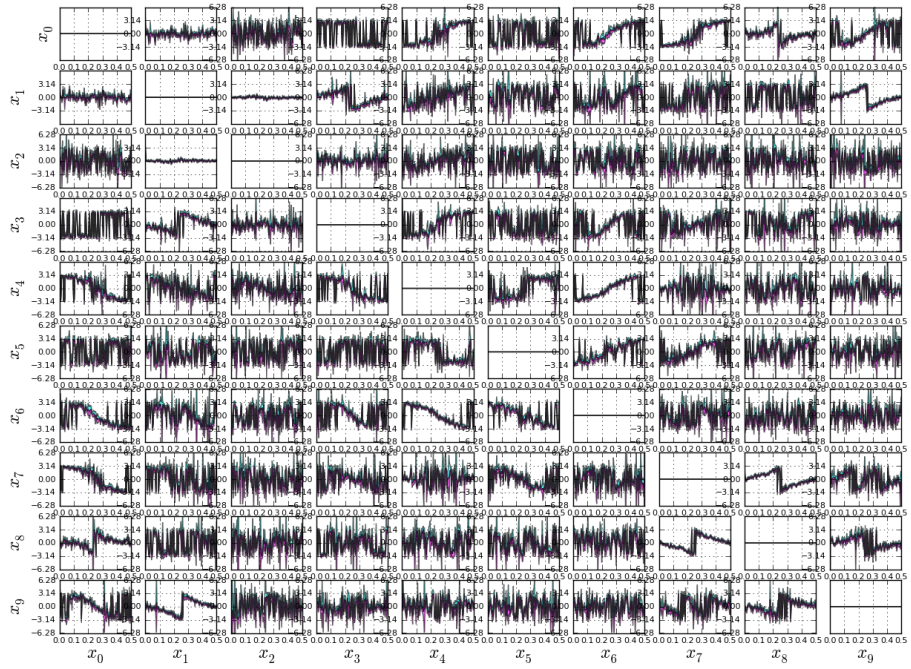

Figure 6.85: 10-node NL(2) network cross-phase matrix estimation after beamforming using the MEG acquisition simulated environment.

The inferred network using the FBN algorithm and the MEG simulated environment is shown in Figure 6.86-left for the 10-node NL(2) system. As in the 10-node NL(1) case, most of the network edges were recovered but some errors are also present. The network inference measures for this experiment are shown in Table 6.13.

Table 6.13: Single run results: 10-node NL(2) network.

Edges in $\mathbf{E}_{bd}$:	13	Edges in $\lfloor \mathbf{C}_{bd} \rfloor$:	13	
Number of edges in $\mathbf{C}_w$	20	Real network weight:	8.2536	
SC:	9	WSC:	6.4714	78.407%
CD:	9	WCD:	6.4714	78.407%
WD:	0	WWD:	0.0	0.0 %
SE:	4	WSE:	1.7821	21.591%

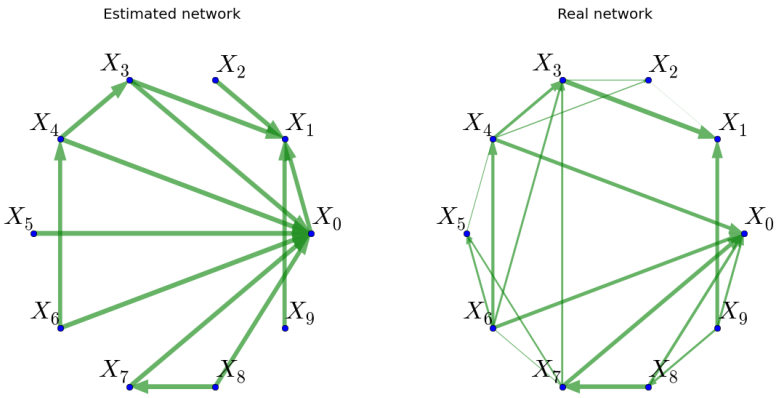

Figure 6.86: 10-node NL(2) inferred network structure using the MEG acquisition simulated environment.

6.5.5 Inferring large networks using LCMV beamforming

In Section 6.4 the LCMV beamforming was implemented to create NAI maps for the 10-node and 100-node network systems. Source extraction was reasonably fine for the 10-node networks, both the MVAR and NL. The 100-node network systems are not inferred correctly. An example of inferred network is shown in Figure 6.87 for the 100-node MVAR(1) network after beamforming. From the figure, it can be seen that the inferred network is completely different from the original shown at the right of Figure 6.87.

The differences between the original network and the estimated one can be observed more easily by watching the connectivity matrices shown in Figure 6.88. There are almost no similarities between the two matrices and the network inference measures shown in Table 6.14. The original network system was thresholded from 200 known weighted edges to 150 directed edges for comparison with the inferred network and only 18.05% of the original structure was retrieved correctly.

Experiments for the remaining 100-node systems will not be shown here because their performance are similar to the MVAR(1) case.

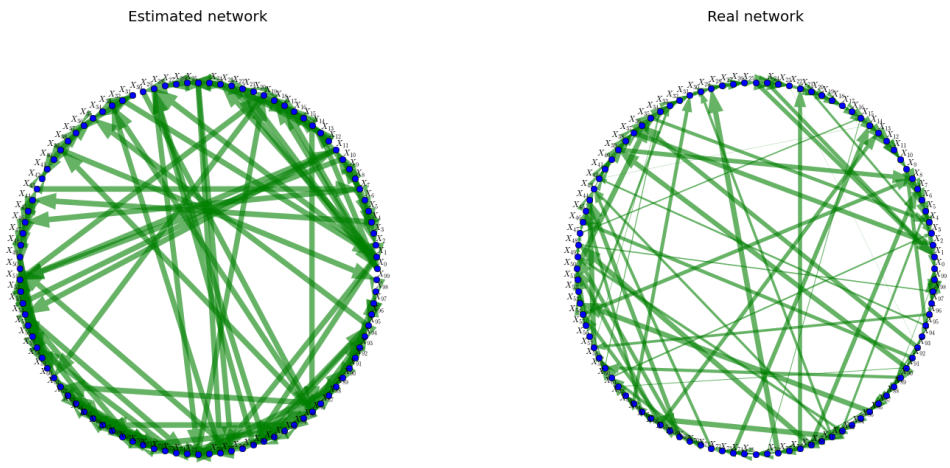

Figure 6.87: 100-node MVAR(1) inferred network structure using the MEG recording simulated environment.

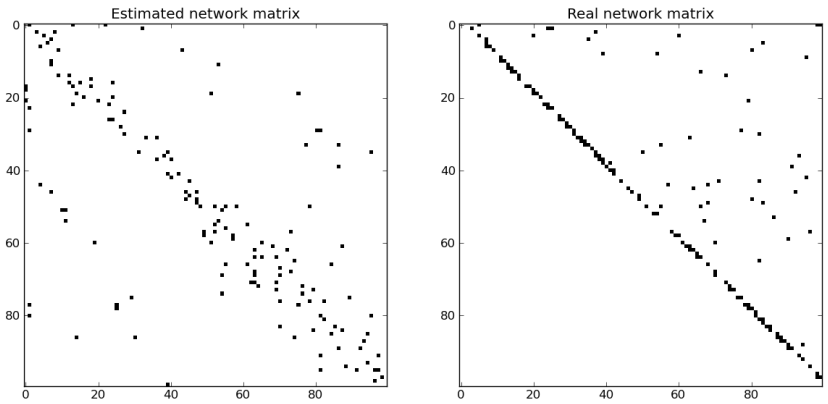

Figure 6.88: 100-node MVAR(1) inferred connectivity matrix shown at the left of the figure. The true connectivity matrix is shown at the right.

Table 6.14: Single run results: 100-node MVAR(1) network.

Edges in $\mathbf{E}_{bd}$:	150	Edges in $\lfloor \mathbf{C}_{bd} \rfloor$:	150	
Number of edges in $\mathbf{C}_w$	200	Real network weight:	85.3519	
SC:	28	WSC:	15.4079	18.0522%
CD:	17	WCD:	10.0614	11.7881%
WD:	11	WWD:	5.3464	6.2639 %
SE:	122	WSE:	69.9440	81.9477%

6.5.6 Beamforming behaviour in presence of multiple sources

The previous experiment using large networks with 100 sources and the LCMV beamforming raised an interesting question about how many sources can be accurately extracted by the beamforming virtual electrodes. Using the MEG simulated recording environment, several independent brain sources were placed in the brain mesh. Their position were chosen randomly with uniform distribution from 1 source to 200 sources. The brain sources were modelled as Gaussian noise segments of 30×256 samples.

Figure 6.89 shows the average correlation between the original sources within the brain mesh and the extracted times series using the LCMV beamforming. For every dot in this curve 20 iterations of the experiment were computed and at each iteration the positions of the sources within the brain mesh were chosen randomly with uniform distribution. We can see that the correlation curve in Figure 6.89 falls with an exponential decay and an average correlation of 0.5 is reached when 35 independent brain sources (modelled as Gaussian noise) are placed in the brain mesh.

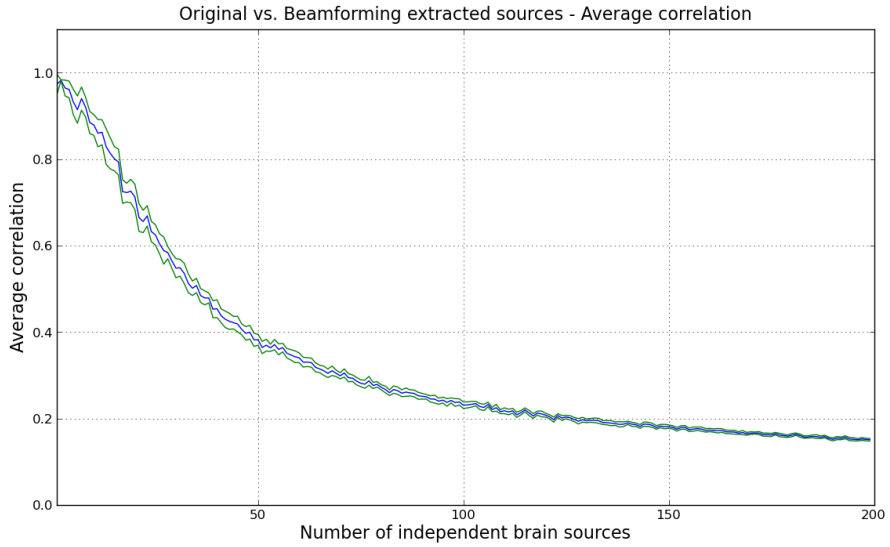

Figure 6.89: Beamforming vs. multiple brain sources. When the number of sources is low, the LCMV beamforming is able to extract correctly the activity of the brain sources. As the number of independent sources increases, the performance of the beamforming technique at the virtual electrodes tends to zero. The upper and lower lines in the curve indicate standard deviation.

This result is very important for the performance of the network inference technique proposed in this thesis and explains why the 100-node MVAR(1) network in the previous section was inferred with very low accuracy. For 100 independent sources the average correlation among the extracted sources is 0.23.

Chapter summary

In this chapter several experiments were done to test the performance of the FBN method. To achieve this, two different network systems were implemented; a linear one using the MVAR model and a non-linear version of the MVAR model, referred in this chapter as NL model. The delay of the implemented systems was made variable. In this chapter delays $d = 1$ and $d = 2$ were used. In addition of using two models with two different delays, two network sizes were implemented, 10-node and 100-node network. All these variables give a total of eight different networks which were used to study the behaviour of the FBN algorithm.

The different network models MVAR and NL helped us to test how the FBN network estimation method worked. First with a model that follows the linearity assumption of the FBNs. The NL model was implemented to test the performance of the FBN method with a model that does not follow completely the FBNs system assumptions.

Using the eight network systems, two different approaches were followed. First the FBN algorithm was tested by inferring network structures. The time series for these experiments were obtained directly from the defined network systems. This was done in order to test FBNs for network inference with the assumption that full access is gained to the network sources. Nevertheless, the aim application for FBNs is MEG and beamforming. For this reason a MEG recording simulated environment was also implemented.

In general, FBNs showed a good performance inferring both the 10-node MVAR and NL system for both model orders. FBNs were also compared with the well known DBNs inferring both systems, the MVAR and NL networks using different delays, $d = 1$ and $d = 2$. The simulations in this chapter showed that FBNs and

DBNs have similar results inferring $d = 1$ systems. For the case of DBNs this was expected since this algorithm was designed using one network unfolding with one time delay. Nevertheless, for the experiments involving order 2, MVAR(2) and NL(2), the performance of DBNs was highly affected by this parameter change while FBNs performance remained almost constant. This is an interesting characteristic for FBNs, which is not affected by the delay parameter. This property is inherited by the Fourier transform which makes FBNs a non-parametric technique.

The FBN algorithm was also proved for different frequency bands for the 10-node MVAR(1) network. Four frequency bands were proved: $\lambda_1 = 0.0 - 0.1$Hz, $\lambda_2 = 0.0 - 0.2$Hz, $\lambda_3 = 0.1 - 0.2$Hz, $\lambda_4 = 0.0 - 0.5$Hz. The simulations proved that using wider frequency bands improves the network estimation at the cost of computation load, and the inverse happens for narrower frequency bands. A nice trade-off between computational load versus network estimation accuracy might be choosing a band of frequencies where it is known that the cross spectrum is higher as occurs for narrower signals.

Simulations were also carried using the MEG recording simulated environment and FBNs. The same network systems tested for network inference were placed in the brain cortical mesh for beamforming analysis. The positions of the network sources were chosen randomly and then fixed. All the experiments presented in this chapter for MEG were single run simulations. The experiments showed that network inference performance is highly attached to the beamforming ability of extracting original network sources. This was proved for the 10-node systems where sources were extracted correctly in comparison with the 100-node network systems. As the number of sources increases, the LCMV beamforming performance decreases because the limited number of MEG sensors is not able to stop the magnetic fields radiated by sources at positions different to the position analysed.

Chapter 7

MEG resting state data sets

Chapter 2 reviewed MEG analysis and beamforming for brain source extraction. In this chapter the real data sets are studied. The database belongs to a previous study at the Hull York Medical School and the York Neuroimaging Centre (YNiC) on mild trauma brain injury (MTBI). It is composed of MEG recordings from three groups: a) a MTBI group whose MEG data were recorded 2 weeks after injury, b) The same group but now 6 months after injury, and c) a control group of healthy patients.

The MEG recording codes for every patient are shown in Table 7.1 as available in the YNiC database. All MEG recordings are resting state, meaning that the subjects just sat without doing a specific cognitive task. The recordings last 6 minutes where in the first half the patient keeps the eyes open while watching a fixation cross. The last 3 minutes the patients were in resting state with eyes closed, see Figure 7.1. The recordings were acquired with a sampling rate of 678.17 Hz and pre-filtered in the bands 2-40 Hz using a Butterworth filter of order 3.

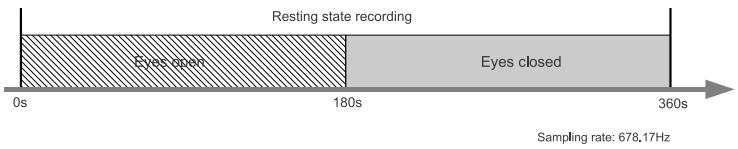

Figure 7.1: Resting state recording; 3 minutes eyes open with a fixation cross and then 3 minutes eyes closed. The sampling rate for these recordings is 678.17 Hz.

Table 7.1: YNiC MTBI MEG datasets (2 weeks after injury) and control group.

MTBI subjects	Control subjects
R1805	R1681
R1914	R2016
R1916	R2089
R1952	R2153
R1959	R2202
R1964	R2200
R1995	R2222
R2049	R2229
R2050	R2325

The analysis in this chapter is focused on the study of subject R2016 and was chosen randomly from the available MEG database in Table 7.1. The FBN algorithm will be applied for network inference to the eyes closed segment. The pre-analysis steps are shown in Figure 7.2 and these are explained in the following paragraphs.

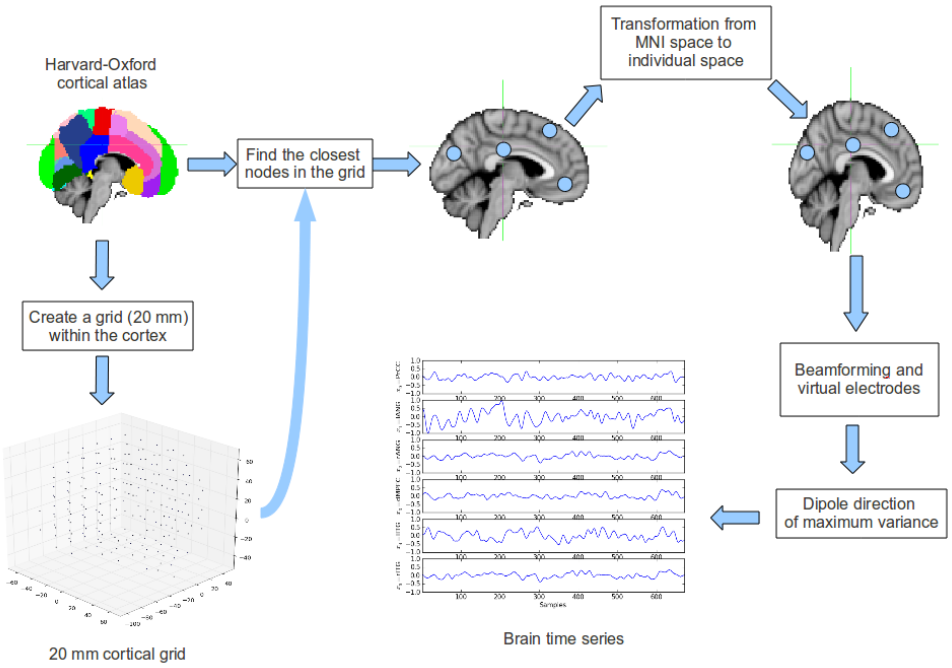

Figure 7.2: Preprocessing steps for the experiments in this chapter. Colour image can be seen in Figure B.21.

In order to obtain brain source time series, the vectorized type I beamforming is applied (see Section 2.6.1). This beamforming is implemented in the YNiC's Neuroimaging Analysis Framework[1] (NAF) software. The NAF software is a suite

[1] http://vcs.ynic.york.ac.uk/docs/naf/

of python classes and GUIs that perform several preprocessing tasks for the MEG recordings. Using the NAF, the original datasets were coregistered to the subjects' MRI brain and then transformed to MNI standard brain volume. All virtual electrodes (VE) are also extracted in the coregistered subjects' brain volume space and then transformed to MNI brain standard. This procedure can also be implemented backwards by choosing VEs in the MNI space and then these coordinates can be transformed to the individuals' brain space. The brain space transformation steps are performed automatically inside the NAF software. This means that once all subjects' brains were coregistered and transformed to MNI standard brain, it is possible to work in the MNI space domain seamlessly for intersubject comparisons.

The study in this chapter is based on a study published by Jiao et al. (2011). Jiao et al. (2011) published a directed graph describing the causal connectivity for the resting state network (or default mode network (DMN)) using fMRI and Granger causality as the connectivity technique. This network is reproduced in Figure 7.3 and it is composed of seven brain regions: The dorsal medial prefrontal cortex (dMPFC), ventral medial prefrontal cortex (vMPFC), left and right inferior temporal gyri (lITG and rITG), left and right angular gyri (lANG and rANG), and the posterior cingulate/precuneus cortices (PCC/PrCC).

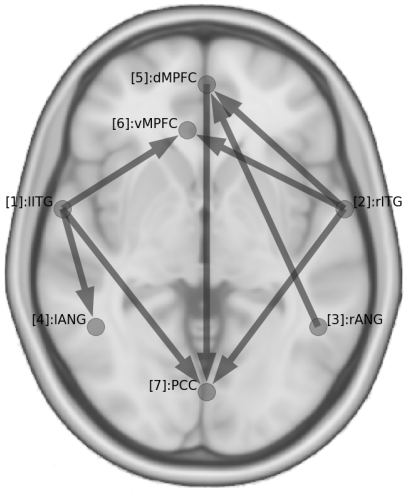

Figure 7.3: Granger causality inference of the default mode network or DMN. Figure adapted from Jiao et al. (2011).

In order to define the nodes of interest, a node grid of 20 mm space between

neighbour nodes is defined using the Harvard-Oxford cortical atlas in MNI space. This step gives a total of 220 nodes within the cortex. Then using the coregistration matrices between the subject's MRI brain and the MNI brain, the MNI grid is transformed to the subject's head space for beamforming analysis. This pre-analysis step is implemented in the NAF software.

Using FSL and the structural MNI brain, the regions defined in Figure 7.3 were chosen and their XYZ coordinates are shown in Table 7.2. Then using the defined cortical grid, the grid nodes closest to the defined points are estimated. These points are shown in Table 7.2 as grid coordinates. The grid coordinates are the ones used for the beamforming virtual electrodes.

Table 7.2: DMN regions and grid coordinates.

DMN regions	XYZ coordinates	Grid coordinates
PrCC	[0, 36, 28]	[-10, 34, 28]
PCC	[0, 14, 30]	[10, 14, 28]
lANG	[56, -54, 26]	[50, -46, 28]
rANG	[-46, -56, 26]	[-50, -46, 28]
dMPFC	[0, 30, 52]	[10, 34, 48]
vMPFC	[0, 44, -24]	[10, 54, 28]
lITG	[52, -20, -34	[50, -26, -32]
rITG	[-54, -30, -26]	[-50, -26, -32]

The vectorized LCMV beamforming (Huang et al. , 2004) described in Section 2.6.1 and available in the NAF software gives three normalized NAI components representing three dimensional coordinates XYZ. The NAI components represent the normalized power of the extracted time series using virtual electrodes. The NAF software divides the MEG time series in contiguous segments of 990 milliseconds, giving a total of 671 samples per segment. The covariance matrix is computed using a concatenated time series using all segmented epochs. This produces 180 segments of 990 milliseconds from which the vectorized beamforming is computed.

For connectivity analysis in MEG using beamforming it is important to define a fixed direction for the analysed dipoles at each virtual electrode, see Fuchs (2007). Here the dipole direction vector is computed as the one that maximizes the power of the resultant time series using the X, Y, and Z normalized VEs (VE_x, VE_y, and VE_z). The resultant VE time series can be represented as

$$VE(t) = \max_{a,b,c} \|aVE_x(t) + bVE_y(t) + cVE_z(t)\| \tag{7.1}$$

where $VE_x(t)$ is the x time series component given by the vectorized Type I beamforming. and the constants a, b, c define a unitary vector whose direction is the direction of the current dipole at the grid node of interest. The a, b, c vector is fixed for all 180 segments and this assures that the computed current dipole does not rotate its direction during the recording. With the resultant VEs computed using Eq. 7.1 at the grid coordinates defined in Table 7.2, it is now possible to run the FBN algorithm to find the best network per segment.

Due to their close physical and network relation, the regions for PrCC-PCC and dMPFC-vMPFC were fused to create two new regions PrCC=[10,14,28] and dMPFC=[10,34,48]. This means that regions PCC and vMPFC are not included in the following experiments. The main reason for this is because the beamforming weights of these regions are highly correlated, meaning that the VEs will infer highly correlated signals. This is explained in Section 7.1.

For demonstration purposes, single VE extraction and network estimation are shown for segments 25 and 55. Figure 7.4 shows the VE time series for segment 25 using the procedure described previously. As for the segment 25, the other 179 VE segments are extracted and the FBN algorithm is run for each segment to infer connectivity among the VE sources.

The coherence matrix for the time series in Figure 7.4 is shown in Figure 7.5-top using a FFT with 128 samples, 75% overlapping and a square window for the periodogram estimation. The cross-phase matrix is shown in Figure 7.5-bottom.

Using the spectral properties of FBNs, the algorithm is set to estimate the connectivity network for the band 2-40 Hz. Notice that this band gives 8 rounded up power and phase coefficients when the FFT is computed with 128-sample windows. The simulated annealing hyperparameters for these simulations are shown in Table 7.3 and the estimated network is shown in Figure 7.6. The steps for network estimation using FBNs are shown in Figure 6.32 but here real VE-MEG time series are used.

Figure 7.7 shows the VE time series for segment 55. As in the previous exam-

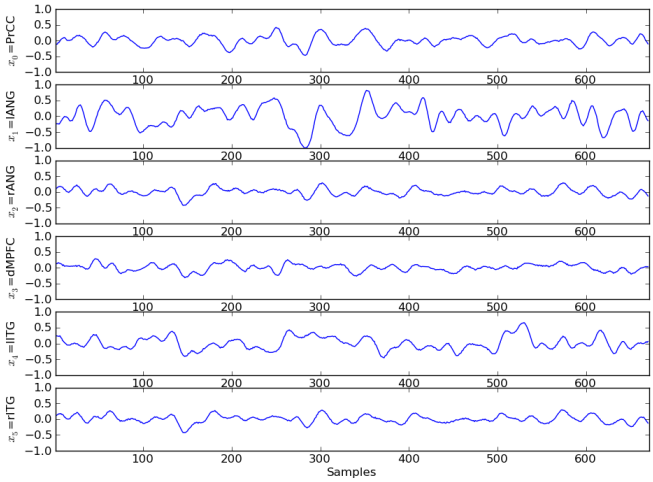

Figure 7.4: Extracted and resultant VEs from subject R2016. The segment shown corresponds to segment 25 from 180 available.

Table 7.3: Simulated annealing hyperparameters for real MEG data sets R2016.

numruns:	12000		initemp:	2.0
coolruns:	1500		fintemp:	0.02
burnedruns:	8000		alpha:	0.04
			beta:	0.02

ple, coherence and cross-phase matrices for this segment are shown in Figure 7.8. Figure 7.9 shows the estimated network for these time series.

Notice that the spectral nature of the FBN method allows to try different segment lengths for the FFT. Much of the reliability of the FBN method is based on the estimation of the spectra and cross-phases among the nodes of interest. The following examples are obtained using the same data segments 25 and 55 from recording R2016 but now a 256-sample window size is used for the FFT algorithm with 75% of overlapping and a square weight window to compute the periodogram. Furthermore, the hyperparameters changed for these examples; alpha=0.085 and beta=0.04. The rest of the hyperparameters kept their values. The frequency band analysed by the FBN method is again 2-40 Hz, which for a 256-sample window means that 15 rounded up spectral coefficients are used in FBN method.

Figure 7.10 shows the coherence and cross-phase matrices for segment 25. Notice these coherence spectra are noisier or with higher variability that the ones shown in Figure 7.5. Recall that longer FFT windows increases power spectrum variability

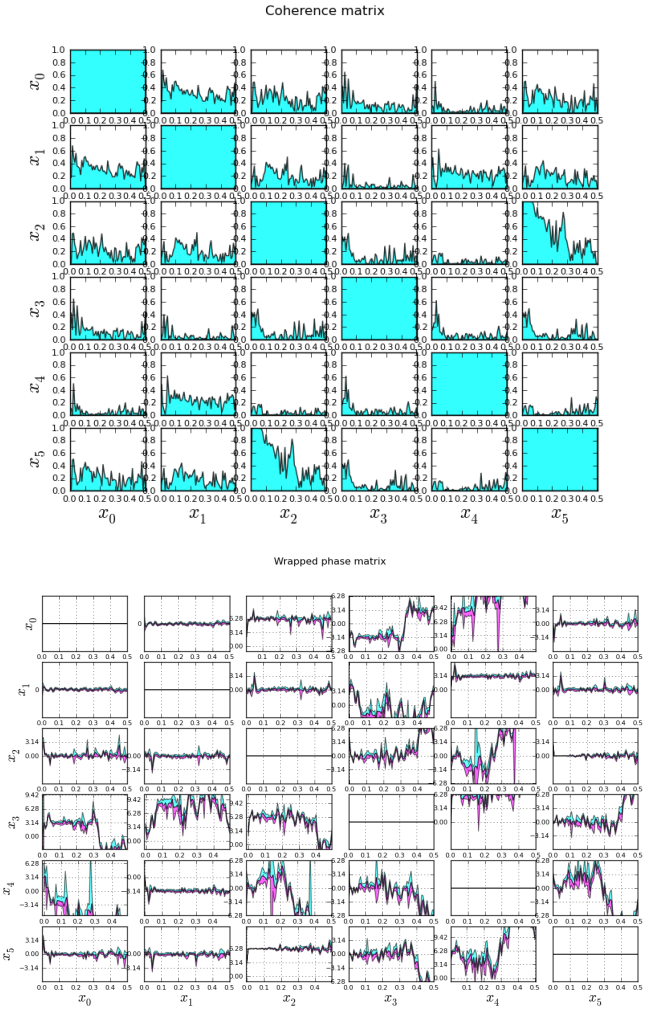

Figure 7.5: Coherence and cross-phase matrices using 128 samples for the FFT for the extracted VE segment 25.

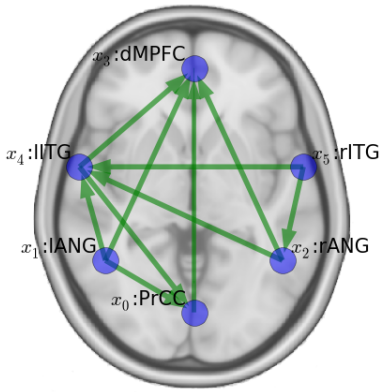

Figure 7.6: Estimated network using the proposed FBN algorithm for real MEG data segments. This network was estimated for recording R2016, segment 25, using a 128-sample window for the FFT algorithm and a frequency band 2-40 Hz for the FBN method.

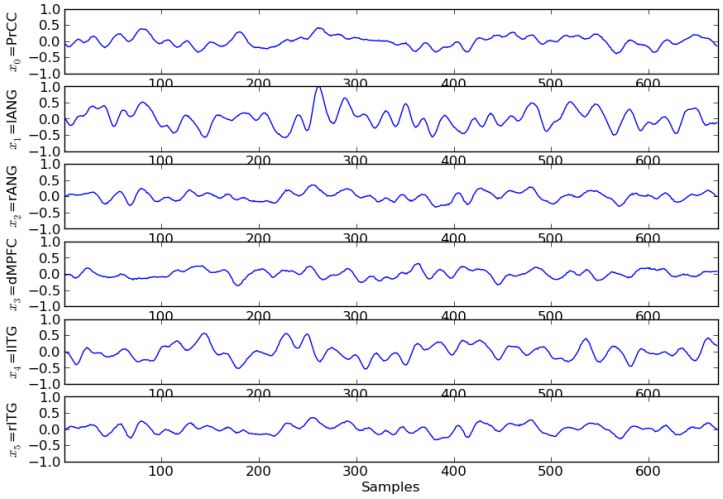

Figure 7.7: VE time series for the defined DMN regions. These time series are obtained from segment 55.

but improves frequency resolution.

Figure 7.11 shows the estimated network using 256 sample segments for the FFT algorithm. Notice that the network structure is drastically changed just by modifying the length for FFT window, showing the importance of the spectral estimation step.

The same previous experiment is repeated for segment 55. The coherence and phase matrices are shown in Figure 7.12 and the estimated network in Figure 7.13. Again notice that the structure of the estimated network changed when compared with the one shown in Figure 7.9.

As noticed, the differences among inferred networks for the same VE segment are produced by the length of FFT window. This highlights the need of establishing a procedure or criterion to assure statistical significance for the estimated network and/or their edges. This topic will be covered in the discussion section.

The previous experiments showed results for segments 25 and 55. Showing results for the other 178 segments will not be possible here, but it is possible to show some statistics about the behaviour of the network inference in the entire eyes closed recording. Figure 7.14 shows the inferred causality for the 180 edges for all found edges using 256-sample window for the FFT. The maximum number of times a directed edge can be estimated is 18000 (180 segments times 100 network samples

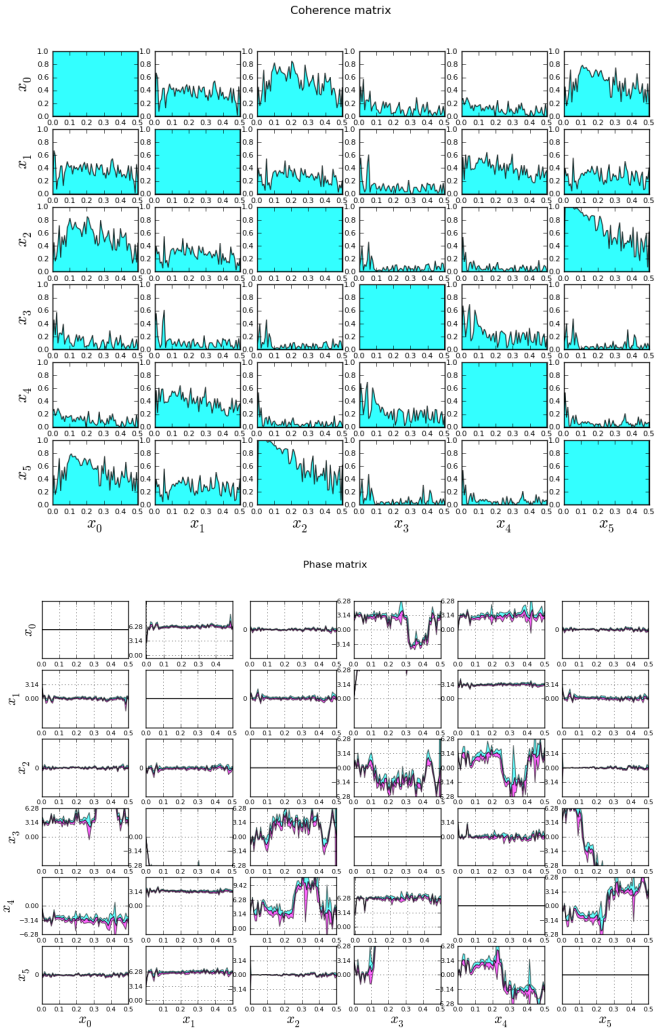

Figure 7.8: Coherence and phase matrix for segment 55. The FFT uses 128 samples with 75% overlapping, and square window for the periodogram.

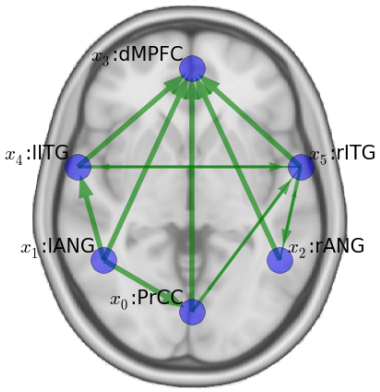

Figure 7.9: Estimated network for segment 55 using 128-sample window for the FFT algorithm and a frequency band of 2-40 Hz for the FBN method. The hyperparameters for the simulated annealing are shown in Table 7.3.

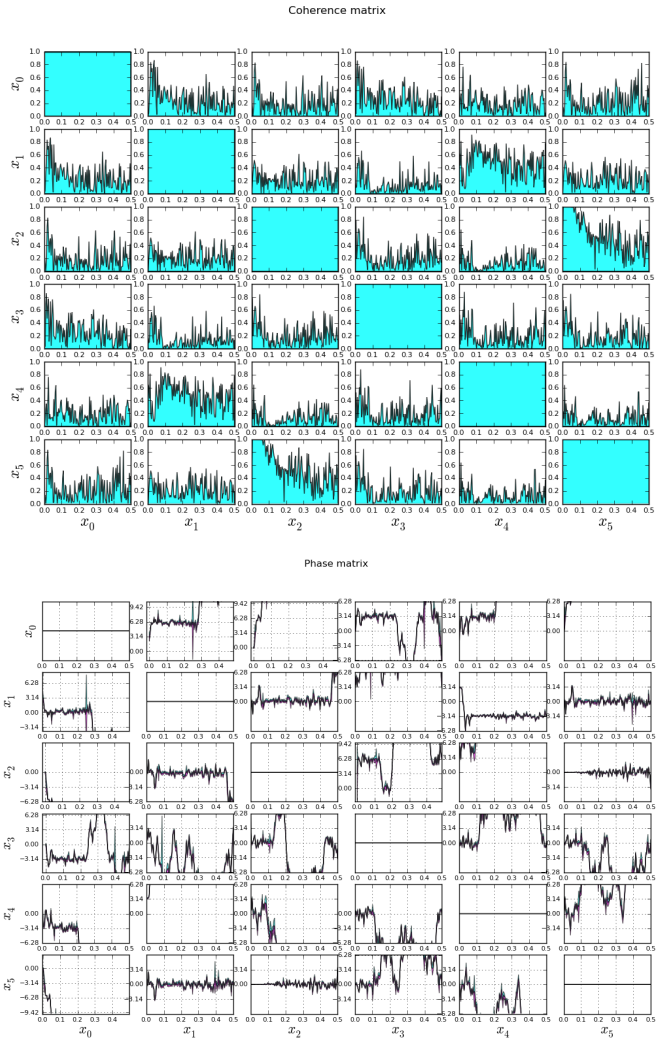

Figure 7.10: Coherence and cross-phase matrices for segment 25 using 256-sample window for the FFT algorithm.

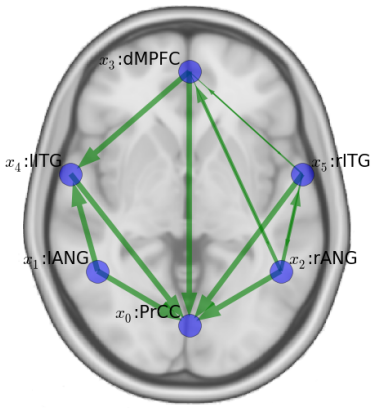

Figure 7.11: Estimated network for segment 25 using 256-sample windows for the FFT. The analysed frequency band is 2-40 Hz.

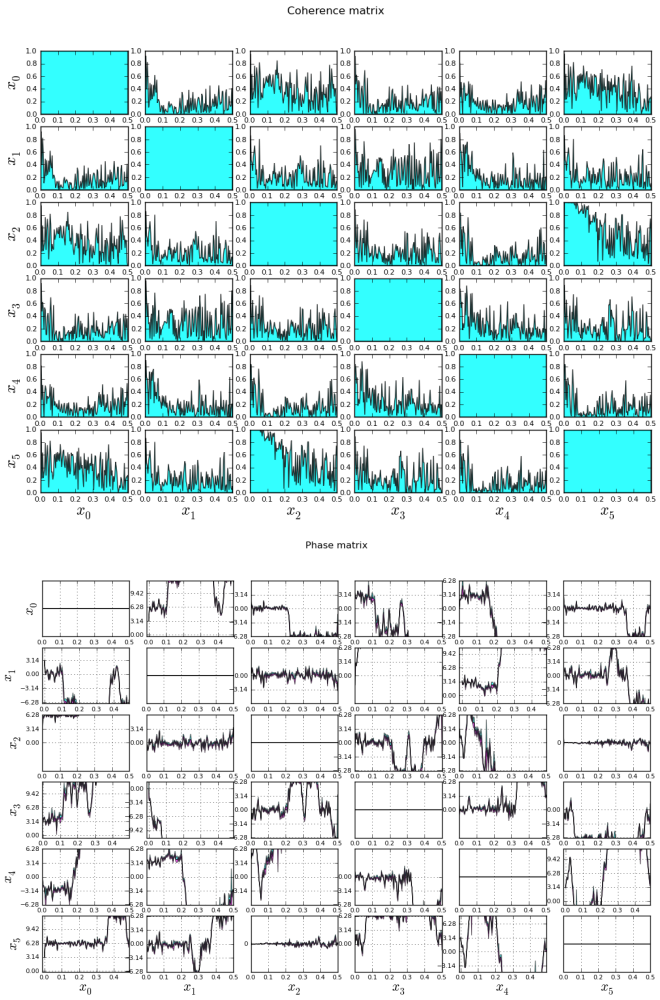

Figure 7.12: Coherence and cross-phase matrices for segment 55 using 256-sample windows to compute the FFT with 75% overlapping and square weighting window for the periodogram.

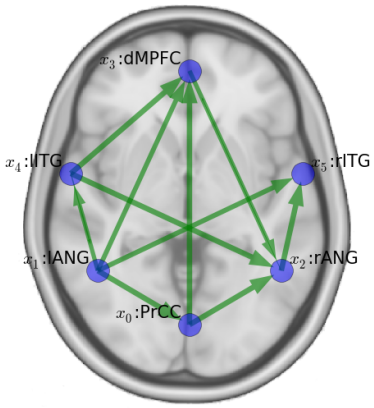

Figure 7.13: Estimated network for segment 55 using a 256-sample windows for the FFT with 75% overlapping and squared weighting window. Notice the differences with Figure 7.9.

from simulated annealing). As expected some edges are estimated more frequently than others as it is the case for rANG-rITG, lANG-lITG, and lITG-rITG, but unexpectedly, causality was estimated practically evenly for both directions ($A \rightarrow B$ and $A \leftarrow B$).

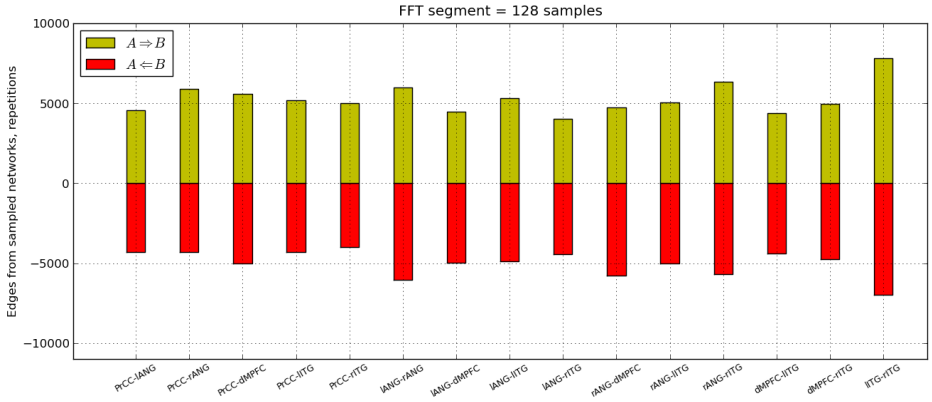

Figure 7.14: Inferred edge causality for the 180 MEG segments using 128-sample windows for the FFT. The maximum number of times an arrow can be sampled is 18000. The bars shows all possible edges in the network.

Figure 7.15 shows the same information as in Figure 7.14 but ignoring causality. This is in order to show the frequency of the estimated edges for the 180 segments. As in Figure 7.14, the bar plot in 7.15 was created by adding up the 100 network matrices sampled by the FBN method per each one of the 180 segments, giving a total of 18000 sampled networks and this number is the maximum number of times each edge can be sampled, see Figure 6.32.

Figure 7.16 shows the same experiments but now for FFT windows of 256 samples with the same periodogram computation (75% overlapping and square weighting window). Notice that results are similar to the previous results for the 128-FFT case but now there are more estimated edges in average for the networks. Figure 7.17 shows all sampled edges but ignoring causality. In this case it seems that all possible edges are sampled with approximately same frequency.

The almost even causality estimations for the inferred edges seems to be caused by the highly dynamic nature of the brain and the activity it generates. This can be seen by plotting the cross-phase frequency for some of the edges of interest for the analysed frequency band 2-40 Hz. Figure 7.18 shows the cross-phase between

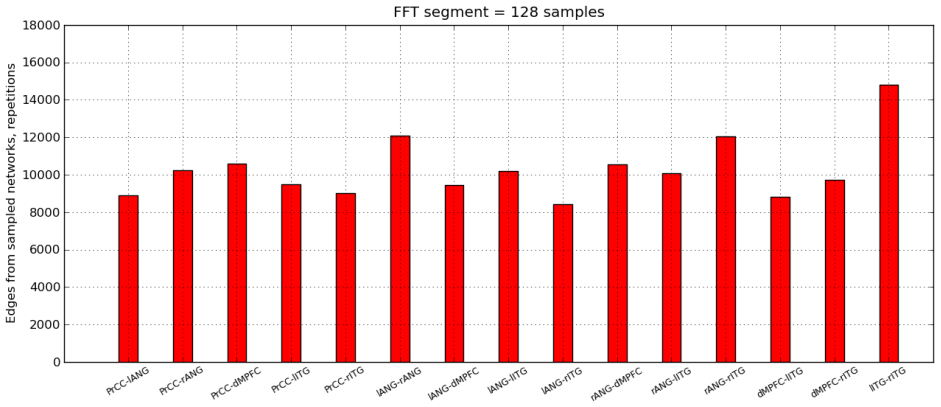

Figure 7.15: Edge estimation frequency of all possible edges in the networks using windows of 128 samples for the FFT. These bars show same information than Figure 7.14 but causality is ignored here.

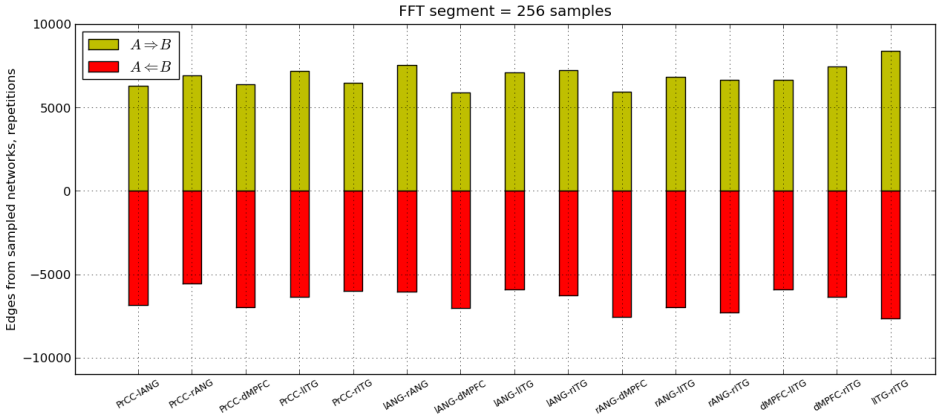

Figure 7.16: Inferred edge causality for 180 MEG segments using a 256-sample windows for the FFT algorithm.

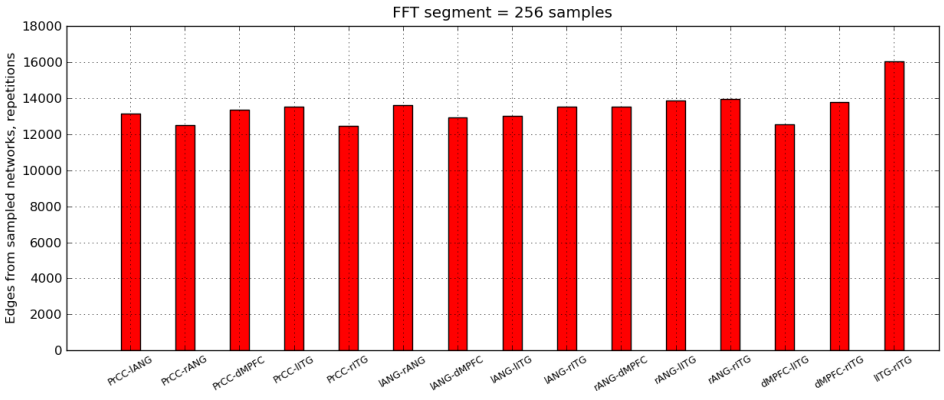

Figure 7.17: Edge estimation frequency of all possible edges in the networks using segments of 256 samples for the FFT. These bars show same information than Figure 7.16 but causality is ignored here.

six node pairs of interest: PrCC-lANG, PrCC-dMPFC, lANG-lITG, rANG-rITG, dMPFC-lITG, and dMPFC-rANG. The six graphs show the cross-phase of 180 segments for the frequency band 2-40 Hz. It can be seen that the phase slope sign changes almost evenly (there are similar number of positive and negative slopes), proving the dynamic nature of the brain recordings. Recall that the direction of the brain current dipoles is constant for all 180 segments.

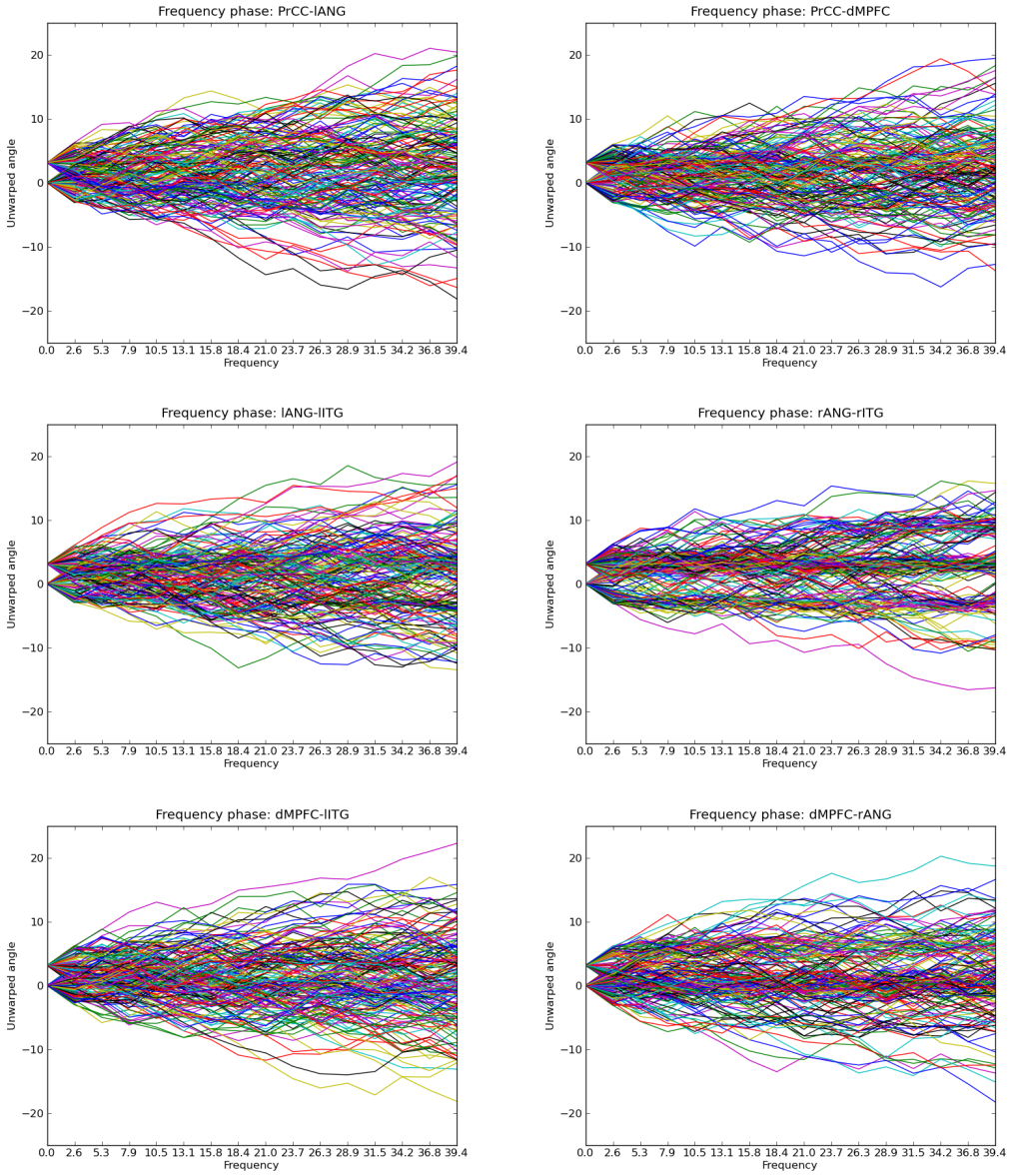

Figure 7.18: Cross-phase frequency for six edges of interest between 2 and 40 Hz. The edges are from top to bottom and from left to right: PrCC-lANG, PrCC-dMPFC, lANG-lITG, rANG-rITG, dMPFC-lITG, and dMPFC-rANG. Colour image can be seen in Figure B.22.

The high dynamics and variability of the inferred networks suggest that there are network patterns that are repetitive along the entire MEG recording. Since the recording was made for resting state with eyes closed it is plausible that some patterns repeat. This was partially proved empirically by running the FBN algorithm several times for several segments, but a manner to see this phenomenon is implemented here. The FBN algorithm was run for all 180 segments from which 180 networks were estimated. Then, the directed acyclic connectivity matrices inferred for every segment were reshaped from 6×6 matrices to 36×1 column vectors. Using these connectivity vectors a correlation matrix among them is computed. Figure 7.19 shows the correlation matrix of the estimated 180 connectivity vectors where only correlation values above zero are preserved and the rest are zeroed out, this was done because negative correlations do not have a structural meaning.

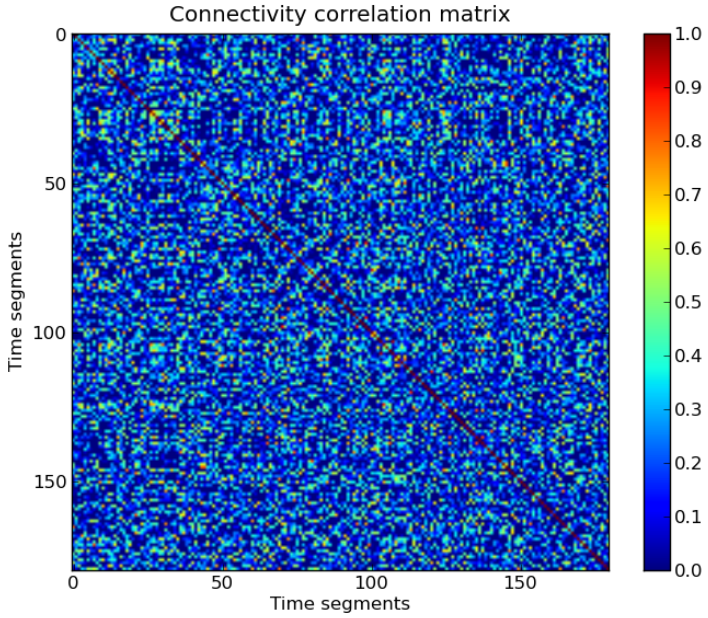

Figure 7.19: Correlation matrix of the connectivity vectors. The 180 acyclic directed estimated matrices were reshaped from 6×6 to 36×1 column vector. Then, correlations among all of them are computed. Colour image can be seen in Figure B.23.

In order to identify network structures, a clustering method is performed on the network correlation matrix. This procedure will group networks with similar structure and causality. The method used for the hierarchical clustering is the farthest point algorithm available for python (python module: scipy.cluster.hierarchy.linkage).

The clustered matrix is shown in Figure 7.20 with its respective dendogram which uses a tree graph showing the hierarchical clustering. There are approximately 13 identifiable network clusters which define network structure groups.

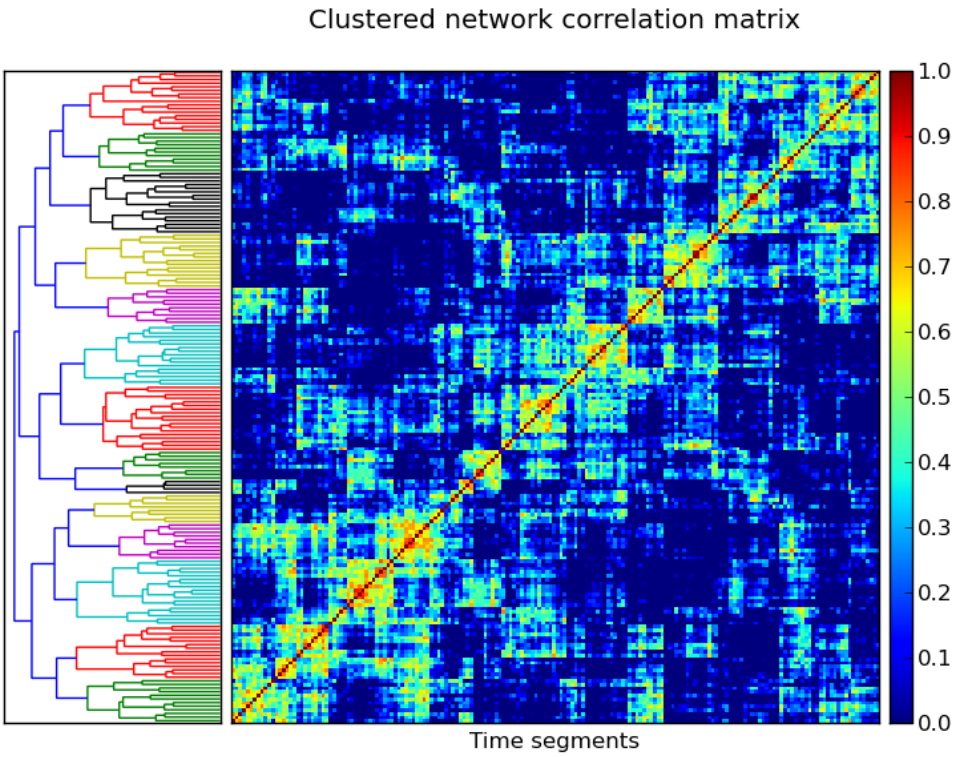

Figure 7.20: Clustered network correlation matrix. The hierarchical clustering is shown by the dendogram graph which shows approximately 13 network clusters. Colour image can be seen in Figure B.24.

Figure 7.21 shows a histogram for the clustered networks for the 180 segments showing which network patterns were more common. As can be seen the most common networks for these results are the resting state networks (RSN) RSN2 and RSN6 followed by RSN7, RSN10, and RSN12. These networks are shown in Figure 7.22 where the nodes lANG, rANG, lITG are the most connected nodes. The network patterns in Figure 7.22 were obtained by averaging the networks within each cluster in order to obtain a representative network for each cluster. Here the edge threshold $Pnum = 10$.

An interesting property to study is the evolution of the inferred network patterns through time. This time evolution can be seen in Figure 7.23 for the 180 MEG

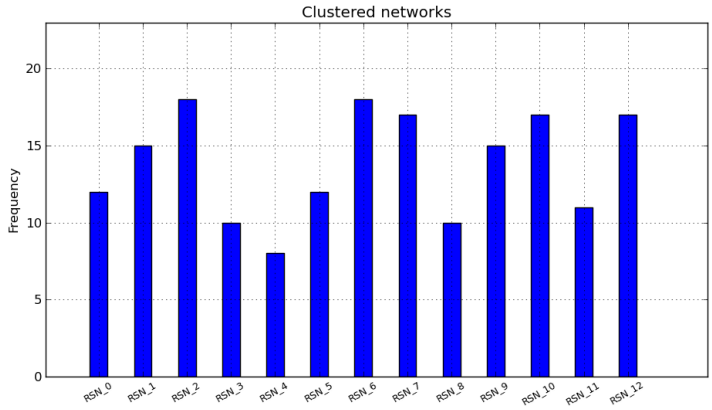

Figure 7.21: Network pattern histogram. The histogram shows the number of times each of the identified network patterns in the clustered correlation matrix appears in the recording.

segments. Interestingly a smooth network change from pattern to pattern can be appreciated in the graph through time. This suggests that occurrence of network patterns is not completely random.

7.1 Discussion of the experiments

An important issue to have in mind when estimating networks from MEG datasets using beamforming is the weight vector of each of the regions of interest. As it was explained in previous chapters the beamforming weights defines the spatial filter that linearly unmixes the recording to infer the activity in the desired voxel. Weight vectors of different and separated voxels might infer highly correlated signals if their weights are similar. This issue is pointed out in Brookes et al. (2011b) where it is suggested to compute correlations among the weights to find possible spurious brain connections. For the experiments in this chapter, the beamforming weight correlation matrix is shown in Figure 7.24 for the 8 regions involved in the default mode network shown in Figure 7.3. For instance, Figure 7.24 shows that beamforming weights of dMPFC and PrCC are highly correlated and this might be the reason for finding an edge between these two regions.

Nevertheless, in beamforming there is still the possibility that the high correlation between two spatial filter weights is caused by a real interaction between two

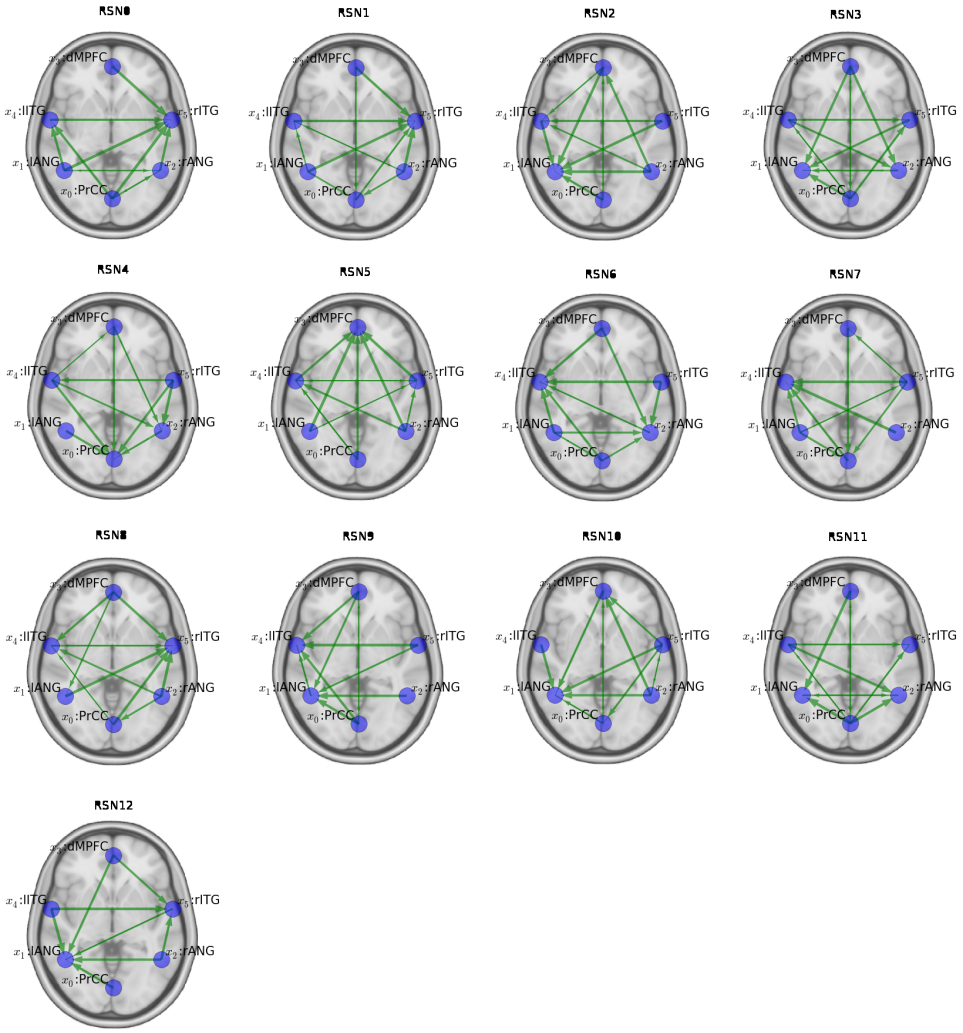

Figure 7.22: Network patterns for the resting state MEG recording, subject R2016. This figure shows the 13 identified networks patterns that are estimated by the FBN method. The patterns were found using a hierarchical clustering method and then each cluster is averaged to obtain a representative network of each cluster. Colour image can be seen in Figure B.25.

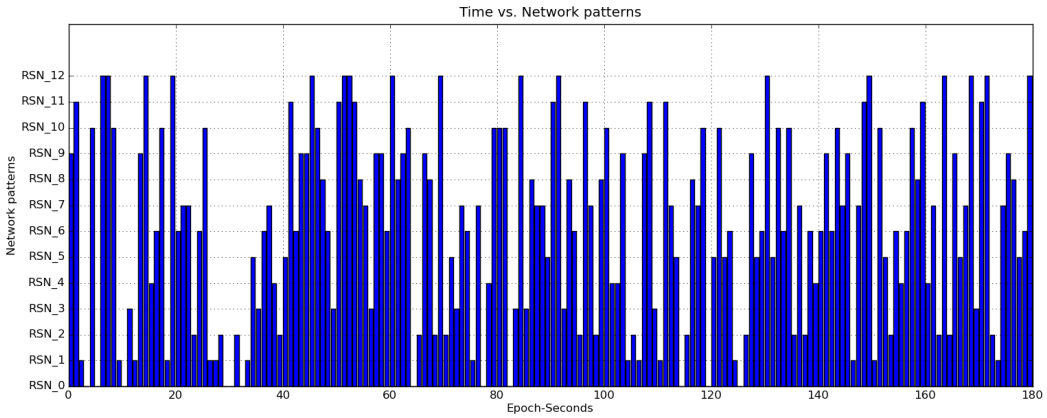

Figure 7.23: Network pattern evolution through time during the MEG eyes closed segments.The bars show the networks in Figure 7.22 vs. time. Notice that there are smooth changes from pattern to pattern through time.

regions. Recall from Chapter 2 that when two regions or voxels are highly correlated, the beamforming technique tends to fuse both sources in a single source which also produces similar spatial filter weights for both regions. This problem might be solved by using different beamforming techniques designed to deal with correlated sources by imposing extra constraints for the spatial filter in the Lagrangian, like dynamic imaging of coherent sources (DICS) (Gross J. et al. , 2001) or nulling beamforming (Hui et al , 2010). But these techniques were not implemented for simulations in this thesis and they are not implemented in the YNiC's NAF software.

The networks found by the FBN method in the R2016 MEG resting state recording show that recurring network patterns appear in a resting state brain. These network patterns are shown in Figure 7.22 and their occurrence in the recording is shown in Figure 7.23. This is the most compelling result of this chapter, nevertheless it is a preliminary result. For instance, FBNs are able to find the best network that fits the datasets, but their ability to find the network depends on the hyperparameters used, especially the alpha and beta parameters which tune the algorithm by increasing or decreasing the weight of the number of edges and the phase slope respectively in the posterior probability function. For instance in this chapter, using 128 or 256 samples for the FFT algorithm requires different beta and alpha values. In the future more work will be needed to correctly tune the FBN algorithm.

The inferred networks using FBNs should also be assessed by statistical signif-

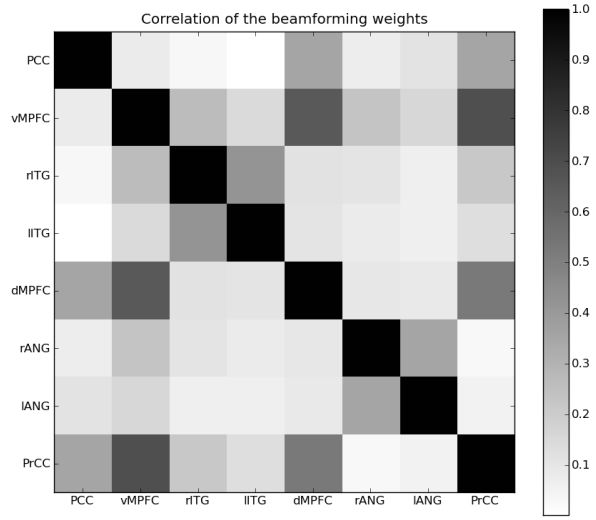

Figure 7.24: Correlation matrix of beamforming weights. The beamforming weight correlation matrix might indicate if a link between two virtual electrodes is produced by real brain connectivity or by volume conduction.

icance. This means that a test should be applied to all the inferred edges in order to assure that there is really an edge there.

In general, this chapter proves that FBNs have potential application to MEG network studies. This technique is designed to deal with large number of sources and also large datasets. Because it is based on the Fourier transform it is catalogued as a non-parametric technique. Power and phase are analysed separately in the posterior probability distribution and no suppositions or models are assumed for power and phase besides the coefficients provided by the FFT-periodogram and phase estimation. Furthermore, FBNs can also find networks at specific band of frequencies thanks to its spectral nature.

Chapter summary

In this chapter the FBN algorithm was tested on real datasets for the first time. The datasets at hand belong to a current project at YNiC (University of York) on MTBI. This dataset is composed of 27 MEG recordings: 9 controls, 9 MTBI subjects 2 weeks after injury and the same 9 MTBI subjects 6 months after injury. For the experiments in this chapter subject R2016 was chosen randomly and the vectorized

LCMV beamforming was applied to the regions defined by the DMN published by Jiao et al. (2011), see Figure 7.2.

After all the pre-analysis steps what is obtained are 180 segments of 990 milliseconds. Each segment is composed of 6 time series which represent the 6 defined nodes of interest. Then the FBN method is applied to all 180 segments to infer connectivity among these sources. Interestingly, after several runs of the algorithm using the extracted time series, it was noticed that common network patterns appeared in several segments. This phenomenon led to analysis of which and where in time these common networks occur. In order to achieve this, a hierarchical clustering was implemented using correlations among all connectivity vectors obtained from the inferred network structures. This created a 180×180 clustered matrix where 13 clusters could be identified, see Figure 7.20. The networks composing each cluster are averaged to obtain a representative network of each cluster. These 13 networks are shown in Figure 7.22. This is possibly the most important result of this thesis, because this suggests that something recurring is happening in the dynamics of a resting state brain and these network patterns appear frequently during the recording. Their occurrence is not chaotic, there is an smooth changing of patterns through time as can be seen in Figure 7.23. These results definitively compel future research.

Chapter 8

Discussion and future work

8.1 Discussion

Brain networks have won great importance in recent years. The paths used by brain neurons to communicate is an important field of knowledge which is still partially explored even with the current technologies and the extensive research work that has been done. The resting state network or default mode network is maybe one of the most recent and important findings in Neuroscience and Neuroimaging. At the beginning it raised scepticism in the research community but today it is a well established research area.

The resting state network has been pointed as a potential marker for brain related diseases and conditions. The reason of this is because this network is assumed to be constant in all humans and this has been confirmed by several research groups in different countries. A universal functional brain network like the DMN would allow to study how this network is affected given that there is a possible brain condition or disease.

MEG network studies have not been as popular as in fMRI also known as fcMRI (functional connectivity MRI). The reasons of this might be because MRI devices are much more common than MEG ones, the fMRI community is bigger than the MEG community, and also the availability of Open Source software is higher in fMRI. Nevertheless in recent years and possibly fuelled by the need of studying the dynamic activity of the brain and its effective connectome, the number of research

works relating MEG and brain networks has increased. This also raises the need to develop new techniques and protocols designed for brain connectivity in MEG.

The proposition of this thesis is to apply Bayesian network theory in the Fourier domain, hence the name of Fourier Bayesian networks or FBNs, to the study of effective brain networks. Bayesian networks have been applied with success in fMRI studies where the BOLD time series values are rounded to discrete levels to feed the BN searching algorithm using discrete distributions. Discrete BNs have the property of being non-linear which improves the network estimation since an interaction model among the network nodes is not assumed at the cost of losing information due to discretization of the data.

Applications of discrete BNs for MEG brain functional studies is not possible using the VE time series directly, current approaches consist of taking the signals envelope (or instant power) using the Hilbert transform and converting these to discrete levels for the BN algorithm. A second approach is using Gaussian Bayesian networks which can deal with time series at the cost of assuming a linear interaction among the sources which is mainly governed by a linear regression whose error is assumed Gaussian (hence the name). Causality in Gaussian BNs is inferred by network unfolding which means duplicating sources as delayed versions of the original time series. This approach might work in some applications where the delay among sources can be guessed but in the case of Electrophysiology this delay is not known, and sometimes it can be variable due to the dynamics of the nervous system. For these reasons, this thesis proposes FBNs as an approach for brain connectivity.

Properties of FBNs

FBNs work by transforming the source time series in the Fourier domain. Working in the Fourier domain allows to separate the Likelihood distributions in two factors taking the advantage that in this domain power spectrum is independent of phase spectrum, leading to a likelihood function for the network power and a likelihood function for the network phase. The likelihood phase allows to evaluate causality without setting the source delay as a parameter, a common practice in discrete dynamic BNs. FBNs have interesting properties that makes them an appealing

approach for brain functional connectivity and maybe other fields too. These are:

- *FBNs inherits the ability of BNs of dealing with large number of sources.* One of the properties that made BNs so important in genetic or protein networks is their ability to build networks from large number of sources which is the case of the proteins transcribed inside the cell. In the case of MEG maybe it is not possible to have hundreds of sources due to the physical constraints in MEG given by the number of SQUIDs but it is still an important property.

- *FBNs are able to work with long time series.* Long time series are common in Electrophysiology studies, which is a positive characteristic due to the amount of information provided but in the case of Gaussian BNs it increases the amount of computations needed for network structure inference since the linear regression is performed in time domain. By transforming the time series in the Fourier domain a long time series segment is transformed to a fixed window defined by the spectral coefficients which at the level of the algorithm produces smaller matrix computations. Also, if FBNs are estimated at specific band of frequencies it also decreases the size of the matrices. This FBN property might also represent a disadvantage since brain time series are known to be non-stationary and by computing the Fourier transform and the periodogram the entire segment is considered stationary.

- *FBNs deal with dynamic phase variability - causality.* As previously explained FBNs work with the frequency power and frequency phase. In order to estimate the delay between the parent source and the child source the frequency phase slope is estimated within the band of interest. In Gaussian dynamic BNs (DBNs) causality is estimated by delaying the time series a certain amount of samples, commonly one or two. FBNs do not need to specify the number of delayed samples, because it estimates the phase slope which will always be positive or negative regardless of the delay length. Also, this delay does not have to be homogeneous among the sources, it can have different values. This is not possible in Gaussian DBNs. In DBN the network must be unfolded several delays and network searching algorithm must optimize the entire network.

- *FBNs are non-parametric.* FBNs inherits this property from the Fourier transform. The spectral regression of the Fourier transform does not imply a signal model besides the one provided by the Fourier coefficients. These also allows to estimate parent-child delay from the phase slope only. Nevertheless, the network structure searching algorithms require hyperparameters that control or tune the searching.

- *FBNs can perform network searching at user-defined frequency bands.* This property is also inherited from the Fourier transform and it was tested in Section 6.5.3. Network searching at frequency bands can be performed by just doing spectral regression and slope estimation using the Fourier coefficients that represent the desired frequencies.

Thesis simulations

The simulations presented in Chapter 6 were designed to study the performance of FBNs under controlled environments. The first set of simulations tested how FBNs behave inferring simulated networks using two source-to-source interaction models: a linear model MVAR(p) and a non-linear NL(p) model, where p defines the time lag or delay parameter. These simulations proved that FBNs are successful at inferring the modelled network for all cases and behave better than DBNs even when the time lag parameter p changes. FBN performance was practically invariant to changes in the model order p.

For the second set of experiments a MEG simulated environment was implemented. These simulations were designed to test the performance of FBNs for the application of interest in this thesis in a controlled environment where the network system to be inferred is truly known. In these simulations FBNs showed good performance inferring networks although not as good as the previous set of simulations due to the volume conduction problem produced by the space between the brain sources and MEG SQUIDS, and the LCMV beamforming. These simulations also provided an idea of how difficult it is to infer networks after beamforming and also that the number of inferred sources can not be large due to physical constraints of the MEG device such as the number of SQUIDs. For instance, simulations showed

results for a 100-node network which was impossible to estimate after beamforming even knowing the exact position and orientation of dipoles.

Possibly, simulations using the MEG environment were not as accurate as desired because no volume head model for the secondary dipole currents was implemented such as the sphere model, multisphere model or finite element method that describes the electromagnetic properties of the head flesh and skull. Nevertheless, all these models at the end provide a linear mixing of the sources that reflect their activity to the MEG squids. A linear mixing occurs due to the space between the modelled sources and the MEG SQUIDs, this second mixing is modelled in our simulations for the primary dipole currents. The use of a more realistic head model needs to be covered as future work.

FBNs and the MEG resting state datasets

The experiments with real MEG datasets were more challenging than simulations implemented in Chapter 6 since in a real resting state brain it is not possible to know in advance what to find or search for and this topic will certainly be a focus of future research. However, these experiments were able to test the behaviour of the proposed technique with real MEG time series. From the available MEG database the recording of subject R2016 from the control group was chosen for analysis. This recording consists of 360 seconds of resting state with the first half being eyes open and the second half (180 seconds) eyes closed. The 180 seconds for eyes closed were chosen to avoid eye movement or blink artefacts. Then 6 regions which are currently known to be part of the DMN were defined, these are: PrCC, lITG, rITG, lANG, rANG and dMPFC. Then, beamforming was applied to extract the source activity at these regions. Beamforming is implemented in the YNiC's NAF software for MEG data analysis.

The results of these experiments showed promise even though these are preliminary results that will need much more research work. The FBN method was able to infer from subject R2016 approximately 13 common network structures or patterns that manifest in the recording during the resting state with eyes closed. Importantly, these networks do not show up in a chaotic manner but there seems to be a smooth

change among the 13 inferred networks. These results are shown in Figures 7.20, 7.22 and 7.23.

8.2 Future work

The results obtained from this thesis and the proposed technique compel future research in order to study more exhaustively the behaviour, performance, and properties of FBNs and their potential applications. In this section eleven areas of future research are described. These areas of future research are mentioned with the intention of improving the current state of FBNs.

1. *Noise analysis.* For the experiments in Chapter 6, noise was fixed to a SNR= 30 for MEG. This is a high SNR which might not reflect the real difficulties found in MEG recordings. A study of the behaviour and perfomance of FBNs under different SNRs for network estimation and also beamforming will be needed in the future.

2. *Statistical significance of the inferred edges.* The FBN method using the simulated annealing algorithm or other optimization algorithms aims to find the best or more probable network within the posterior probability distribution where constraints such as the complexity of the network can be added during the network sampling. These sampled networks will represent the best set of networks that describes the data. Nevertheless, a separate statistical test that indicates or proves that there exists real interaction between the sources connected by the inferred network should be computed. This needs to be done in order to prove that there is indeed a network with statistical significance. The most direct method to analyse statistical significance between a child node and its parent nodes is by studying the regression coefficients in the spectral domain. Recall that FBNs work by performing a spectral regression which is governed by regression coefficients. These are an indicator of connection strength.

3. *Likelihood function for the network phase probability.* This is a pending issue

for the FBN development. The current function in Eq. 5.79 is a heuristic approach that aims to evaluate a network with a higher value if the network edges have the correct causality and with a lower value for the contrary. For practical purposes it works well but the FBN method needs a likelihood phase distribution with a physical meaning which should come from the proposed structure S_h. Future work on this function is necessary.

4. *Hyperparameters methodology.* The simulations in Chapter 6 and the experiments with real MEG datasets in Chapter 7 showed that the algorithm's hyperparameters which control the behaviour of the network searching are of paramount importance for the correct estimation of the system's network. A wrong set of hyperparameters might lead to completely incorrect results in the worst scenario. The hyperparameters in this thesis were chosen by continued experimentation and taking those values that showed good results, but this cannot be done on real datasets and real applications. The hyperparameters depend on the characteristics of the time series analysed, new datasets might require new hyperparameter values. For this reason a more scientific methodology should be found to set the FBN hyperparameters.

5. *FBN performance at different band of frequencies.* Section 6.5.3 covered in part the FBN property for inferring networks at user's defined frequency bands and the advantage of decreasing the number of computations at the algorithm level. Choosing a limited band of frequencies instead of the full Nyquist range reduces the size of the matrices used for spectral regression. Nevertheless a study about how FBNs behave at different frequency bands should be done in the future. In Electrophysiology, frequency band analysis is a very important topic since the bands such as alpha, beta, theta and gamma, are related with physical or cognitive conditions in the human brain.

6. *Realistic brain model for MEG-beamforming simulations.* For the simulations in this thesis a MEG simulated acquisition environment was implemented. The simulated environment is able to simulate the linear source mixing caused by the separation between the brain sources and MEG SQUIDs where only the

magnetic fields from the primary currents of modelled dipoles were recorded. Nevertheless in a real brain the secondary currents which travel through the head's flesh are also important to create more realistic simulations. Future work should also focus on the implementation of a brain model such as the sphere model, the multisphere model for the brain or to model completely the subject's head and tissues using finite element method, see for instance Huang et al. (1999); Darvas et al. (2004).

Realistic head models like the ones obtained by the finite element method are able to model the shape of the cortex. For the case of brain connectivity this is advantageous since it allows to define the direction of the current dipole which otherwise should be estimated by maximum variance, for a review on this see Fuchs (2007).

7. *Inference of DMN regions.* For the experiments in this thesis, regions that are currently known to belong to the DMN system were used for network connectivity inference. The experiments were based on the work of Jiao et al. (2011). Nevertheless, for the resting state database in the YNiC the existence of a DMN system and its regions should be proved first. Brookes et al. (2011b) and de Pasquale et al. (2010) published a methodology for the inference of the DMN from MEG recordings. It will be necessary and interesting to reproduce the results in Brookes et al. (2011b) using the MTBI MEG database.

8. *Prior distributions and network probabilities.* Since this thesis is a Bayesian approach, the prior distribution takes an important role. In this thesis, the prior can be seen as a probability distribution function that gives high probabilities to sparsely connected networks and low probabilities to highly connected ones. This is done under the supposition that the expected network is simple and concise or in other words sparse. This sparsity is supposed to be the previous knowledge. Nevertheless different approaches can be added to the prior if other interaction models, or assumptions for the network structure and its parameters are included. This topic will also be interesting for future research.

9. *Filtering or no filtering?*. FBNs are able to infer networks at user's defined frequency bands, hence analysing the coefficients at the desired band of frequencies might count as a kind of filtering. In the simulations this was not tested, neither for experiments using the real MEG datasets. Certainly a filtering or prefiltering will affect the spectral regression in the likelihood power function. Experiments and research in this area will also be interesting in the near future.

10. *Wavelets and other orthogonal signal decompositions.* FBNs as their name indicates uses the Fourier transform to perform the signal decomposition in the spectral domain and separate this information from the phase. But mathematically there is no constraint against applying other kind of orthogonal decomposition for the signals of interest. One suggestion for future work is to apply wavelets and perform wavelet spectral regression and phase analysis. Wavelets have interesting characteristics relating to resolution and smoothness in spectral estimation.

11. *Number of subjects in the study.* The MTBI database at hand is composed of 27 recordings (controls and MTBI). For the experiments in this thesis only the recordings from the control subject R2016 were analysed. It will be necessary once the previous points in this section are covered, to study all recordings in the database to relate the MTBI condition with the DMN system for clinical applications.

Network analysis and inference in the spectral domain as proposed in this document shows to be a potential approach for network inference studies and for brain connectivity studies. FBNs are a novel technique born in the Electrophysiology field and designed for network inference from MEG time series. This spectral approach has also raised importance in the research community, see for instance a recent published PhD thesis on the same field in Meng (2011). Nevertheless to the knowledge of the author of this document, Peraza and Halliday (2010b) is the first publication suggesting this approach for network structure inference.

Appendix A

Implementation details of FBNs

In this Section details related to the implementation of the simulated annealing algorithm are explained. The algorithm aims to sample networks from the network probability distribution $p(S_h|\mathbf{D})$ which is then transformed to $p(S_h|\mathbf{F}, \mathbf{\Theta})$. From Section 5.9.3 recall the acceptance ratio for the simulated annealing algorithm in Eq 5.104,

$$\beta = \min\left[1, \left(\frac{p(\mathbf{F}|S_h^+)p(\mathbf{\Theta}|S_h^+)p(S_h^+)}{p(\mathbf{F}|S_h)p(\mathbf{\Theta}|S_h)p(S_h)}\right)^{1/T(t)}\right], \tag{A.1}$$

where

$$T(t) = \max\left[T_0\left(\frac{T_f}{T_0}\right)^{t/b}, T_f\right], \tag{A.2}$$

The behaviour of the simulated annealing algorithm and in specific the implementation in this thesis is governed by seven hyperparameters:

- **numruns**: Number of runs. The maximum number of iterations for the MCMC algorithm.
- **initemp**: Initial temperature. According to the simulated annealing, this parameters governs the acceptance ratio at the start of the network searching. It is defined by T_0 in Eq. A.2.
- **fintemp**: Final temperature. This parameters governs the acceptance ratio at the end of the network searching algorithm and its influence starts at time

b. This parameter is defined as T_f in Eq. A.2.

- **coolruns**: Cooling runs. This is the number of iterations to reach the final temperature T_f in Eq. A.2. This hyperparameter is represented in Eq A.2 as b.

- **burnedruns**: Burned runs. The number of iterations that are assumed to be stable after algorithm reaches T_f. For this implementation "burnedruns" will be the last iterations before the algorithm reaches "numruns".

- **alpha**: Network complexity. This tuning parameter punishes network that are densely connected with the intention of favouring network models that are simpler.

- **beta**: Phase influence. This tuning parameter controls the influence of the network phase probability distribution

Adding the hyperparameters to the simulated annealing algorithm, the likelihood distributions in Eq. A.1 is

$$p(\mathbf{F}|S_h)p(\mathbf{\Theta}|S_h)p(S_h) \equiv p(\mathbf{F}|S_h)p(S_h)^{alpha}p(\mathbf{\Theta}|S_h)^{beta} \ , \tag{A.3}$$

where the network complexity *alpha* and the phase influence *beta* parameters have been added. Notice that these parameter are included as power in order to leave them as factors when the logarithm is computed. For this case the $p(\mathbf{F}|S_h)p(S_h)^{alpha}$ term is equivalent to the Fourier Bayesian information criterion (FBIC) in Section 5.7. Hence the previous definition might also be written as

$$\ln(p(\mathbf{F}|S_h)p(\mathbf{\Theta}|S_h)p(S_h)) \equiv \ln p(\mathbf{F}|S_h) + alpha \times \ln p(S_h) + beta \times \ln p(\mathbf{\Theta}|S_h) \ , \tag{A.4}$$

$$\ln(p(\mathbf{F}|S_h)p(\mathbf{\Theta}|S_h)p(S_h)) \equiv FBIC(\mathbf{F}|S_h, alpha) + beta \times \ln p(\mathbf{\Theta}|S_h) \ . \tag{A.5}$$

Recall that FBIC is independent from the phase. The error spectrum $f_{\epsilon\epsilon}$ is a real number which is not influenced by the argument of the Fourier coefficients. Eq. A.5 is the function used for the Bayesian ratio and the simulated annealing algorithm in Chapter 6 and Chapter 7.

Appendix B

Colour images

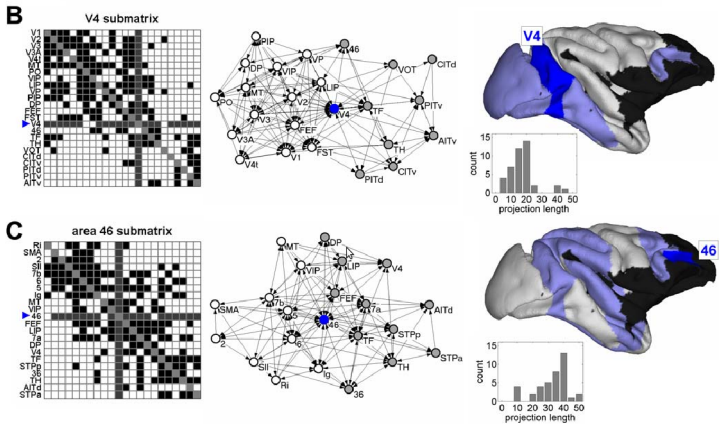

Figure B.1: Macaque cortex. Regions V4 and 46 are hubs that communicate two important clusters of nodes. In the left column the connectivity matrices are shown. The middle column shows the network where hubs 46 and V4 are highlighted in blue and the two regions they communicate in white and gray respectively. The third column shows the cortex surfaces with regions V4 and 46 shaded in blue and their direct neighbours shaded in light blue. Image from Sporns et al. (2007).

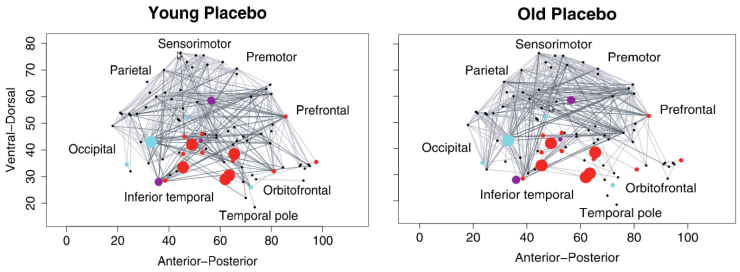

Figure B.2: Brain functional networks of a young person and old person following placebo. The node efficiency is colour coded: The red nodes have a reduced efficiency by age, blue nodes by sulpiride (a dopamine antagonist), and purple nodes by both, age and sulpiride. Image from Achard and Bullmore (2007).

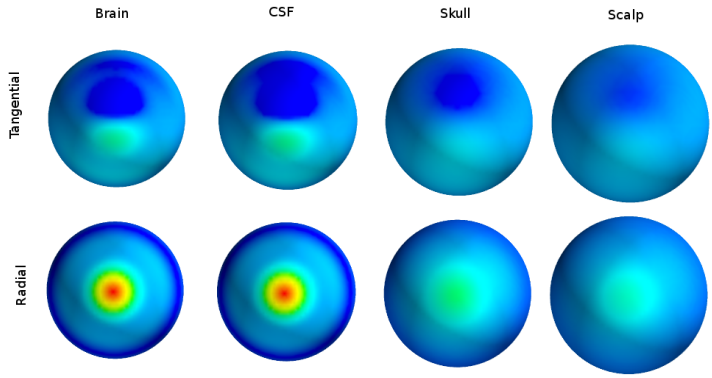

Figure B.3: The four sphere model for volume conduction. The top row shows the four spheres representing the cortex, the CSF, skull, and the scalp when a tangential dipole is active. The bottom row shows the same but for a radial dipole. Notice that surface voltage is more concentrated for radial (also called normal) dipoles.

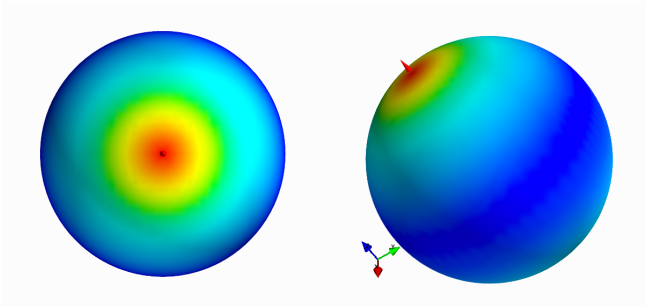

Figure B.4: Orthogonal dipole on head's scalp. An orthogonal dipole is simulated using the four sphere model (Nunez and Srinivasan , 2006). The electric dipole was placed inside the brain. The figure shows the voltage field produced on the head's scalp, which is shown on the outermost sphere (recall the four sphere model). Notice how the fields maxima is located just above the current dipole, whose position is represented by a red cone above the scalp.

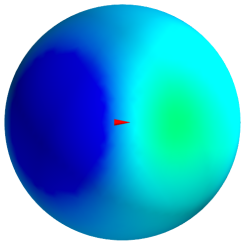

Figure B.5: Tangential dipole on head's scalp. As in Figure 2.3, a current dipole was place in the brain but in this case its direction is tangential to the scalp cortex. The position of the dipole inside the four sphere model is shown with a red cone with its vertex pointing to the positive pole. Notice that in this case, the field's distribution is not concentrated as in the orthogonal case. The field is much wider and its voltage lower. Also, the field shows one minimum and one maximum and the real position of the dipole is located in the middle of these two points. This pattern is characteristic of tangential dipoles.

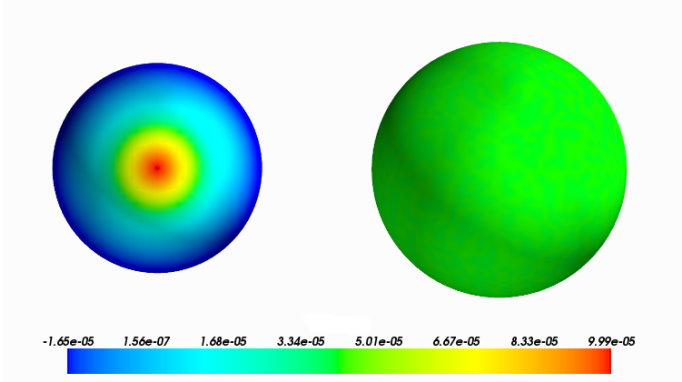

Figure B.6: Magnetic field generated by an orthogonal dipole. The sphere at the right shows in colour the magnitude of the orthogonal components of the magnetic field that crosses the sphere. The sphere's surface is located four centimetres above the scalp sphere. Notice that the magnitude shown is zero (green colour), meaning that there are no orthogonal components.

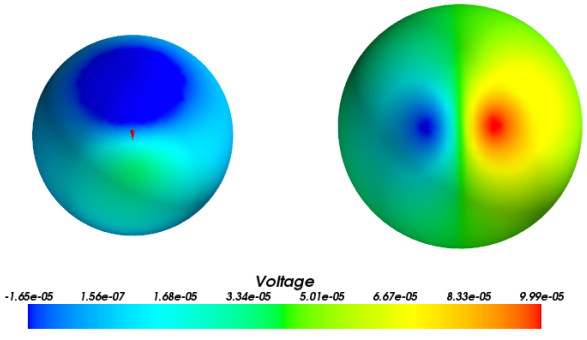

Figure B.7: Magnetic field generated by a tangential dipole. The sphere at the right shows in colour the magnitude of the orthogonal components of the magnetic field that crosses the sphere. Notice that in this case a minimum and a maximum can be appreciated, and that the real position of the dipole is in the middle of these two points.

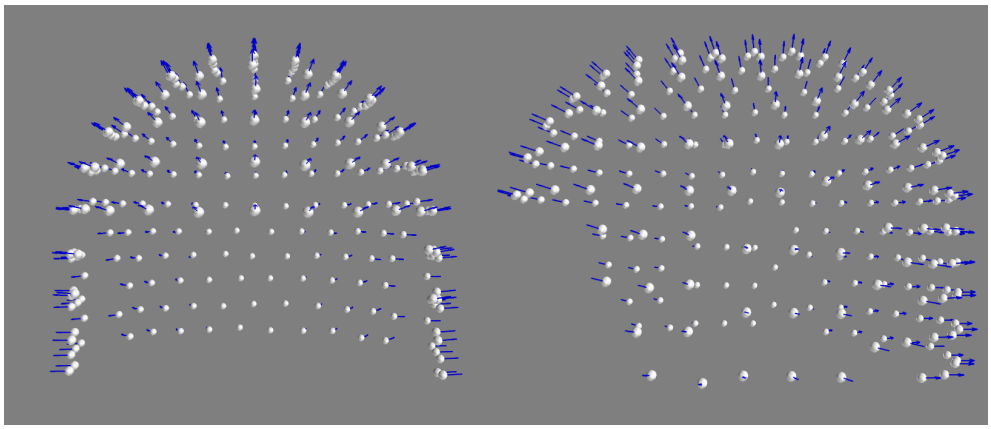

Figure B.8: 248 SQUID layout from a 4D Neuroimaging MEG device. The blue arrows represent the SQUIDs' directions. Hence the SQUIDs are more sensitive to fields parallel to the blue arrows.

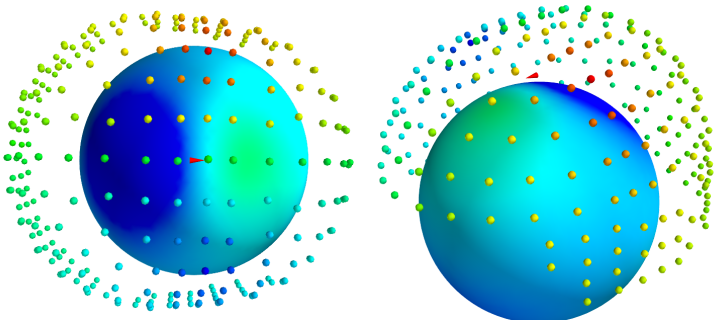

Figure B.9: Magnetic field acquisition given by the MEG SQUIDs. Here the coloured smaller 248 spheres represent the magnitude of the acquired magnetic field using SQUIDs (one sphere per SQUID). This simulation was done by applying Eq. 2.9 to sense a tangential dipole. Also the scalp's potential distribution is shown using the four sphere model for EEG.

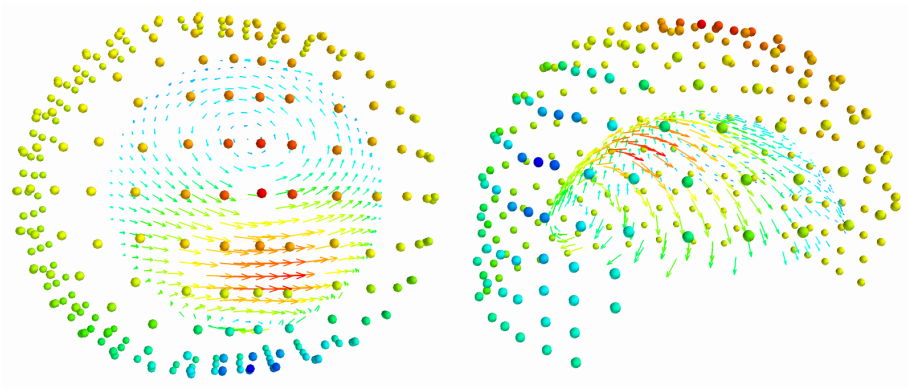

Figure B.10: Minimum norm source estimation. For this simulation the surface of the scalp was tessellated to create regions of interest. Then, dipoles were estimated using regularised least squares or minimum norm. The red arrows indicate the most likely place where the dipole might be. This is the best dipole that explain the brain magnetic field whose magnitude is shown in coloured spheres.

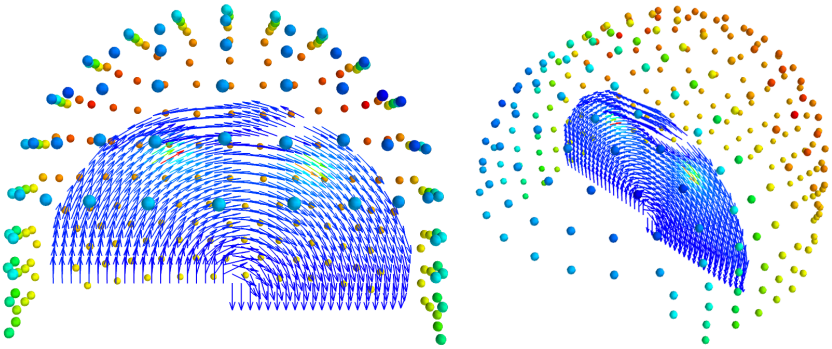

Figure B.11: Beamforming simulation, localization of two dipoles. The LCMV beamforming finds the position of two tangential dipoles. Using a sphere to define the universe of possible places, a coronal slice was tessellated to define the regions of interest. Then, a dipole was estimated at each region. These dipoles are represented as coloured arrows being the red ones the best estimations.

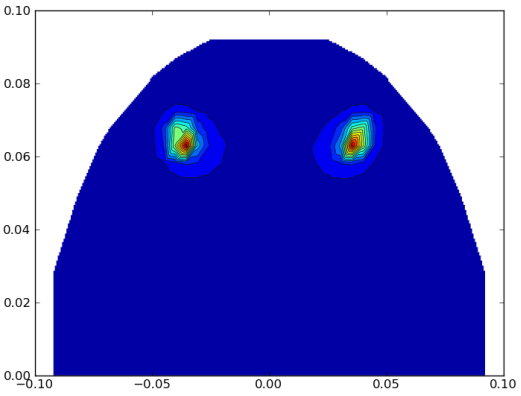

Figure B.12: Same simulation as in Figure 2.12. Here the NAI index are shown using a heat map with spline interpolation. Notice that the LCMV beamforming can estimate accurately sources at deeper regions of the brain.

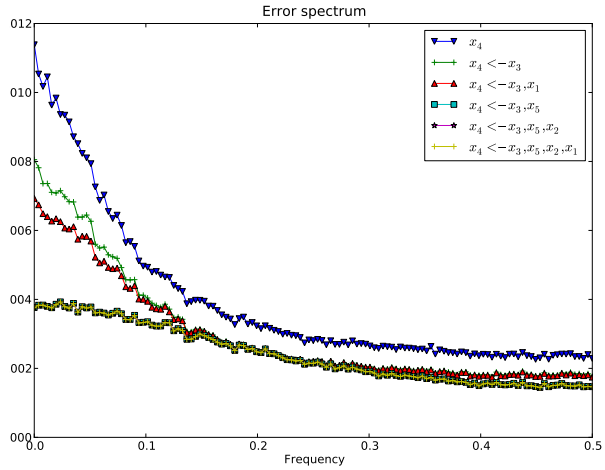

Figure B.13: The error spectrum is shown for networks formed by node x_4 and different combinations of parents. It is possible to see how the addition of a new source node in the parent vector decreases the error spectrum until the two neighbouring nodes of x_4 are included, these are x_3 and x_5. Once these nodes are given, any information coming from other sources in the network is irrelevant. This shows that x_4 is conditional independent from the rest of the network given x_3 and x_5.

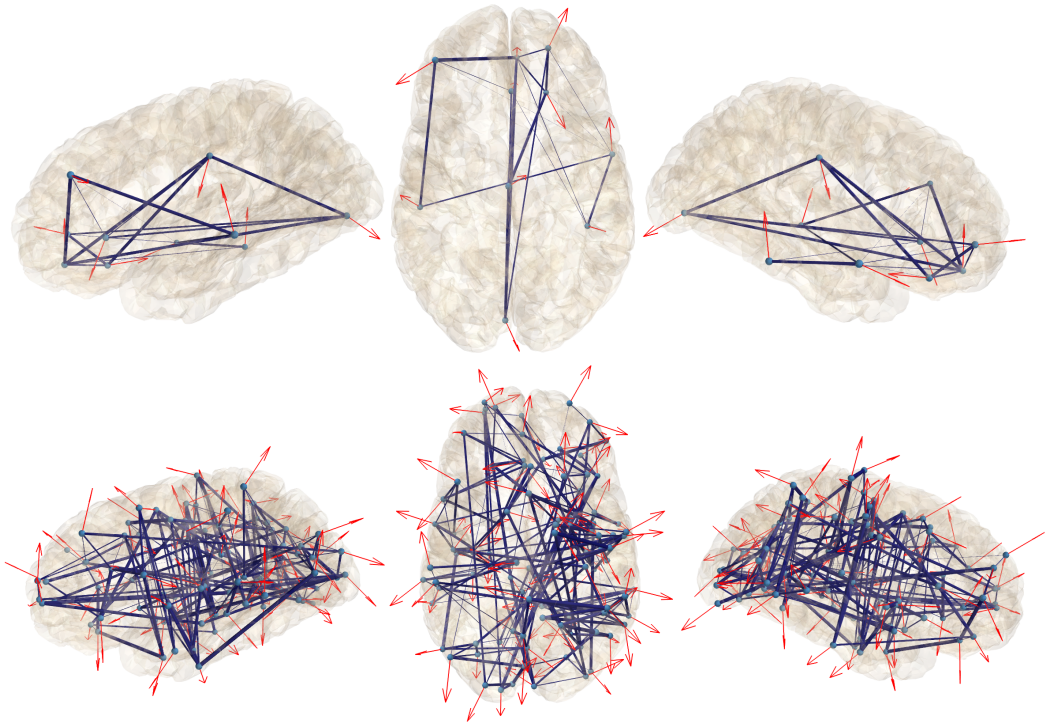

Figure B.14: Generated network structures in the brain cortex mesh for the network structure inference experiments using the MVAR(d) model. The upper row shows the 10-node network and the 100-node network is shown in bottom row.

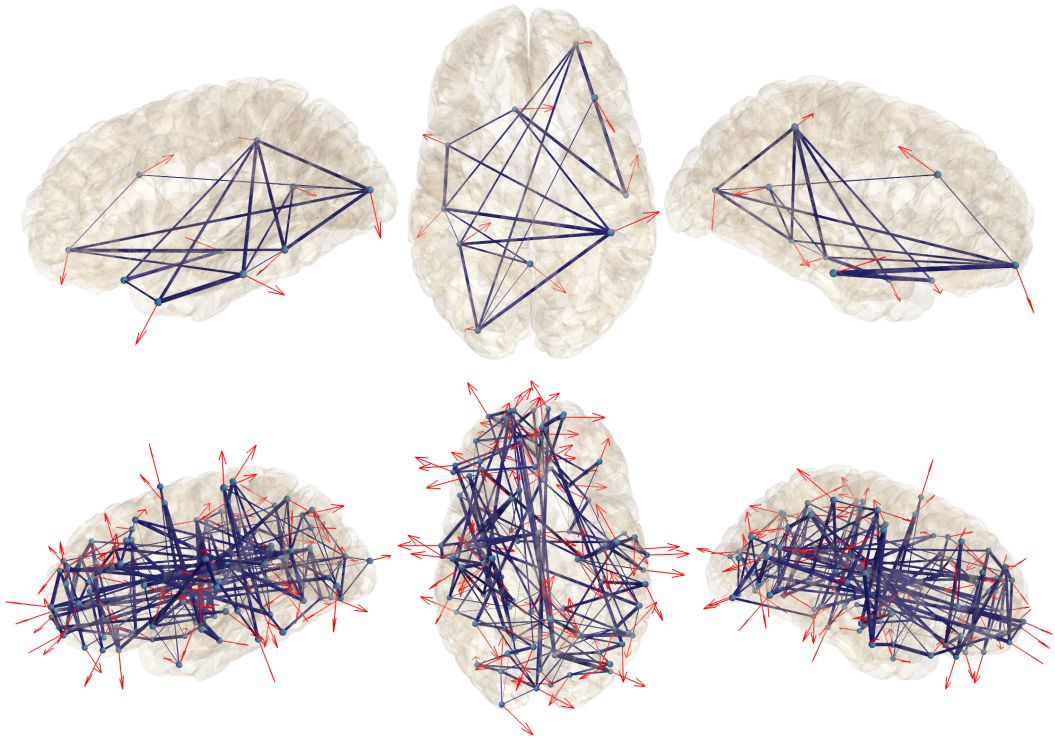

Figure B.15: Generated network structures in the brain cortex mesh for network structure inference experiments using the NL(d) model. The upper row shows the 10-node network and the 100-node network is shown at the bottom row.

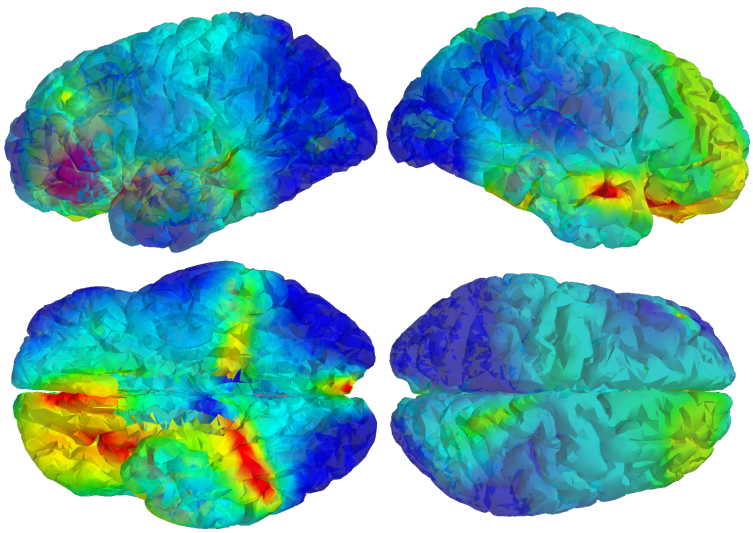

Figure B.16: Beamforming maps from independent sources. The current dipoles were modulated by independent Gaussian noise sources and the source arrangement is the one of the 10-node linear network in Figure 6.20.

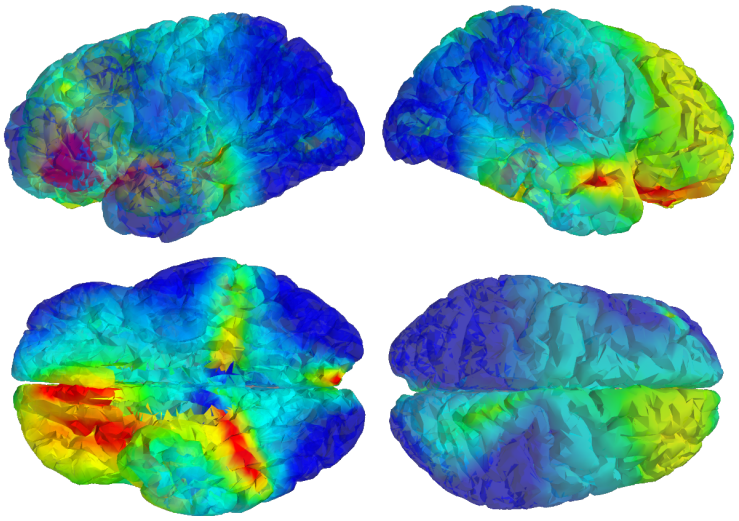

Figure B.17: Beamforming maps using the 10-node MVAR(1) brain network shown in Section 6.20.

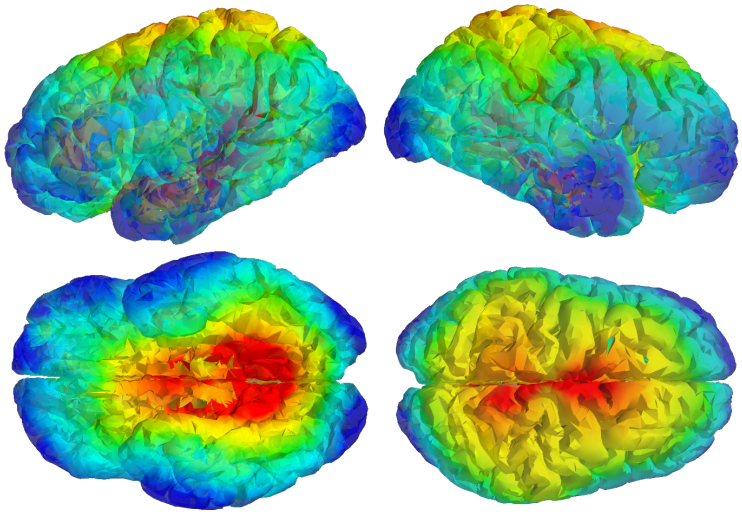

Figure B.18: Beamforming map using the 100-node MVAR(1) network model shown in Figure 6.20.

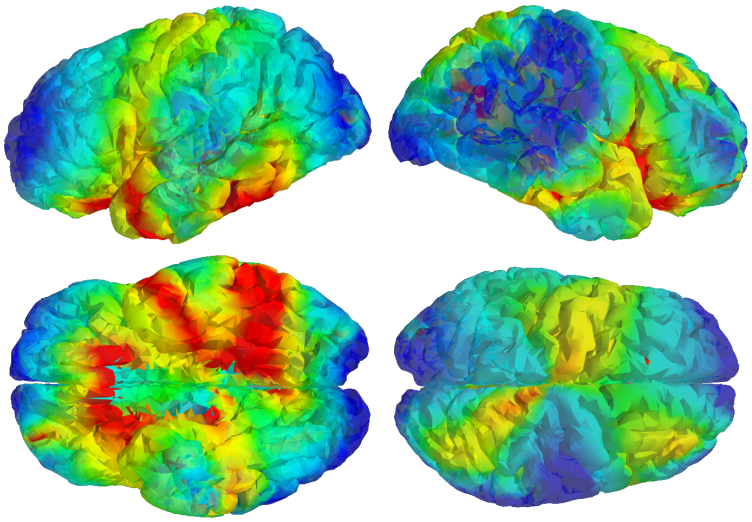

Figure B.19: Beamforming maps using the 10-node NL(1) network. The original network structure system is depicted in Figure 6.21.

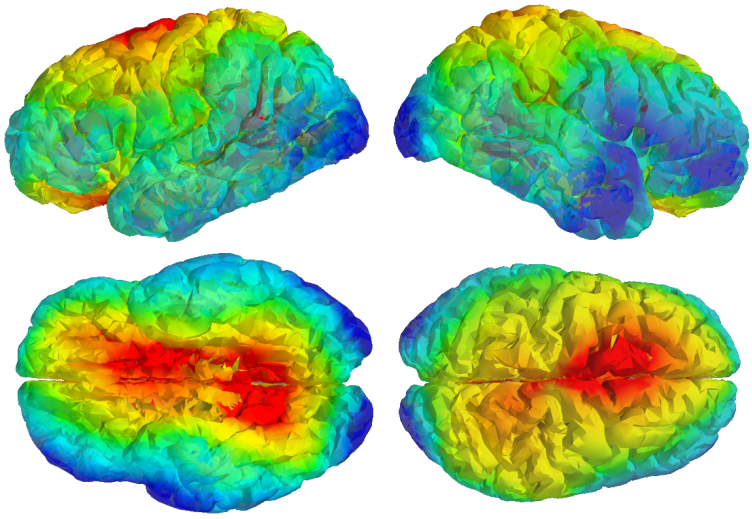

Figure B.20: Beamforming map for the 100-node NL(1) network. The original network structure systems are depicted in Figure 6.21.

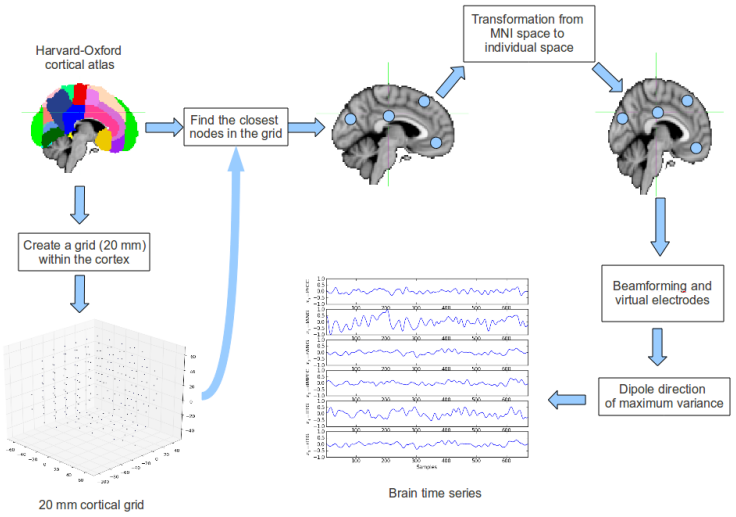

Figure B.21: Preprocessing steps for the experiments in this chapter.

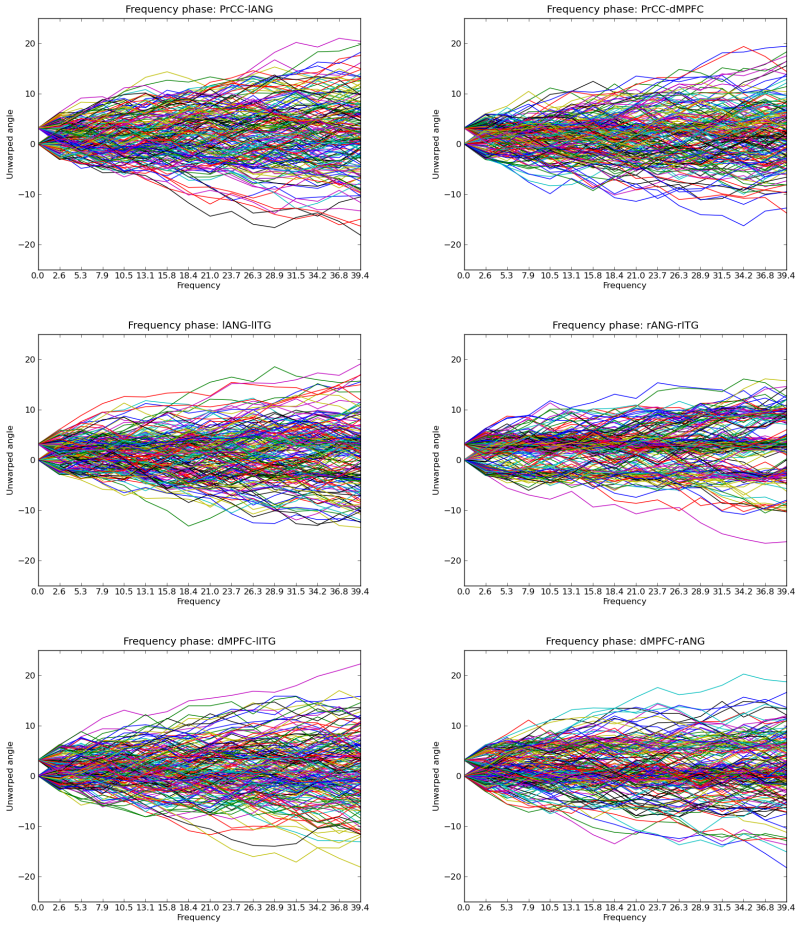

Figure B.22: Cross-phase frequency for six edges of interest between 2 and 40 Hz. The edges are from top to bottom and from left to right: PrCC-lANG, PrCC-dMPFC, lANG-lITG, rANG-rITG, dMPFC-lITG, and dMPFC-rANG.

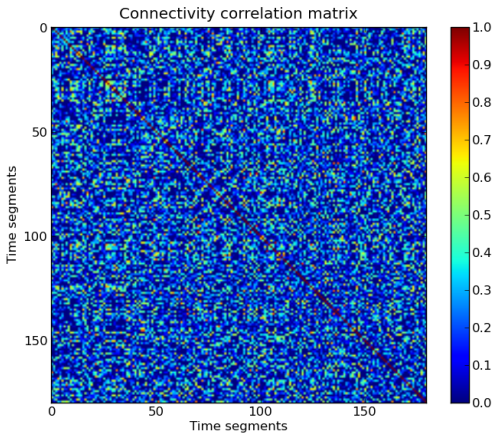

Figure B.23: Correlation matrix of the connectivity vectors. The 180 acyclic directed estimated matrices were reshaped from 6×6 to 36 column vector. Then, correlation among all of them is computed.

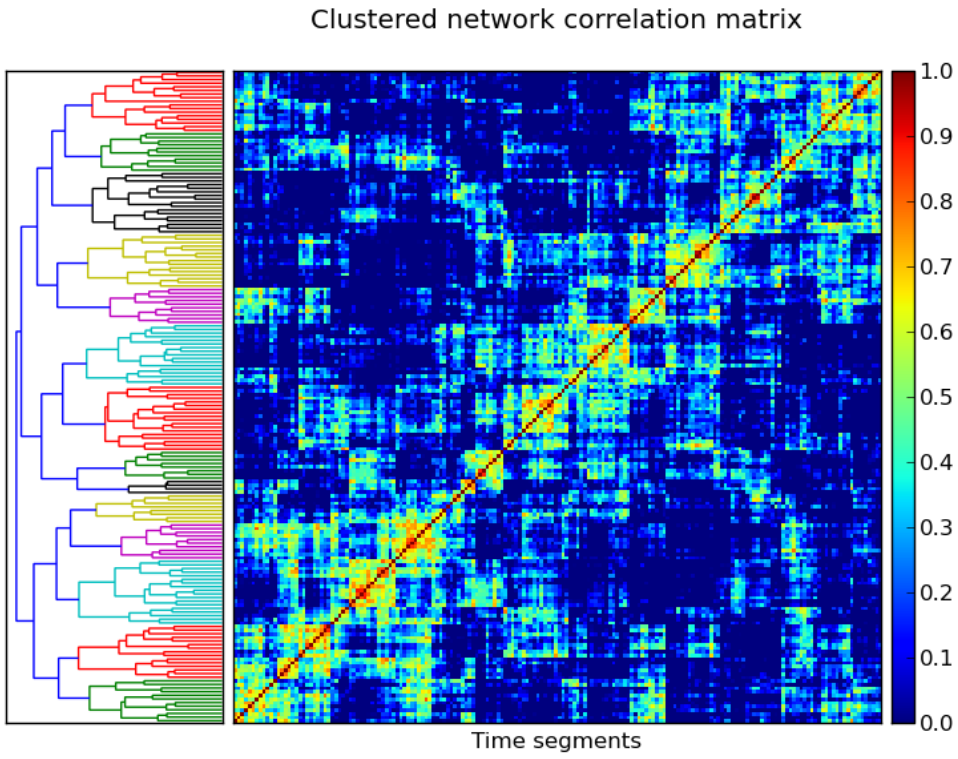

Figure B.24: Clustered network correlation matrix. The hierarchical clustering is shown by the dendogram graph which shows approximately 13 network clusters.

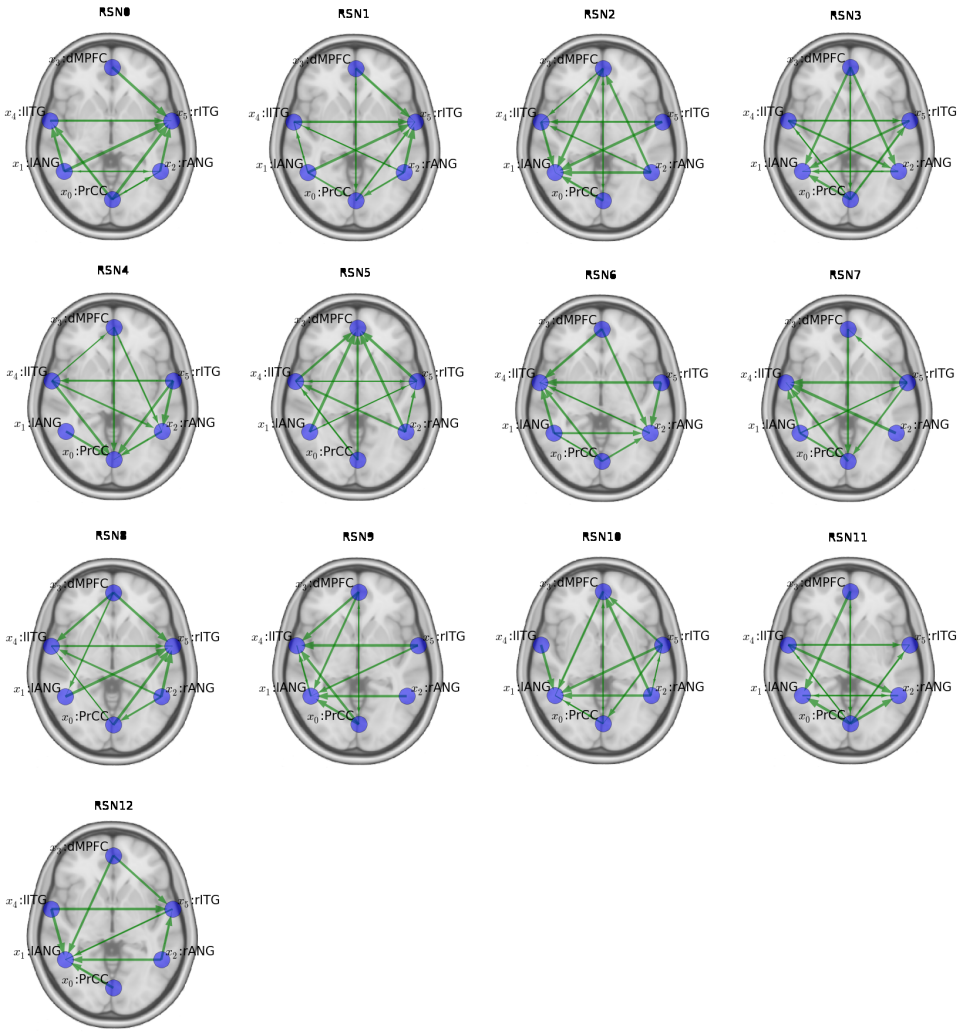

Figure B.25: Network patterns for the resting state MEG recording. This figure shows the 13 identified networks patterns that are estimated by the FBN method. The patterns were found using a hierarchical clustering method and then each cluster is averaged to obtain a representative network of each cluster.

Bibliography

Abeles M,. *Corticonics, neural circuits of the cerebral cortex*. Cambridge university press, 1991.

Achard S., Bullmore E., *Efficiency and cost of economical brain functional networks*. PLoS Comput Biol, no. 3, vol. 2, e17, 2007.

Anderson T. W., *An introduction to multivariate statistical analysis*. Wiley, 2003.

Baccala L. A., Sameshima K., *Partial directed coherence: a new concept in neural structure determination*, Biol. Cybern., vol. 84, pp. 463-474, 2001.

Basset D.S. and Bullmore E., *Small-world brain networks*, The Neuroscientist, vol. 12, no. 6, pp. 512-523, 2006.

Baillet S., Mosher J. C., Leahy, R. M., *Electromagnetic brain mapping*, IEEE Signal Processing Magazine, Nov. 2001.

Bear M. F., Connors B. W., Paradiso M. A., *Neuroscience, exploring the brain*, Lippincott Williams & Wilkins, 2007.

Biswal B. B., et al, *Toward discovery science of human brain function*, PNAS, vol. 107, no. 10, 2010.

Borgiotti G. V., Kaplan L. J., *Superresolution of uncorrelated interference sources by using adaptive array technique*, IEEE Transactions on Antennas and Propagation, vol. AP-27, no. 6, Nov. 1979.

Brillinger D. R., *Times series, data analysis and theory*. Holden-day, Inc. 1981.

Brookes M. J., Gibson A. M., Hall S. D., Furlong P. L., Barnes G. R., Hillebrand A., Singh K. D., Holliday I. E., Francis S. T., Morris P. G., *A general linear model for MEG beamformer imaging*, Neuroimage, vol. 23, pp. 936-946, 2004.

Brookes M. J., Wood J. R., Stevenson C. M., Zummer J. M., White T. P., Liddle P. F., Morris P. G., *Changes in brain network activity during working memory tasks: A magnetoencephalographic study*, NeuroImage, vol. 55, pp. 18041815, 2011.

Brookes M. J., Hale J. R., Zumer J. M., Stevenson C. M., Francis S. T., Barnes G. R., Owen J. P., Morris P. G., Nagarajan S. S., *Measuring functional connectivity using MEG: Methodology and comparison with fcMRI.* Neuroimage 56, 1082-1104, 2011.

Buckner R. L., Andrews-Hanna J. R., Schacter D. L., *The brain's default network, anatomy, function and relevance to disease*, Ann. N. Y. Acad. Sci., vol. 1124, pp. 1-38, 2008.

Buckley K. M., *Spatial/Spectral filtering with linearly constrained minimum variance beamformers*, IEEE Transactions On Acoustics, Speech, and Signal Processing, vol. ASSP-35, no. 3, Mar. 1987.

Bullmore E., Sporns O., *Complex brain networks: graph theoretical analysis of structural and functional systems*, Nature, no. 10, 2009.

Chen M., Shao Q., Ibrahim J.G., *Monte Carlo methods in Bayesian computations*, Springer series in Statistics, Springer, 2000.

Chen X., Anantha G., Lin X., *Improving Bayesian network structure learning with mutual information-based node ordering in the K2 Algorithm*, IEEE Trans. Knowledge and Data Eng., vol. 20, no. 5, May 2008.

Cooper G., Herskovits E., *A Bayesian Method for the induction of probabilistic networks from data*, Machine Learning, vol. 9, pp. 309-347, 1992.

Dalal S. S., Zumer J. M., Agrawal V., Hild K. E., Sekihara K., Nagarajan S. S., *NUTMEG: A neuromagnetic source reconstruction toolbox*, Neurol Clin Neurophysiol., vol. 52, 2004.

Darvas, F., Pantazis, D., Leahy, R. M., *Mapping human brain function with MEG and EEG : methods and validation*, NeuroImage, 23, 289-299. (2004).

de Pasquale, F., Della Penna, S., Snyder, A. Z., Lewis, C., Mantini, D., Marzetti, L., Belardinelli, P., et al., *Temporal dynamics of spontaneous MEG activity in brain networks*. Proceedings of the National Academy of Sciences of the United States of America, 107(13), 6040-5, (2010).

de Vico Fallani F., Astolfi L., Cincotti F., Mattia D., la Rocca D., Maksuti E., Salinari S., Babiloni F., Vegso B., Kozmann G., Nagy Z., *Evaluation of the brain network organization from EEG Signals: A preliminary evidence in stroke patient*, The Anatomical Record, vol. 292, pp. 20232031, 2009.

Fair D.A., Cohen A.L., Power J.D., Dosenbach N.U.F., Church J.A., et al., *Functional brain networks develop from a local to distributed organization*, PLoS Comput Biol, no 5, vol. 5, pp. e1000381, 2009,

Finger S., *Origins of neuroscience, A history of explorations into brain function*, Oxford university press, 1994.

Fuchs A. *Beamforming and its applications to brain connectivity*, in handbook of brain activity, Jirsa V.K., McIntosh A.R., eds., Springer Verlag, Berlin 2007.

Gamerman D., Lopes H.F., *Markov chain Monte Carlo, stochastic simulation for Bayesian inference*, Chapman & Hall/CRC, 2006.

Gazzaniga M. S., *Cognitive Neuroscience, the biology of the mind*, Norton, 1998.

Geiger D., Heckerman D., "Learning Gaussian networks," Technical Report, Microsoft Research, Mar. 1994.

Granger C. W. J., *Investigating causal relations by econometric models and cross-spectral methods*, Econometrica, vol. 37, no. 3, Jul. 1969.

Gross J., Kujala J., Hämäläinen M., Timmermann L., Schnitzler A., *Dynamic imaging of coherent sources: Studying neural interactions in the human brain*, Proc. Natl. Acad. Sci. USA, 98(2): 694-699, 2001.

Guevara R., Perez Velazquez J.L., Nenadovic V., Wennberg R., Senjanovic G., Dominguez, L. G., *Phase synchronization measurements using electroencephalographic recordings What can we really say about neuronal synchrony?*, Neuroinformatics, vol. 3, no. 4, pp. 301-14, 2005.

Hagmann P., Cammoun L., Gigandet X., Meuli R., Honey C.J., et al., *Mapping the structural core of human cerebral cortex*, PLoS Biol, no 6, vol 7, pp. e159, 2008.

Halliday D. M., Rosenberg J.R., Amjad A.M., Breeze P., Conway B.A., Farmer S.F., *A framework for the analysis of mixed time series/point process data - Theory and application to the study of physiological tremor, single motor unit discharges and electromyograms*. Progress in Biophysics and molecular Biology, vol. 64, pp. 237-278, 1995.

Hämäläinen M., Hari R., Ilmoniemi R.J., Knuutila J., Lounasmaa O. V., *Magnetoencephalography - theory, instrumentation, and applications to noninvasive studies of the working human brain*. Reviews of Modern Physics, no. 65, 1993.

Hämäläinen M. S., Ilmoniemi R. J., *Interpreting magnetic fields of the brain: minimum norm estimates*, Med. & Biol. Eng. & Comput., vol. 32, pp. 35-42, 1994.

Haykin S., *Adaptive filter theory*. Prentice Hall, 2001.

He Y., Chen Z., Gong G., Evans A., *Neuronal networks in Alzheimer's disease*, The Neuroscientistis, vol. 15, no. 4, pp. 333-350, 2009.

Heckerman D. *A tutorial on learning with Bayesian networks*. Technical report, Microsoft Research, 1996.

Heckerman D., et. al. *Learning Bayesian networks: The combination of knowledge and statistical data*. Machine Learning. 20, 197-243, 1995.

Herskovits, E., Cooper, G. *Kutató: entropy-driven system for construction of probability expert systems from databases* Proceedings of the sixth conference on uncertainty in AI (pp. 54-62) Cambridge, MA.

Huang M. X., Mosher J. C., Leahy R. M. *A sensor-weighted overlapping-sphere head model and exhaustive head model comparison for MEG*, Phys Med Biol, 44, 423-440, 1999.

Huang M-X, Shihˆ J. J., Lee R. R., Harrington D. L., Thoma R. J., Weisend M. P., Hanlon F., Paulson K. M., Li T., Martin K., Miller G. A., Canive J. M., *Commonalities and differences among vectorized beamformers in electromagnetic source imaging*, Brain Topography, vol. 16, no. 3, Spring 2004.

Huettel S. A., Song A. W., McCarthy G., *Functional magnetic resonance imaging*. Sinahuer, 2009.

Hui H. B., Pantazis D., Bressler S. L., Leahy R. M., *Identifying true cortical interactions in MEG using the nulling beamformer*, Neuroimage, 49(4), 2010.

Jeffs B., Leahy R., Singh M., *An evaluation of methods for neuromagnetic image reconstruction*, IEEE Transactions On Biomedical Engineering, vol. BME-34, no. 9, Sept. 1987.

Jensen F.V., Nielsen T.D., *Bayesian networks and decision graphs*, Second Edition, Springer, 2007.

Jiao Q., Lu G., Zhang Z., Zhong Y., Wang Z., Guo Y., Li K., Ding M., Liu Y., *Granger causal influence predicts BOLD activity levels in the default mode network*. Human Brain Mapping 32:154-161, 2011.

Kaiser M., Hilgetag C.C., *Nonoptimal component placement, but short processing paths, due to long-distance projections in neural systems* . PLoS Comput Biol, no. 2, vol. 7, pp. e95, 2006.

Kaminski M., Ding M., Truccolo W.A., Bressler S.L., *Evaluating causal relations in neural systems: Granger causality, directed transfer function and statistical assessment of significance*, Biol. Cybern., vol. 85, pp. 145-157, 2001.

Kay S. M., *Modern spectral estimation*. Prentice Hall, 1988.

Kirkpatrick S., Gelatt C.D. Jr., Vecchi M.P., *Optimization by Simulated Annealing*, Science, vol. 220, no. 4598, pp. 671-680, May 1983.

Lauglin S. B., *Energy as a constraint on the coding and processing of sensory information*, Current Opinion in Neurobiology, vol. 11, no. 4, pp. 475-480, 2001.

León-Garcia A., *Probability and random processes for electrical engineering*. Addison Wesley, 1994.

Malmivuo J., Plonsey R., *Bioelectromagnetism*, Oxford university press, 1995.

Malmivuo J., Suihko V., Eskola H., *Sensitivity distributions of EEG and MEG measurements*, IEEE Transactions On Biomedical Engineering, vol. 44, no. 33, Mar. 1997.

Meng X., *Spectral Bayesian Network and Spectral Connectivity Analysis for Functional Magnetic Resonance Imaging Studies*, PhD Thesis, University of Cincinnati, 2011.

Metropolis N., Ulam S., *The Monte Carlo method*, J. Amer. Statist. Assoc. 44: 335341, 1949.

Metropolis N., Rosenbluth A. W., Rosenbluth M. N., Teller A., Teller H., *Equations of state calculations by fast computing machines*, Journal of Chemical Physics 21: 10871091, 1953.

Mormann F., Lehnertz K., David P., Elger C.E., *Mean phase coherence as a measure for phase synchronization and its application to the EEG of epilepsy patients*, Physica D, vol. 144. no. 3-4), pp. 358-69, 2000.

Mosher J. C., Lewis P. S., Leahy R. M., *Multiple dipole modeling and localization from spatio-temporal MEG data*, IEEE Transactions on Biomedical Engineering, vol. 39, no. 6, Jun. 1992.

Nakamura T., Hillary F. G., Biswal B.B., *Resting network plasticity following brain injury*, PLoS ONE, no. 4, vol. 12, pp. e8220, 2009.

Neapolitan R. E. *Learning Bayesian networks*. Prentice Hall, 2004.

Newman M.E.J., *Networks: An introduction*, Oxford, 2010.

Niedermeyer E., Lopes F. L., *Electroencephalography, basic principles, clinical applications, and related fields*, Williams & Wilkins, 1993.

Nolte G., Bai O., Wheaton L., Mari Z., Vorbach S., Hallett M., *Identifying true brain interaction from EEG data using the imaginary part of coherency*. Clinical Neurophysiology, vol. 115, pp. 2292-2307, 2004.

Nunez, P. L., Srinivasan, R., *Electric fields of the brain - The neurophysics of EEG*, Second Edition, Oxford University Press, 2006.

Oostenveld R., Fries P., Maris E., Schoffelen J., *Fieldtrip: Open source software for advanced analysis of MEG, EEG and invasive electrophysiological data*, Computational Intelligence and Neuroscience, doi:10.1155/2011/156869, 2011.

Pearl J., Russel S., *Bayesian Networks*, Handbook of brain theory and neural networks (M. Arbib (Ed.)), MIT Press, 2001.

Penny W. D., Friston K., Ashburner J. T., Kiebel S. J., Nichols T. E.; Editors. *Statistical parametric mapping: The analysis of functional brain images*. Elsevier, 2007.

Peraza L.R., Asghar A., Green G., Halliday D.M., *Volume conduction effects in brain network inference from electroencephalographic recordings using phase lag index*, submitted to Neuroscience Methods, 2012.

Peraza L.R. and Halliday D.M., *A fast dynamic Bayesian network algorithm for structure learning using a time-lagged partia correlation matrix*, IEEE International Conference on Signals and Electronic Systems, Poland, Sept 7-10, 2010.

Peraza L.R. and Halliday D.M., *Fourier Bayesian information criterion for network structure and causality estimation*, IEEE International Conference on Signals and Electronic Systems, Poland, Sept 7-10, 2010.

Preissl H. (editor). *Magnetoencephalography*. Academic press, 2005.

Priestley M.B., Spectral Analysis and Time Series. Volumes I and II [paperback], Academic Press, 1983.

Purves D., Augustine G. J., Fitzpatrick D., Hall W. C., LaMantia A., McNamara J. O., Williams S. (Editors), *Neuroscience*, 3rd edition, Sinauer Associates Inc, 2004.

Quiroga R.Q., Kraskov A., Kreuz T., Grassberger P., *Performance of different synchronization measures in real data: A case study on electroencephalographic signals*, Phys. Rev. E, vol. 65, no. 4, pp. 041903, 2002.

Quiroga R. Q., Panzeri S., *Extracting information from neuronal populations: information theory and decoding approaches*. Nature Reviews Neuroscience, vol. 10, pp. 173-185, 2009.

Raichle M. E., MacLeod A. M., Snyder A. Z., Powers W. J., Gusnard D. A., Shulman G. L., *A default mode of brain function*, Proc. Natl. Acad. Sci. U.S.A., vol. 98, no. 2, pp. 676-682, 2001.

Raichle M. E., Snyder A. Z., *A default mode of brain function: A brief history of an evolving idea*, NeuroImage, no. 37, pp. 1083-1090, 2005.

Rajapakse J.C., Zhou J., *Learning effective brain connectivity with dynamic Bayesian networks*, Neuroimage, vol. 37, 2007.

Rosemberg J. R., Amjad A.M., Breeze P., Brillinger D.R., Halliday D.M., *The fourier approach to the identification of functional coupling between neuronal spike trains*. Prog. Biophys. molec. Biol., vol. 33, pp. 1-31, 1989.

Rosemberg J. R., Halliday D.M., Breeze P., Conway B.A., *Identification of patterns of neuronal connectivity- partial spectra, partial coherence, and neuronal interactions*. Journal of Neuroscience methods, vol. 83, pp. 57-72, 1998.

Rubinov M., Sporns O., *Complex network measures of brain connectivity: Uses and interpretations*, NeuroImage, vol. 52, no. 3, pp. 10591069, 2010.

Salvador R., Suckling J., Schwarzbauer C., Bullmore E., *Undirected graphs of frequency-dependent functional connectivity in whole brain networks*, Phil. Trans. R. Soc. B, vol. 360, pp. 937-946, 2005.

Sarvas J., *Basic mathematical and electromagnetic concepts of the biomagnetic inverse problem*, Phys. Med. Biol., no. 32, vol 1, pp. 11-12, 1987.

Schelter B., Winterhalder M., Eichler M., Peifer M., Hellwig B., Guschlbauer B., Lcking C.H., Dahlhaus R., Timmer J., *Testing for directed influences among neural signals using partial directed coherence*, Journal of Neuroscience Methods., vol. 152, pp. 210-219, 2005.

Sekihara K., Nagarajan S. S., Poeppel D., Marantz A., Miyashita Y., *Reconstructing spatio-temporal activities of neural sources using an MEG vector beamformer technique*, IEEE Transactions on Biomedical Engineering, vol. 48, no. 7, Jul. 2001.

Sekihara K., Nagarajan S. S., Poeppep D., Marantz A., Miyashita Y., *Application of an MEG eigenspace beamformer to reconstruction spatio-temporal activities of neural sources*, Human Brain Mapping, vol. 15, pp. 199-215, 2002.

Schlogl A., *A comparison of multivariate autoregressive estimators*. Signal Processing, vol. 86, pp. 2426-2429, 2006.

Schoffelen J., Gross J., *Source connectivity analysis with MEG and EEG*, Human Brain Mapping, vol. 30, pp. 1857-1865, 2009.

Shachter R.D., Kenley C.R., "Gaussian influence diagrams," Manage. Sci., vol. 35, no. 5, pp. 527-549, May 1989.

Smith V.A., Jarvis E.D., Hartemink A.J., *Evaluating functional network inference using simulations of complex biological systems*, Bioinformatics, vol 18, no. 1, 2002.

Smith V.A., Yu J., Smulders T.V., Hartemink A.J., Jarvis E.D., *Computational inference of neural information flow networks*, PLoS Comp. Biol., vol. 2, no. 11, 2006.

Sporns O., Chialvo D. R., Kaiser M., Hilgetag C. C., *Organization, development and function of complex brain networks*, Trends in Cognitive Sciences, vol. 8, no. 9, Sept. 2004.

Sporns O., Honey C. J., *Small world inside big brains*, Proc. Natl. Acad. Sci. U.S.A, vol. 103, pp. 19219-19220, 2006.

Sporns O., Honey C. J., Kötter R., *Identification and classification of hubs in brain networks*, PLoS ONE, no. 2, vol. 10, pp. e1049, 2007.

Stam C. J., *Functional connectivity patterns of human magnetoencephalographic recordings: a "small-world" network?* Neuroscience Letters, vol. 355, pp. 25-28, 2004.

Stam C. J., *Use of magnetoencephalography (MEG) to study functional brain networks in neurodegenerative disorders*, Journal of the Neurological Sciences, vol. 289, pp. 128134, 2010.

Stam C. J., de Haan W., Daffertshofer A., Jones B. F., Manshanden I., van Cappellen van Walsum A. M., Montez T., Verbunt J. P. A., de Munck J. C., van Dijk B. W., Berendse H. W., Scheltens P., *Graph theoretical analysis of magnetoencephalographic functional connectivity in Alzheimer's disease*, Brain, vol. 132, pp. 213-224, 2009.

Stam C.J., Nolte G., Daffertshofer A., *Phase lag index: assessment of functional connectivity from multi channel EEG and MEG with diminished bias from common sources.*, Hum. Brain Mapp., vol 28, no. 11, pp. 1178-93, 2007.

Stuart A., Ord K., *Kendall's advanced theory of statistics*, Wiley-Blackwell, 6th Edition, 1994.

Tass P., Rosenblum M.G., Weule J., Kurths J., Pikovsky A., Volkmann J., Schnitzler A., Freund H.J., *Detection of $n : m$ Phase Locking from Noisy Data: Application to Magnetoencephalography*, Phys. Rev. Lett., vol. 81: pp. 3291-94, 1998.

Timmermann L., Gross J., Dirks M., Volkmann J., Freund H., Schnitzler A., *The cerebral oscillatory network of Parkinsonian resting tremor*, Brain, vol. 126, pp. 199-212, 2003.

Uhlhaas P., Pipa G. ,Lima B., Melloni L., Neuenschwander S., Nikolic D., Singer W., *Neural synchrony in cortical networks: history, concept and current status*, Frontiers in Integrative Neuroscience, 2009.

Van Veen B. D., Buckley K. M., *Beamforming: A versatile approach to spatial filtering*, IEEE ASSP magazine, Apr. 1988.

Van Veen B. D., van Drongelen W., Yuchtman M., Suzuki A., *Localization of brain electrical activity via linearly constrained minimum variance spatial filtering*, IEEE Transactions on Biomedical Engineering, vol. 44, no. 9, Sept. 1997.

Varshavsky R., Horn D., Linial M., *Clustering algorithms optimizer: A framework for large datasets*, Springer-Verlag Berlin Heidelberg, 2007.

Vrba J., *Magnetoencephalography: the art of finding a needle in a haystack*, Physica C, no. 368, pp. 1-9, 2002.

Vrba J. I., Robinson E. S., *Signal processing in magnetoencephalography*, Methods, vol. 25, pp. 249-271, 2001.

Walsh B., *Markov chain Monte Carlo and Gibbs sampling*, Lecture notes for EEB 596z, 2002.

Wang J., *Minimum-norm least-squares estimation: Magnetic source images for a spherical model head*, IEEE Transactions on Biomedical Engineering, vol. 40, no. 4, Apr. 1993.

Wang J., Williamson S. J., Kaufman L., *Magnetic source images determined by a lead-field analysis: The unique minimum-norm least-squares estimation*, IEEE Transactions on Biomedical Engineering, vol. 39, no. 7, Jul. 1992.

Wang J., Williamson S. J., Kaufman L., *Magnetic source imaging based on the minimum-norm least-squares inverse*, Brain Topography, vol 5, no. 4, 1993.

Watts D. J., Strogatz S. H., *Collective dynamics of small-world networks*, Nature, no. 393, pp. 440-442, Jun. 1998.

Wheaton LA, Nolte G, Bohlhalter S, Fridman E, Hallet M. Synchronization of parietal and premotor areas during preparation and execution of praxis hand movements. Clin. Neurophysiol., 2005; 116: 1382-90.

Whittaker J., *Graphical models in applied multivariate statistics*, Wiley-Blackwell, 1990.

Zou C. and Feng J., *Granger causality vs. dynamic Bayesian networks inference: a comparative study*. BMC Bioinformatics, 2009.