University Of Surrey
Faculty of Engineering and Physical Sciences
Department Of Computing

PhD Thesis

Combined Brain Connectivity and Cooperative Sensor Networks for Modelling Movement Related Cortical Activities

Konstantinos Eftaxias

Guildford, Surrey
July 2015
University Of Surrey  
Faculty of Engineering and Physical Sciences  
Department Of Computing  

PhD Thesis  

Combined Brain Connectivity and Cooperative Sensor Networks for Modelling Movement Related Cortical Activities  

Konstantinos Eftaxias  
Supervisor: Saeid Sanei  

Guildford, Surrey  
July 2015
“The best for the group comes when everyone in the group does what’s best for himself and the group”

John Nash
Abstract

The elucidation of the brain’s anatomical and functional organisation during specific tasks is a challenging field in modern brain research. There is also a growing interest in the field of brain connectivity and its relation to specific motor and mental tasks, as well as neurodegenerative diseases like Parkinson’s and Alzheimer’s.

In this thesis, a novel approach for modelling motor tasks is proposed. This approach combines diffusion adaptation and brain connectivity measures in order to build models which describe complex tasks through time and space. In particular, an S-transform based measure is introduced to estimate the connectivity on single-trial basis. The connectivity values, corresponding to different frequency bands across time, are effectively coupled with diffusion adaptation. The diffusion strategy exploits the time-space characteristics in a distributed and collaborated manner, and leads to an enhanced model for motor or mental tasks.

Specifically, the imaginary part of S-transform coherency is introduced as an EEG connectivity measure. The performance improvement over the existing connectivity measures on a single-trial basis is demonstrated. Moreover, diffusion Kalman filtering is used as it performs well for nonstationary problems like this.

This novel method is tested on various scenarios. Initially, its performance is demonstrated for simulated datasets which are based on realistic scenarios. Then, the method is applied to two datasets of real data. The first set of experiments includes a complex motor task of clockwise and anticlockwise hand movement and the second set includes a multi-modal dataset acquired from Parkinson’s patients. The results show that the connectivity enhanced modelling outperforms the simple case where connectivity information is ignored, and can build a robust task-related model.
I would like to express my sincere gratitude to my supervisor Dr. Saeid Sanei for all of these years of continuing support. His guidance helped me to overcome the difficulties and face the challenges of the project. Also, I would like to thank my second supervisor Prof. Yaochu Jin and the members of staff in the Department of Computing for their support.

I am also grateful to Prof. Ali Sayed who contributed, especially in the beginning of the project, with his valuable input. My sincere thanks to Dr. Oana Geman for providing me the opportunity to acquire very valuable data.

Last but not least, I would like to thank my friends and family for their love and support. Specifically, I would like to thank my friend Spyros for the long scientific discussions and my partner Evi for her patience and endurance over the last three years.
Contents

List of Figures ................................................................. 5

Abbreviations ........................................................................ 7

1 Introduction ....................................................................... 9
  1.1 Brain Organisation and EEG ........................................... 10
  1.2 Brain Computer Interfaces ............................................. 13
  1.3 Brain Connectivity ....................................................... 15
  1.4 Motivation .................................................................... 16

2 Brain Connectivity ............................................................ 21
  2.1 Model-based Measures of EEG Connectivity ..................... 22
     2.1.1 Granger Causality - Parametric approach ................. 22
     2.1.2 Directed Transfer Function .................................... 25
     2.1.3 Partial Directed Coherence .................................... 26
     2.1.4 Dynamic Connectivity ........................................... 27
  2.2 Phase-based Measures of EEG Connectivity ..................... 29
     2.2.1 Phase Locking Value ............................................. 31
     2.2.2 Phase Lag Index .................................................. 31
  2.3 Coherency-based Measures of EEG Connectivity ............... 32
     2.3.1 Imaginary part of coherency .................................... 33
     2.3.2 Time-frequency representations ............................... 34
        2.3.2.1 Short time Fourier Transform ......................... 34
        2.3.2.2 Wavelet Transform ....................................... 35
        2.3.2.3 S-Transform ................................................ 35
     2.3.3 S-transform coherency and its imaginary part .............. 37
  2.4 Simulations .................................................................. 38
3 Modelling using Diffusion Adaptation ........................................ 47
  3.1 Motivation ................................................................. 48
  3.2 Diffusion Least Mean Squares ....................................... 49
    3.2.1 Simulations ....................................................... 56
  3.3 Diffusion Kalman Filtering ........................................... 58
    3.3.1 Simulations ....................................................... 60
  3.4 Diffusion Adaptation on Multitask Problems ....................... 63
    3.4.1 Simulations ....................................................... 66
  3.5 Modelling using Diffusion Adaptation and Brain Connectivity .. 69
    3.5.1 First Scenario (Diffusion LMS-DTF) .......................... 70
    3.5.2 Second Scenario (Diffusion Kalman-ImSCoh) .................. 73

4 Real Data Experiments ...................................................... 79
  4.1 Experiments on circular hand movement ............................ 80
    4.1.1 Trajectory Voting ............................................... 80
    4.1.2 Experimental setup ............................................. 84
    4.1.3 Description of the modelling and classification method ..... 85
  4.2 Experiments using multi-modal datasets of Parkinson’s patient recordings ............................................. 94
    4.2.1 Experimental setup ............................................. 95
    4.2.2 Description of the modelling method ......................... 97
    4.2.3 Results ........................................................... 100

5 Summary, Conclusions and Future Research ..............................107

Bibliography .................................................................111
List of Figures

1.1 Cerebral lobes. .................................................. 10
1.2 Electrode setup for a 10-20 EEG system of 28 electrodes. ....... 12
1.3 Modelling of fish school movement. .............................. 17

2.1 Simulation scheme for generating signals from a template with adjustable lags. .............................................. 38
2.2 Connectivity map of (a) SDTF and (b) ADTF averaged over 10 trials for the simulation scheme of Fig. 2.1. ......................... 40
2.3 Connectivity map of ImSCoh averaged over 10 trials for the simulation scheme of Fig. 2.1. ........................................ 40
2.4 Simulation scheme for generating signals from two templates with customisable lags and within a specific frequency band. ....... 41
2.5 Propagation scheme of the simulated signals. ....................... 42
2.6 Connectivity map of SDTF averaged over 10 trials for the simulation scheme of Fig. 2.4. ........................................ 42
2.7 Connectivity map of ADTF averaged over 10 trials for the simulation scheme of Fig. 2.4. ........................................ 43
2.8 Connectivity map of ImSCoh averaged over 10 trials for the simulation scheme of Fig. 2.4. ........................................ 44
2.9 Connectivity map of ImSCoh for a single trial (simulation scheme of Fig. 2.4). ........................................ 44
2.10 Connectivity map of ImSCoh averaged over 10 trials for the simulation scheme of Fig. 2.4 for (a) SNR=-5dB and (b) SNR=-10dB. 46

3.1 A network of 5 nodes. .............................................. 48
3.2 Performance of Diffusion LMS compared to the non-cooperative case. .................................................. 57
3.3 Performance of diffusion Kalman compared to the non-cooperative case and the LMS equivalents. .............................. 61
3.4 Performance of diffusion Kalman compared to the non-cooperative case and the LMS equivalents when the target dynamically changes. 63
3.5 Performance of diffusion LMS with clustering compared to diffusion LMS and the non-cooperative case when the relative distances between the optima are small. 66
3.6 Performance of diffusion LMS with clustering compared to diffusion LMS and the non-cooperative case in the general case of three optima with relative large distances between them. 67
3.7 Values of the weighted adjacency matrix for a time instant $i$ for the clustering diffusion LMS case. 68
3.8 Performance of diffusion LMS with clustering compared to diffusion LMS and the non-cooperative case on a large network of 60 nodes. 69
3.9 An illustration of a brain connectivity pattern. 71
3.10 Dataset generation of two classes for the first scenario. 71
3.11 Outline of the method for the first scenario (Diffusion LMS-DTF). 72
3.12 Percentage of classification error versus signal-to-noise ratio for both diffusion and non-cooperation methods. 73
3.13 Outline of the method for the second scenario (Diffusion Kalman-ImScoh). 75
3.14 Connectivity patterns for the signal generation of the two classes. 76
3.15 The result of LDA for the diffusion and the non-cooperation algorithms. 77
4.1 Schematic description of the trajectory method representing the directions of the filter coefficients. 82
4.2 Experimental setup for capturing the hand movement. 85
4.3 Outline of the modelling method. 86
4.4 Results of trajectory voting. 88
4.5 All the possible combinations of the desired and input signals for the estimation of model coefficients. 89
4.6 Clockwise and anti-clockwise hand circular movement classification success rates. 92
4.7 Mean differences between the classification success rates of diffusion and non-diffusion modelling. 93
4.8 Schematic of the experiment setup with the two cylinder’s positions. 96
4.9 Outline of the modelling method. ................. 97
4.10 Change point detection method for separation and alignment of trials. ........................................ 98
4.11 Results for subject 1 using the raw motion sensor signal as desired. 101
4.12 Results for subject 1 using the filtered motion sensor signal as desired. ........................................ 104
4.13 Results for subject 2 using the filtered motion sensor signal as desired. ........................................ 105
4.14 Results for subject 3 using the filtered motion sensor signal as desired. ........................................ 106
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADTF</td>
<td>Adaptive Directed Transfer Function</td>
</tr>
<tr>
<td>ALS</td>
<td>Amyotrophic Lateral Sclerosis</td>
</tr>
<tr>
<td>AR</td>
<td>Autoregressive</td>
</tr>
<tr>
<td>ATC</td>
<td>Adapt-Then-Combine</td>
</tr>
<tr>
<td>BCI</td>
<td>Brain Computer Interface</td>
</tr>
<tr>
<td>BSS</td>
<td>Blind Source Separation</td>
</tr>
<tr>
<td>CNS</td>
<td>Central Nervous System</td>
</tr>
<tr>
<td>CSP</td>
<td>Common Spatial Patterns</td>
</tr>
<tr>
<td>CTA</td>
<td>Combine-Then-Adapt</td>
</tr>
<tr>
<td>CWT</td>
<td>Continuous Wavelet Transform</td>
</tr>
<tr>
<td>DTF</td>
<td>Directed Transfer Function</td>
</tr>
<tr>
<td>DTFT</td>
<td>Discrete Time Fourier Transform</td>
</tr>
<tr>
<td>EEG</td>
<td>Electroencephalography</td>
</tr>
<tr>
<td>EMG</td>
<td>Electromyography</td>
</tr>
<tr>
<td>ERD</td>
<td>Event-Related Desynchronisation</td>
</tr>
<tr>
<td>ERP</td>
<td>Event-Related Potential</td>
</tr>
<tr>
<td>ERS</td>
<td>Event-Related Synchronisation</td>
</tr>
<tr>
<td>fMRI</td>
<td>Functional Magnetic Resonance Imaging</td>
</tr>
<tr>
<td>FPE</td>
<td>Final Prediction Error</td>
</tr>
<tr>
<td>GPDC</td>
<td>Generalised Partial Directed Coherence</td>
</tr>
<tr>
<td>ICA</td>
<td>Independent Component Analysis</td>
</tr>
<tr>
<td>ImSCoh</td>
<td>Imaginary part of S-Transform Coherency</td>
</tr>
<tr>
<td>Infomax</td>
<td>Information Maximisation</td>
</tr>
<tr>
<td>JADE</td>
<td>Joint Approximate Diagonalisation of Eigen-matrices</td>
</tr>
<tr>
<td>LDA</td>
<td>Linear Discriminant Analysis</td>
</tr>
<tr>
<td>LMS</td>
<td>Least Mean Squares</td>
</tr>
<tr>
<td>MVAAR</td>
<td>Multivariate Adaptive Autoregressive</td>
</tr>
<tr>
<td>MVAR</td>
<td>Multivariate Autoregressive</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Full Form</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>MSD</td>
<td>Mean Square Difference</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Square Error</td>
</tr>
<tr>
<td>PDC</td>
<td>Partial Directed Coherence</td>
</tr>
<tr>
<td>PET</td>
<td>Positron Emission Tomography</td>
</tr>
<tr>
<td>PLI</td>
<td>Phase Lag Index</td>
</tr>
<tr>
<td>PLV</td>
<td>Phase Locking Value</td>
</tr>
<tr>
<td>SBC</td>
<td>Schwartz’s Bayesian Criterion</td>
</tr>
<tr>
<td>SCoh</td>
<td>S-Transform Coherency</td>
</tr>
<tr>
<td>SDTF</td>
<td>Short Time Directed Transfer Function</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to Noise Ration</td>
</tr>
<tr>
<td>SOBI</td>
<td>Second Order Blind Identification</td>
</tr>
<tr>
<td>STFT</td>
<td>Short Time Fourier Transform</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machines</td>
</tr>
<tr>
<td>WT</td>
<td>Wavelet Transform</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

The human brain is definitely the most complex evolutionary biological system. It weighs about 1.5 kg and consists of a vast network of interconnected neurons. The complexity of this network makes the functional analysis of the human brain one of the most challenging tasks in modern research. During the last century, there have been a number of important breakthroughs in brain research. Brain imaging techniques such as functional magnetic resonance imaging (fMRI) and positron emission tomography (PET) are proven essential for the elucidation of the brain functionality. Among these technological advances is the electroencephalography (EEG), invented by Hans Berger in 1924. EEG is a simple, non-invasive yet powerful tool to analyse the brain activity.

In recent years, there has been a growing interest in brain connectivity [69]. From the scale of neural cells (neurons) to the scale of anatomically segregated brain regions, connectivity plays an important role in the study of brain functionality and disorders. At the same time, the study of complex networks has been exploited in order to be able to expand our knowledge of this system and the way its components interact. Additionally, great research interest is expressed in the applications of network theories to analyse such cooperative networks of agents in order to simulate behaviours of biological systems [19][22][86].

In this thesis, for the first time, we develop a movement related brain model which fuses the inherent brain connectivity and the notion of cooperative communication network to best describe the brain behaviour. In this introduction, a brief review of the related concepts is presented.
1.1 Brain Organisation and EEG

The central nervous system (CNS) is the main part of the nervous system that is responsible for the integration and processing of the information that receives and the coordination of the actions that follow these stimuli. The CNS consists of two main parts: the brain and the spinal cord.

The primary role of the spinal cord is the transmission of neural signals between the brain and the rest of the body. It also contains neural circuits that can control numerous reflexes.

Although the role of the spinal cord is very important, the organ that can be defined as the center of the nervous system is the brain. Two types of cells can be found in the brain: the neuron and the glial cells. Neurons consist of four main parts: the cell body (soma) which is the central part of the cell and contains the nucleus (where all the genetic information of the cell is stored), the axon which transfers signals to other neurons and the dendrites which receive signals from other neurons. The number of neurons in the brain of a newborn human is approximately $10^{11}$. These neurons are organised in a vast network of high complexity that responds to various stimuli and transfer information over long distances. The information is transmitted between the neurons via their synaptic electrochemical activity.

On a larger scale we can segregate the brain in three parts: the cerebrum, the brainstem and the cerebellum. Each structure does not function in isolation. The cerebral hemispheres which are the largest part of the above parts can be
anatomically divided into four lobes with names based on the names of the overlying bones in the skull. The lobes of the human cerebral cortex are: frontal, temporal, parietal and occipital lobes (see Fig. 1.1). Furthermore, we can divide the human cortex into functional areas that correspond to specific sensations. Sensory tasks like hearing, vision, olfaction and touching are dealt in specific areas in the cortex. Some significant cortical areas that are related with motor tasks are:

- Primary Motor Cortex that generates neural impulses controlling the execution of movement. It is divided by regions responsible for each part of the body.
- Posterior Parietal Cortex that transforms visual information into motor commands.
- Premotor Cortex which is responsible for the sensory guidance of movement.
- Supplementary Motor Area that plans for complex movements and coordinates two-handed movements.

One other cortex that is important for our subject is the primary visual cortex which is located in the occipital lobe and is responsible for the processing of visual stimuli.

The ability of the cerebral cortex is not limited to the processing of sensory information. The structures that process memories and feelings are located deep within the cerebrum. These structures are the basal ganglia, the hippocampus and the amygdala. The basal ganglia are involved in movement and especially in planning and fine-tuning motor actions. A dysfunction can cause Parkinson’s disease [60]. The hippocampus is involved in the formation of long-term memories and the amygdala is responsible for the relations between emotions and sensory information.

Even from this short anatomical description we can easily indicate the significance of cortical connectivity on most of the motor and mental tasks. The elucidation of connectivity patterns and their exploitation for modelling purposes is the key for a deeper understanding of several aspects of brain functionality.

Among the many methods that are used for brain imaging in research and medical applications, the most important ones are EEG and fMRI. There are some advantages in the selection of EEG signals as the main measurements in order
to explore brain connectivity. First of all, fMRI suffers from low time resolution which is a drawback when we want to examine the rapidly changing patterns of brain connectivity. On the other hand, EEG offers us detailed information through the time domain. Some limitations exist on the spatial domain as the number of electrodes is often limited. Another advantage of EEG over fMRI is its low cost and better accessibility for the patients.

The structure of an EEG system is simple. A number of electrodes (high quality electrodes offer lower-noise signals) acquire the electric signals from the scalp. The most common setting for electrode positioning is the 10-20 system. As we can see in Fig. 1.2 the 28 electrodes are named by the position they have on the scalp. For example, F stands for frontal, T for temporal, C for central, P for parietal and O for occipital lobes. In addition there is one reference electrode usually latched on the ear lobe and one ground electrode placed usually on the mastoid.

The next stage on an EEG system consists of a set of amplifiers (one for each channel) and filters that help to reject undesired signals, noise and artefacts. For digital systems an extra stage of analogue to digital converters, samples the ana-
logue signal with a sampling frequency usually close to 256 Hz. According to the Nyquist’s theorem this sampling frequency is enough, as the valuable information of the brain signals lies on frequencies below 128 Hz.

An important aspect of the brain activity that is reflected in the EEG is the brain rhythms. The basic feature of the brain rhythms is their spectral properties. Although the exact frequency band of each rhythm is subjective, the changes in brain rhythms have shown to be useful in study of the brain and the diagnosis of some brain dysfunctions and abnormalities. The main brain rhythms are alpha (8-13 Hz), beta (14-26 Hz), delta (0.5-4 Hz) and theta (4-7.5 Hz). On the same frequency band as alpha rhythms there is another family of rhythms that best represent the motor activity. The mu rhythms are generated by the motor cortex in the event of performing a voluntary motor action or a motor imagery. The study and detection of this rhythm is often used in brain computer interfaces which will be described in the next section.

1.2 Brain Computer Interfaces

EEG machines are small, relatively inexpensive and safe systems that can be accessible for consumer applications. These properties can be considered as reasons for the rapid development of brain computer interfaces (BCIs). BCIs are systems that provide a communication pathway between the brain and a device, bypassing any physical interaction. The term was introduced during the 1970s by Vidal [90] and the first steps in brain research were established. However during the last decades the research area has been constantly growing with applications ranging from medical to entertainment. BCIs became consumer products with a very active community of developers working on the developing both hardware and algorithms.

However, the main motivation for BCI research is to improve the quality of patients’ lives with neurological problems. The most interesting case is the Amyotrophic Lateral Sclerosis (ALS) where the patient, at the late stage of the condition, loses the ability to control any muscle. At this stage, a BCI system might be the only way the patient can communicate with the external world. BCI systems for these cases can be based on invasive EEG techniques by implanting epicortical and microelectrode arrays to eliminate the noise from the scalp and other exogenous sources. In 2002 an experiment by Nicolelis and Chapin [56] was
performed on monkeys in an invasive manner. The electrodes were capturing the electrical activity of a group of motor neurons while the monkeys were trying to control a robotic arm. Modelling the neuronal activity made the monkeys able to control the arm and produce complex hand movements. Subdural electrodes like these offer a high signal to noise ratio and better spatial resolution as they are implanted near the area of interest capturing the activity of individual neurons. However, the invasive methods have some disadvantages. The quality of the signals fade over time and the initial setup as well as the maintenance of the system often require risky operations. For these reasons, experiments on humans are limited and they are conducted only in particular cases of locked-in patients. In this thesis, the focus will be on the non-invasive cases as it is a more accessible technique.

BCI research is focused on the improvement of the methods that translate the brain activity into a control signal. The early BCI systems were based on signal processing methods such as bandpass filters to measure the signal power around a specific frequency range and the relevant region \[97\]. These features are enough for a simple binary decision although it demands thoroughly trained subjects. During the last two decades, many methods from the field of machine learning and adaptive learning have been used to solve this problem. Blind source separation techniques like independent component analysis (ICA) \[66\] as well as classification methods like linear classifiers \[6][94\], support vector machines (SVM) \[75][88\], neural networks \[25][68\] and hidden Markov models \[59\] were used in a variety of cases.

One of the most commonly used and popular methods for BCI is common spatial patterns (CSP) \[67\]. The main idea behind CSP is to estimate a component that can maximise the ratio of the variance between two parts of the EEG signal that correspond to different motor actions. Another technique that is commonly used for motor and visual tasks is by based on event-related synchronisation (ERS) and event-related desynchronisation (ERD) \[65][62\]. ERD/ERS is quantified based on the power of the contra-lateral signal around mu band during a time locked task.

Different research groups around the world have proposed working BCI systems. One of the first was the BCI system proposed by Farwell and Donchin in 1988 \[30\]. They developed a speller system that the user could select letters and numbers using only the EEG information. The letters were formed in a matrix
and they were flashing one by one in a random way. Whenever the desired letter flashed, the user would generate a P300 which is an event related potential (ERP) that is generated 300 ms after a stimulus. Therefore the user could select letters one by one and communicate using only the brain signals from specific electrodes.

Another similar BCI system was introduced by Sutter in 1992 [84]. The system was also designed for locked-in patients as the latter one with the main difference to be the exploitation of steady state visual evoked potentials. These potentials are the natural responses to the visual stimuli at specific frequencies. These frequencies can be picked up as in the electrical brain activity and therefore can be easily associated with the corresponding stimuli. The selection of the particular letter or command is based on the matching of the flashing symbol’s frequency on the screen and the underlying brain rhythm.

One of the pioneers in BCI research was Wolpaw and his group based on Wadsworth Center [97]. In 1991, they developed a BCI system which allowed the control of a prosthetic limb [96]. The control was based on the power of the mu rhythm around C3 electrode (see Fig.1.2). This feature was used with linear discriminant analysis (LDA) in order to control a cursor on a screen which was controlling the prosthetic limb.

Another very active group is the Graz-BCI that is led by Gert Pfurtscheller [73]. The most important contribution of the group was ERD and ERS [65][62] as it was described before. Also the group introduced the concept of using EEG signals during imaginary movement [63].

In this thesis, a general and flexible framework will be analysed that allows the modelling of the task-related brain activity by fusing brain connectivity and diffusion adaptation. In the next paragraphs, a short introduction to both will be given.

1.3 Brain Connectivity

In order for someone to understand how brain works, it is not enough to study the functionality of each part; the connections between these elements are usually more important for the study of a specific scenario. Although the problem was set back in the early days of human anatomy, it was clearly pointed out in a commentary of Francis Crick and Ted Jones in 1993 [77]. In this commentary, they mentioned the need for further information in order to understand how the
human brain works.

In 2005, Sporns et al. [78] introduced the term “connectome” in analogy with the term genome from molecular biology. The main statement was that “the pattern of elements and connections as captured in the connectome places specific constraints on brain dynamics and thus shapes the operations and processes of human cognition” [78]. The goal of this effort is to discover and explore brain networks and link them with specific brain functions and not simply come with a “wiring diagram” of the brain. This problem can be solved in three scales. The microscale in which we examine the connections between neurons, the mesoscale in which we study connections between group of neurons and the macroscale in which the brain regions are considered as nodes of the brain network. In this study we will focus on the macroscale case as it is more compliant with EEG signals.

Brain connectivity can be defined as structural, functional and effective [69]. Structural connectivity has a physical meaning. It can be described by the links between neural circuits of any scale. On the other hand, functional connectivity examines any statistical dependence between any neural regions not essentially anatomically close to each other. Statistical dependence may be estimated by measuring correlation or spectral coherence. Finally, effective connectivity refers to the causal relations between the neural elements and can be described as the combination of functional and structural connectivity.

The importance of connectivity for the brain functionality and the need of a modelling technique that incorporates all its benefits is the main motivation for this work. In the next section, this idea will be given from a network perspective and the motivation of this work will be fully described.

1.4 Motivation

In nature we can find many examples of biological systems that organise themselves in order to achieve a desirable outcome. For example, the movement of schools of fish avoiding a predator [86], the flight and formation of a flock of birds [19] and even in the microscopic scale, the organised movement of bacteria towards nutrients [22]. There is a great interest in modelling these systems by using a network of interacting agents. These connected networks of agents employ distributed and decentralised processing in order to converge to a global
Figure 1.3: Modelling of fish school movement. In these frames the fish (blue dots) are moving towards the food (red dot) while they are trying to avoid the attacks of the predator (green dot). This is modelled using diffusion adaptation where each node moves based on the information that is receiving from its neighbours. Moreover, the coordination of the fish group movement comes from constraints applied to the adaptive process.

For example, a possible application is the simulation of building evacuation on the emergency of a fire. To solve this problem we can use a network of cooperative agents. The connections between them (the amount of information they share through time) are defined by a function of the spatial distance between them. The goal for each agent is to reach the fire exits within a specific period of time. This can be set as an optimum goal for the agents’ network (minimising the distance between the agents and the fire exits). In addition, the hazards and difficulties on the way to reach this goal can be formulated using a series of constraints within
the optimisation process. With this scheme we can simulate, effectively and in a flexible way, the ideal evacuation plan.

Another example is the modelling of movement of a group of fish, towards a goal and avoiding any predators. This can be modelled by using diffusion adaptation [86]. Diffusion adaptation is used to estimate an optimum based on the data that are available over a network of nodes. Each node acts independently and collaborates with its neighbours to reach the target faster in a more realistic scenario. For this application, each fish is represented as a node connected with its neighbours and the desired output is the location of the goal (say nutrition in this case).

To improve the movement and make this model more realistic we need to introduce some constraints in the fish movement and grouping. For example, the fish cannot swim far apart and they need to minimise the distance between them. On the other hand, the fish cannot be too close and they have to keep a minimum distance between each other. One other constraint is coherent motion of a group of fish. For this reason, a group velocity is computed adaptively to ensure that a group maintains a common direction towards the goal or away from the predator. For estimating the group velocity we use a diffusion adaptation scheme where the cost function is sum of the differences between the group velocity and each node’s velocity. This way, the group velocity of the fish follows any changes in the direction of movement of any fish subgroup.

The model was implemented and the results are shown in Fig. 1.3. With blue colour we represent the fish, with red colour we represent the food (goal) and with green colour the predator. As we can see, especially from the fourth, fifth and sixth frame, after getting separated from the attack of the predator the fish regroup and swim back to their goal.

The above scenario is a good example to justify the main motivation of this work. The main idea is that there is a number of goals that the agents, based on their data input, want to reach. Therefore, each node tries to reach the goal surrounded by other nodes having the same goal. This way, in the EEG example, each node (equivalent to each electrode) will be tracking a specific optimum based on the data it has access to. The collaboration between the nodes will be established by using the connectivity values between the electrodes. Consequently the nodes that share some relation based on the connectivity measure will combine their information in order to reach their goal in a coherent way. However, the
goals are dynamic and they are constantly changing. Therefore, there will be no decision based on the convergence to a specific solution but from the time evolvement of the tracking process. In other words, the model is based on the agents’ behaviour though time and space.

This novel approach gives numerous possibilities compared to the existing methods. Firstly, it provides the flexibility to be used in a number of applications, such as classification of motor or mental tasks or modelling of a task in order to quantify an abnormality. The model includes information from the EEG signals as well as from the connectivity measure in a combined way that provides a full physiological description of the underlying task. The model may be used for the classification of complex tasks that are hard to be classified using conventional methods.

In chapter 2, there is a thorough analysis of the existing connectivity measures that can be used in some specific application. A new measure is proposed and the advantages over the existing approaches will be demonstrated using simulated data. In chapter 3, diffusion adaptation will be analysed with simulations for proving the convergence of each algorithm. In addition, some scenarios of modelling simulated data is presented. In chapter 4, experiments on real data are presented for two cases. The first is based on the classification of complex motor tasks and the second is the modelling based on multi-modal data acquired from Parkinson’s disease patients. Lastly, the thesis is concluded with a summary of the most important contributions.
Chapter 2

Brain Connectivity

Over the past 20 years the advances in areas such as neuroimaging have raised the needs for analysing the physiological data. Especially in modalities such as fMRI and EEG the analysis must incorporate both time and space. This led to the development of various measures of connectivity that estimate relations between distinct brain areas. There is growing interest in brain connectivity as it is a crucial measure in the pursuit of uncovering and understanding the functional organisation of the brain [35][36][31][39].

In the early history of neuroscience, the main research on the brain was based only on the anatomical findings. Neuroscientists like Cajal contributed to the understanding of anatomical structure of the CNS. Cajal was also famous for his pioneering work on the representation of the neuronal circuits and the first that noted the directed propagation of the signals over 100 years ago.

The breakthrough in understanding of the neuronal function and transmission came with the development of electrophysiology. At first steps, neuroscientists could record the electric activity from single neurons and managed to identify the action potentials or spikes which are the main mean of communication within the vast neuronal network [42]. The introduction of non-invasive techniques like EEG and fMRI gave easier access to data related to brain functions. On the other hand, each method measures a different aspect of neuronal activity. For example, fMRI signals have weak relation to the spiking output of neural ensemble [47]. Moreover, the CNS is organised in multiple scales, from synaptic connections between neurons to the organisation of brain regions. Therefore, each connectivity measure should be developed based on the characteristics of a specific organisation level. For example, the spatial resolution of fMRI recordings makes the use of graph theory an important tool to explore the structure of the connectivity
patterns over local and global levels [89]. However, the use of some connectivity measures can perform equally well on both modalities [5].

During the recent years, the importance of connectivity has been demonstrated in many studies. Especially in conditions like Alzheimer’s disease the loss of connectivity is proved to be related with the stage of the condition [46] [79] [93] [37] [2]. The correlation between the symptoms and disruptions of the normal connectivity patterns has been shown in the case of Parkinson’s disease as well [98][83][9]. The clear importance of connectivity and its role on the functionality of the brain on the resting state or during specific tasks is the main motivation for developing a connectivity enhanced modelling method.

The existing measures of EEG brain connectivity can be separated into categories based on the main idea behind them and the methods that are used. In this case the techniques that will be analysed in this chapter will be categorised as model-based and coherence-based. In the model-based techniques (parametric), autoregressive (AR) and multivariate autoregressive (MVAR) models are used in order to extract the connectivity patterns. In coherence-based techniques (non-parametric) the coherence in different forms is achieved by estimating the pairwise relationship between channels.

In this chapter, some of the most commonly used connectivity measures from the above categories will be described, analysed and tested on simulated signals before being applied to real data.

2.1 Model-based Measures of EEG Connectivity

In this section, AR and MVAR methods are used in order to estimate the interactions between EEG channels. Specifically, the basic Granger causality will be analysed and extended to the case of directed transfer function (DTF) and the related measures.

2.1.1 Granger Causality - Parametric approach

The Granger causality was introduced in 1969 [34] as a measure of causal influence of one time series to another. It was first used in econometrics but was then adopted in various other fields such as biological and physical sciences. Granger causality is based on AR modelling of the time series. Specifically, we consider
two time series as:

\[ x = [x_1(1), x_1(2), x_1(3), ..., x_1(t)] \]  
(2.1)

\[ y = [x_2(1), x_2(2), x_2(3), ..., x_2(t)] \]  
(2.2)

The AR models of each of the above time series can be expressed as:

\[ x_1(t) = \sum_{i=1}^{p} a_1(i)x_1(t - i) + e_1(t) \]  
(2.3)

\[ x_2(t) = \sum_{i=1}^{p} a_2(i)x_2(t - i) + e_2(t), \]  
(2.4)

where \( p \) is the model order and \( e_1, e_2 \) denote the model prediction errors for \( x_1 \) and \( x_2 \). Following the univariate cases of Eq.(2.5) the bivariate cases can be expressed as:

\[ x_1(t) = \sum_{i=1}^{p} a_{11}(i)x_1(t - i) + \sum_{i=1}^{p} a_{12}(i)x_2(t - i) + e_{1|2}(t) \]  
(2.5)

\[ x_2(t) = \sum_{i=1}^{p} a_{21}(i)x_1(t - i) + \sum_{i=1}^{p} a_{22}(i)x_2(t - i) + e_{2|1}(t). \]  
(2.6)

In order to determine if there is any causal influence of \( x_1 \) on \( x_2 \) and vice-versa, we look at the prediction errors \( e_{1|2} \) and \( e_{2|1} \). If \( e_{1|2} \) is significantly lower than \( e_1(t) \) and given that the model order \( p \) is large enough to describe the model in full, then it can be assumed that \( x_2 \) has a causal influence on \( x_1 \). To quantify this influence Granger causality can be defined as:

\[ GC_{x_2 \rightarrow x_1}(t) = \ln \frac{\text{var}(e_1(t))}{\text{var}(e_{1|2}(t))}, \]  
(2.7)

where \( \text{var}(\cdot) \) denotes variance function.

The above equation defines the Granger causality measure in relation with time. In EEG and generally brain connectivity applications it is more useful to define the measures as a function of frequency as it is important to isolate task-related activity that can be found on some specific frequency bands. In order to achieve that, the above system of equations is transformed to the frequency
domain and by writing Eq. (2.5) in matrix form we have:

\[
\begin{pmatrix}
A_{11}(f) & A_{12}(f) \\
A_{21}(f) & A_{22}(f)
\end{pmatrix}
\begin{pmatrix}
X_1(f) \\
X_2(f)
\end{pmatrix}
=
\begin{pmatrix}
E_1(f) \\
E_2(f)
\end{pmatrix},
\tag{2.8}
\]

where \(A(f)\) is obtained by the Fourier transformation as \(A_{kl}(f) = \sum_{j=1}^{p} a_{kl, j} e^{-i2\pi f j}\).

Eq. (2.8) can be rewritten in order to form the transfer function matrix \(H(f)\) as the inverse of matrix \(A(f)\) that contains the terms \(A_{kl}(f)\):

\[
X(f) = H(f)E(f).
\tag{2.9}
\]

The spectral density matrix can now be formed as:

\[
S(f) = H(f)\Sigma H^*(f).
\tag{2.10}
\]

where \(\Sigma\) is the covariance matrix of the errors \(e_{12}\) and \(e_{21}\). In order to obtain the causality relation between the two signals we look at the autospectrum of \(x_1(t)\) which in our case is \(S_{11}(f)\). However \(S_{11}(f)\) contains cross terms of \(x_1\) and \(x_2\) so a transformation is needed to make the two terms distinguishable [32][26]. We can achieve this by multiplying both terms of Eq. (2.8) by:

\[
\begin{pmatrix}
1 & 0 \\
-\Sigma_{12} & 1
\end{pmatrix}.
\tag{2.11}
\]

The Granger causality of \(x_2\) to \(x_1\) in the frequency domain can be expressed as:

\[
GC_{x_2 \rightarrow x_1}(t) = \ln \frac{S_{11}}{S_{11} - (\Sigma_{22} - \Sigma_{12} \Sigma_{11}) |H_{12}|^2}
\tag{2.12}
\]

The above way of estimating the Granger causality needs the prior knowledge of an AR model which is trivial to estimate. Granger causality bears the disadvantages of using a parametric estimation method, like the estimation error and the model order setting which adds extra erroneousness. Moreover, it is a bivariate measure and therefore it cannot include the full domain of the EEG channels. In the next section the multivariate equivalent will be discussed and a family of brain connectivity estimators based on MVAR modelling will be established.
2.1.2 Directed Transfer Function

Granger causality offers a reliable brain connectivity measure when it is assumed that there are only two channels related and no other channels influencing. In practice this is difficult to assume and it causes a lot of false positives as two channels that are connected indirectly through a third channel cannot be distinguished from a direct relation. In order to overcome this problem our focus is on measures that are based on multivariate modelling.

Considering our application, assume that we can acquire data $x_n(t)$ from $N$ electrodes of an EEG system. $x(t)$ is formed as:

$$x(t) = [x_1(t), x_2(t), x_3(t), ..., x_N(t)]$$

(2.13)

where $t$ denotes discrete time samples. A MVAR model can be described using the following equation:

$$x(t) = \sum_{i=1}^{p} A(i)x(t - i) + e(t)$$

(2.14)

where $e(t) = [e_1(t), e_2(t), e_3(t), ..., e_N(t)]$ is a vector of uncorrelated zero mean noise processes, $A(i)$ is an $N \times N$ matrix of the MVAR model coefficients for each $i$ and $p$ is the model order.

With the MVAR model estimated, the DTF [40][41] can be achieved by transforming Eq. (2.14) to the frequency domain. In order to achieve that Eq. (2.14) is rewritten as:

$$e(t) = A(0)x(t) - \sum_{i=1}^{p} A(i)x(t - i)$$

(2.15)

where $A(0) = I$. By changing the sign of $A(i)$ Eq. (2.15) can be reformed in the following way:

$$e(t) = \sum_{i=0}^{p} \tilde{A}(i)x(t - i)$$

(2.16)

where $\tilde{A}(i) = -A(i)$ apart from $A(0)$ which remains as the identity matrix. The above equation is transformed to the frequency domain:

$$E(f) = \tilde{A}(f)X(f)$$

(2.17)

$$X(f) = \tilde{A}^{-1}(f)E(f) = H(f)E(f)$$

(2.18)
where $H(f)$ denotes the transfer matrix. The transfer matrix can be computed by applying the Z transform [14]:

$$H(f) = \left( \sum_{m=0}^{p} \tilde{A}(m)e^{-2\pi imf} \right)^{-1} \quad (2.19)$$

The transfer matrix $H(f)$ represents the causal connection between two signals of the system. In Eq. (2.18) we can see how the transfer matrix contains this information as it can be considered as a modelling process of the white noise $E(f)$ in order to form the signals $X(f)$. For example, $H_{kl}(f)$ represents the connection between the signals from the $k$th and the $\ell$th electrodes at the frequency $f$. Based on this, DTF can be defined as:

$$\theta_{kl}^2(f) = |H_{kl}(f)|^2 \quad (2.20)$$

A normalised version of DTF can be defined by the following equation:

$$\gamma_{kl}^2(f) = \frac{|H_{kl}(f)|^2}{\sum_{m=1}^{N} |H_{km}(f)|^2} \quad (2.21)$$

The values of the normalised DTF are in the interval $[0, 1]$ and the sum of $\gamma_{kl}^2(f)$ across each row is equal to 1. The normalisation is very important for our method as it’s needed for the diffusion adaptation which will be in studied later chapters of the thesis.

As described above, DTF is derived from the multivariate AR coefficients. This is one of the main advantages of DTF over other methods like Gragner causality which is based on a bivariate AR model. Another advantage of DTF is that it can determine the directionality in the coupling even if the information lies on the overlapping frequencies.

2.1.3 Partial Directed Coherence

Introduced in 2001 [8], the partial directed coherence (PDC) is a measure that follows the same idea as DTF. It is based on the MVAR model of the data as it is described in the previous section. In terms of definition, unlike DTF which uses $H(f)$, PDC employs $\tilde{A}(f)$ as:
where $\mathbf{a}_\ell(f)$ is the $\ell$-th column of $\mathbf{A}$ and the superscript $H$ denotes the complex conjugate and transpose operation. As its name implies, PDC is a directed measure of connectivity the same way that DTF is. The main difference between DTF and PDC stands on the fact that PDC reflects the outflow of information from node $\ell$ to node $k$ compared to the total outflow of information coming out of node $\ell$ as the denominator from Eq. (2.22) shows.

Another variation of PDC, called generalised PDC (GPDC), follows the DTF formula for the normalisation and therefore is focused more on the sources of information and not on the sinks. The GPDC \[15\] is defined as:

$$gP_{k\ell}^2(f) = \frac{A_{k\ell}(f)}{\sum_{m=1}^{N} |A_{m\ell}(f)|^2} \quad (2.23)$$

However, one can see some problems with this definition as well. First of all, as the measure represents the influence of channel $\ell$ to channel $k$, adding terms to the sum of the denominator will alter the measure from process to process (or even across different frequencies), making it dependent on the number of sources. Also PDC in both versions is not scale-invariant which means that care has to be taken when comparing data from different measurements or modalities.

2.1.4 Dynamic Connectivity

In previous sections three types of connectivity measures that are based on parametric models were explained. In order to apply these measures to EEG data, we need first to estimate the MVAR model using a representative model. There are several criteria to choose the model order including Akaike’s Final Prediction Error (FPE) criterion and Schwarz’s Bayesian Criterion (SBC). In \[50\] Lütkepohl compares the above criteria and concludes that SBC offers better results in predicting the model order. For the computation of the MVAR coefficients, the algorithm ARfit \[54\] is widely used. The algorithm uses a stepwise least squares algorithm to estimate the coefficients of the model with the optimal order $p_{opt}$ chosen from the above criteria.

However, the problem is that on a time series of real biological data such as the EEG data, stationarity is not guaranteed. In order to capture any dynamical changes in brain connectivity we need to be able to estimate the MVAR
coefficients across time. A first simple solution is to apply a sliding overlapping window on the data. For each time instant $t$, the ARfit algorithm is applied to the windowed data segment and the coefficients $A_t$ are estimated. Following that procedure, DTF or PDC can be estimated and have a time-frequency representation of the connectivity patterns. The above technique is usually referred to as short time DTF (SDTF) [33]. In SDTF the choice of the window length is important. Larger windows may cause the loss of rapid changes in connectivity and smaller windows may introduce larger errors to the MVAR fitting. As a rule of thumb the window must be much larger than the model order and at least twice the period of the main harmonic of the signal. Specifically if the number of channels is $n$ and the window length is $N_w$ then $nN_w$ must be much larger than $n^2p$ which represents the parameter number of the MVAR model.

However, keeping a short data window may cause some loss on the quality of the model fitting. For this reason, more than one trial is needed in order to reach more reliable results. This applies to all the parametric measures that need MVAR model fitting and have been discussed until now. In order to use multiple trials on the MVAR fitting we need to average the correlation matrix over the trials. This way the model that is resulted by solving the Yule-Walker equations using the existing algorithms, includes all the data across different trials and best represents the under study system. However, this is a limitation when a single trial needs to be estimated as it will be seen on the next chapters.

Apart from using a sliding window to capture the temporal dynamics of connectivity, another technique uses Kalman filtering to estimate a multivariate adaptive autoregressive (MVAAR) model and tackles the non-stationarity problem [95].

So, Eq. (2.14) can be rewritten in order to include the adaptive nature of the MVAR model as follows:

$$x(t) = \sum_{i=1}^{p} A(i, t)x(t - i) + e(t)$$ (2.24)

The above equation can be characterised as an observations equation, with the state equation as follows:

$$A(i, t) = A(i, t - 1) + W(i, t - 1)$$ (2.25)

where $W(i, t)$ of size $N \times N$ denotes the process noise. Solving the above equations
using Kalman filter, a set of model coefficients $A(i, t)$ that are adapted through time can be achieved. Compared to SDTF, the adaptive directed transfer function (ADTF) that is computed from the MVAAR coefficients, offers a smoother estimation of dynamic connectivity. This happens because of the adaptive computing of $A$. For the SDTF, every $A(i, t)$ is not connected with the previous values $A(i, t - k)$ as every step is independent from the previous ones. As it will be discussed and proved in the next sections, ADTF using MVAAR model fitting is one of the most reliable ways of estimating the brain connectivity among the parametric methods.

2.2 Phase-based Measures of EEG Connectivity

Another prospective for the estimation of brain connectivity is by exploitation of the phase relations between two signals. If we think connectivity as the result of a propagation scheme there should be a constant phase difference between specific spectral components between the two channels. The effect of phase-locking over a specific period of time (or phase synchrony as it is often met in the literature) leads us to a non-linear way of estimating EEG connectivity. The methods based on these principles are not designed to replace the methods based on coherence or MVAR modelling but they give useful complementary information. The motivation for studying this category of measures came from the theory of chaotic oscillations and the application of deterministic chaos to the EEG signals [69].

In order to take advantage of the phase properties to estimate the synchrony between two signals, two steps need to be followed. First, the estimation of the instantaneous phase is needed. This is quite important as we need to be able to capture any temporal changes of the phase. The second step is to set a measure in order to quantify the level of phase synchrony.

Before exploring the usual ways of estimating the instantaneous phase, let us take a look on the definition. Assume that we have two signals $x_1(t)$ and $x_2(t)$ that we want to measure the synchrony between their phases. The phases are $\phi_1(t)$ and $\phi_2(t)$. According to the definition a constant difference between the two instantaneous phases for a small period of time is needed. This can be described by the following equation:
\[ |n\phi_1(t) - m\phi_2(t)| \approx \text{const} \tag{2.26} \]

For the estimation of \( \phi_1(t) \) and \( \phi_2(t) \) several methods have been proposed, particularly the two most common ones which use the wavelet convolution and the Hilbert transform \([44]\). In the case of wavelet convolution a wavelet function is chosen in order to match the application’s frequency range and the phase difference can be determined using the angles of the wavelet coefficients of each signal. Mathematically, for signal \( x_j \) we have:

\[ W_{x_j}(\tau, f) = \int_{-\infty}^{+\infty} x_j(u)\Psi^*_{\tau,f}(u)du \tag{2.27} \]

where \( \Psi^*_{\tau,f}(u) \) is the complex conjugate of the wavelet at frequency \( f \) and time \( \tau \). Then, the instantaneous phase difference can be defined using:

\[ e^{j(\phi_1(f,\tau) - \phi_2(f,\tau))} = \frac{W_{x_1}(\tau, f)W^*_{x_2}(\tau, f)}{|W_{x_1}(\tau, f)||W_{x_2}(\tau, f)|} \tag{2.28} \]

An alternative way of estimating the instantaneous phase is the use of Hilbert transform. The complex analytic signal \( z_1(t) \) can be formed by using the original signal \( x_1(t) \) as the real part and its Hilbert transform \( \tilde{x}_1(t) \) as the imaginary part:

\[ z_1(t) = x_1(t) + i\tilde{x}_1(t) \tag{2.29} \]

with the Hilbert transform obtained from the following equation:

\[ \tilde{x}_1(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x_1(\tau)}{t - \tau} d\tau. \tag{2.30} \]

The analytic expression can be written in polar form as:

\[ z_1(t) = A_1(t)e^{i\phi_1(t)} \tag{2.31} \]

from where the amplitude \( A_1(t) \) and the instantaneous phase \( \phi_1(t) \) can easily be represented as:

\[ A_1(t) = \sqrt{x_1^2(t) + \tilde{x}_1^2(t)} \tag{2.32} \]

\[ \phi_1(t) = \arctan \frac{\tilde{x}_1(t)}{x_1(t)} \tag{2.33} \]
The Hilbert transform in its discrete form can be computed by performing a Fourier transform (discrete time Fourier transform - DTFT), shifting the phase by $\pi/2$ and return to the time domain with an inverse Fourier transform.

2.2.1 Phase Locking Value

After estimating the instantaneous phase for each signal, the pairwise phase differences from Eq. (2.26) can be estimated with $m = n = 1$ for making this case simpler (comparing the phases for the same frequencies). From this point, it can be easily seen if the instantaneous phase difference is bounded. A simple way, proposed in [43], is to project the differences of multiple samples on a circle using the following equation:

$$PLV = \left| \frac{1}{N} \sum_{k=0}^{N-1} e^{i\Delta \phi(t_k)} \right|$$

At the phase locking state the instantaneous phase difference $\Delta \phi(t_k)$ takes the values close to 0 and the $PLV$ value approaches 1. In every other case of random differences the value will be closer to 0, quantifying the phase synchronisation between two signals at a specific time segment. The above method was introduced as phase locking value or PLV [43].

2.2.2 Phase Lag Index

However, this simple form of phase synchronisation is not reliable for an EEG application. The main reason is that the mean phase coherence cannot distinguish the phase synchronisation caused by volume conduction from the phase synchronisation between two different sources.

In order to make this distinction, the focus is on discarding any phase synchronisation terms with small differences. That will ensure that the contribution of volume conductivity will be eliminated, as small phase difference between channels are expected to enable capturing common sources. So in order to solve this issue, phase differences near 0 or $\pi$ need to be discarded. As proposed in [80], a way of doing this is to look at the asymmetry of the distribution of the phase differences. If there is no connection between the two signals a flat distribution of phase differences across time is expected. If there is some common source affecting the two signals (volume conduction) then a symmetric distribution around 0
or $\pi$ is expected. On any other cases the distribution is asymmetrical and this indicates phase synchronisation.

The phase lag index (PLI) exploits this property in order to reach to a measure for EEG brain connectivity. In order to express the asymmetry of the phase difference, the signs of the phase difference over time need to be estimated:

$$PLI = \left| \frac{1}{N} \sum_{k=0}^{N-1} \text{sign}(\Delta \phi(t_k)) \right| \quad (2.35)$$

Following this approach, if the sum in Eq. (2.35) is 0, it is an indication of a flat distribution of phases or having symmetric distribution around 0. In any other significant asymmetries the PLI is greater than 0 with the maximum value approaching 1. In order to see which of the signals are leading the absolute value in Eq. (2.35) needs to be ignored.

PLI was later extended to directed PLI [81] to show clearer which of the signals are leading and to Weighted PLI [92] which is proved more robust compared to the simple version.

### 2.3 Coherency-based Measures of EEG Connectivity

One of the first connectivity measures was the correlation between two signals. Soon the need of a frequency based measure led to the replacement of correlation with coherency. Coherency is a measure of the relation between two signals (or EEG channels) at a specific frequency. Assume two signals $x_k(t)$ and $x_l(t)$ with the respective Fourier transforms of $X_k(f)$ and $X_l(f)$ the cross spectrum can be defined as:

$$S_{k\ell}(f) = \langle X_k(f)X^*_\ell(f) \rangle \quad (2.36)$$

where $\ast$ denotes the complex conjugate and $\langle \cdot \rangle$ the expectation value. Then coherency can be defined as:

$$C_{k\ell}(f) = \frac{S_{k\ell}(f)}{\sqrt{S_{kk}(f)S_{ll}(f)}} \quad (2.37)$$

Thus, coherency is a complex measure that gives the relation between channels. As noted in [57] there is a small difference between the term coherency and coherence with the latter usually referring to the absolute value of coherency. In
this section the interest is in the complex version (coherency) as all the information (especially phase) needs to be exploited.

If the cross-spectrum is rewritten in polar form one can notice that there is a close relation between this definition to the PLI from the previous section. The difference lies on the fact that PLI is independent of the amplitudes whereas coherency is not. This is not an obvious drawback for coherency as its dependency on the amplitudes can lead to elimination of some low amplitude components that their phases affected more by noise.

2.3.1 Imaginary part of coherency

Apart from the amplitude dependency, coherency is a similar measure to PLI. Therefore, it captures all the interactions between the channels. As mentioned in the previous section, in EEG applications it is preferable to eliminate the contribution of small phase differences as they often reflect a common source captured by both electrodes. In order to achieve this, in the phase-based measures PLI is introduced. For coherency we may need to develop a more suitable strategy.

In order to distinguish and isolate the common sources, one can claim that a blind source separation approach can give a solution. However, the constraints that have to be imposed and the complexity of such a solution is restrictive for this case. Assume that two signals from channels \(k, \ell\) that are an exclusive superposition of \(M\) independent sources \(S_m(f)\) as:

\[
X_k(f) = \sum_{m=1}^{M} \alpha_{km} S_m(f) \quad (2.38)
\]

\[
X_\ell(f) = \sum_{m=1}^{M} \alpha_{\ell m} S_m(f) \quad (2.39)
\]

Therefore, when the cross-spectrum is computed we have:

\[
S_{k\ell} = \langle X_k(f)X_\ell^*(f) \rangle = \sum_{m} \alpha_{km} \alpha_{\ell m} \langle S_m(f)S_m^*(f) \rangle = \sum_{m} \alpha_{km} \alpha_{\ell m} \langle S_m(f)S_m^*(f) \rangle = \sum_{m} \alpha_{km} \alpha_{\ell m} |S_m(f)|^2 \quad (2.40)
\]

\[
= \sum_{m} \alpha_{km} \alpha_{\ell m} |S_m(f)|^2 \quad (2.41)
\]

which is real-valued. This leads to a real-valued coherency, which means that the real part of coherency is highly affected by volume conduction components.
This is the main motivation for keeping the imaginary part of coherency when we want to avoid these components. As both the real and imaginary parts represent the same thing (coherence) keeping the imaginary part is like looking the same measure from another angle [57].

However, the above methodology applies to stationary signals and we need to assume quasi-stationarity for non-stationary signals. In the next section, ways of doing that by using time-frequency representations like short time Fourier transform (STFT), wavelet transform (WT) and Stockwell transform (S-transform) will be examined.

2.3.2 Time-frequency representations

One important limitation with all the above methods is the requirement of stationarity. In most of the cases, stationarity or quasi-stationarity can be assumed in a short time interval and the spectrum can be computed across time. This way, a time-frequency representation of the signal is achieved. Estimating coherence from this representation, we reach to the so called “coherogram”. Estimating the coherograms from all the combinations of channels provide us a complete view of the interactions that occur in the whole time and frequencies domains. Therefore, a task can be related with a specific connectivity pattern by focusing on the corresponding time interval and frequency band.

2.3.2.1 Short time Fourier Transform

The Fourier transform in Eq. (2.36) can be replaced with STFT to capture the signal frequency changes across time and better deal with the signal dynamics such as nonstationarities. For channel $x_k(t)$ we have:

$$X_k(\tau, f) = \int_{-\infty}^{+\infty} x_k(t)w(t - \tau)e^{-i2\pi ft}dt$$ (2.42)

where $w$ is the window function. The right choice of the length for the window length is crucial for tackling the non-stationarity problem. Too long window lengths violate the quasi-stationary property we are trying to achieve and too short lengths cause low frequency resolution. So, a trade-off between time and frequency resolutions is always desired for the right time-frequency representation.
2.3.2.2 Wavelet Transform

As an alternative to the use of STFT for calculating coherency is the use of WT. The WT needs a-priori information in order to reach to optimal parameters for the time-frequency resolution but it has an advantage over the STFT. The advantage is that the WT offers variable frequency and time resolution. For example, in high frequencies shorter windows can be used in order to achieve better time resolution. On the contrary, in low frequencies the time resolution is sacrificed by using longer windows but better frequency resolution is achieved. This is achieved by the WT. In more detail the continuous wavelet transform (CWT) can be expressed using the following equation:

$$X_k(\tau, \delta) = \int_{-\infty}^{+\infty} x_k(t) \frac{1}{\sqrt{\delta}} \psi\left(\frac{t - \tau}{\delta}\right) dt$$

where $\psi$ denotes the mother wavelet function with $\delta$ the dilation and $\tau$ the time translational value. The WT is actually a time-scale representation as it doesn’t directly represent the frequency domain information. One drawback compared to STFT is that the phase information in WT is not clear and depends on the type of wavelet used.

Our main concern is to have a time-frequency representation that has the least parameters to be defined and keep the WT’s good properties of balanced time-frequency resolution. In this direction, the S-Transform, detailed in the next section, is employed.

2.3.2.3 S-Transform

The S-transform was introduced by Stockwell in 1994 for use in geophysics data [82]. The main motivation was to establish a time-frequency transformation that could avoid the limitations of STFT (low resolution) and WT (no phase information). Therefore, the idea behind S-transform is to combine the advantages of STFT and WT by using a scaled window and a phase factor. In order to derive the S-transform we can either begin from STFT or WT.

Starting from STFT, Eq. (2.42) is the starting point and a window function that provides the necessary scaling for different frequencies is chosen. A good choice for the window function is to use a normalised Gaussian as:

$$w(t) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{t^2}{2\sigma^2}}$$

(2.44)
where $\sigma$ is the dilation. Expressing $w(t)$ as a function of dilation $\sigma$ and time translation $\tau$ we reach to the following equation.

$$X_k(\tau, f, \sigma) = \int_{-\infty}^{+\infty} x_k(t) \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(t-\tau)^2}{2\sigma^2}} e^{-i2\pi ft} dt$$  \hspace{1cm} (2.45)

However, as our goal is to form a parameter-free time-frequency transformation it is important to not leave $\sigma$ as a variable and set it in a way that will make $w(t)$ into a scaled window function that depends on the frequency. In order to achieve this the constant $\sigma$ is replaced with the following function:

$$\sigma(f) = \frac{1}{f}$$  \hspace{1cm} (2.46)

and the final S-transform equation is obtained:

$$X_k^{(ST)}(\tau, f) = \int_{-\infty}^{+\infty} x_k(t) \frac{f}{\sqrt{2\pi}} e^{-\frac{(t-\tau)^2}{2f^2}} e^{-i2\pi ft} dt$$  \hspace{1cm} (2.47)

A similar result can be achieved if we start from the WT definition. In Eq. (2.43) where the CWT is defined, a mother wavelet function (denoted as $\psi$) is used. We set $\psi$ as:

$$\psi\left(\frac{t-\tau}{\delta}\right) = e^{-\frac{(t-\tau)^2}{2\delta^2}} e^{-i2\pi ft}$$  \hspace{1cm} (2.48)

Then, by replacing $\delta$ with the inverse of frequency, as before, and multiplying by a correction factor Eq. (2.43) becomes:

$$X_k^{(ST)}(\tau, f) = \sqrt{\frac{f}{2\pi}} e^{-i2\pi f\tau} \int_{-\infty}^{+\infty} x_k(t) \sqrt{f} e^{-\frac{(t-\tau)^2}{2f^2}} e^{-i2\pi f(t-\tau)} dt$$  \hspace{1cm} (2.49)

which is the resulted S-transform as in Eq. (2.47). However, the above transform is not consider exactly as wavelet as the wavelet in Eq. (2.48) doesn’t satisfy the criterion of zero mean.

By examining the wavelet function we can come up with some interesting conclusions about how the S-transform differs from WT. First, the wavelet function can be described as the product of two parts: the phase information part resulting from the factor $e^{-i2\pi ft}$ selects the localising frequency and the Gaussian function $e^{-\frac{(t-\tau)^2}{2\sigma^2}}$ acts like the window function and localises in time. By keeping the factor $e^{-i2\pi ft}$ without any time translation we are making sure that the

Konstantinos Eftaxias

36
phase information is referenced at $t = 0$. This is a property that WT doesn’t have. Also, the normalisation factor $\sqrt{\frac{f}{2\pi}}$ normalises the Gaussian function that is responsible for the amplitude spectrum. Therefore, the amplitude response is independent from the frequency which again doesn’t happen in the WT. So, the S-transform is an elegant tool for estimating the time-frequency transformation of a non-stationary signal without having to define any extra parameters.

### 2.3.3 S-transform coherency and its imaginary part

As mentioned in the previous section the S-transform has certain advantages over the traditional methods for time-frequency representation. Especially, for the application to non-stationary signals and applications where the phase information needs to be kept intact. In order to form a connectivity measure based on S-transform, the cross spectrum terms are formed as in Eq. (2.36):

$$S^{(ST)}_{k\ell}(\tau, f) = \langle \hat{X}_k(\tau, f)\hat{X}_\ell^*(\tau, f) \rangle$$

where $\hat{X}_k$ denotes the S-transform of $x_k(t)$. In the same manner, the so-called S-transform coherency (SCoh) can be estimated as:

$$C^{(ST)}_{k\ell}(\tau, f) = \frac{S^{(ST)}_{k\ell}(\tau, f)}{\sqrt{S^{(ST)}_{kk}(\tau, f)S^{(ST)}_{\ell\ell}(\tau, f)}}$$

SCoh is a complex measure with real and imaginary parts from where the magnitude (coherence) and most importantly the phase difference $\Delta\phi$ can be estimated from SCoh argument. As mentioned in the previous section, the imaginary part of coherency is important when we want to isolate real interactions between areas and reject common source elements that result from the volume conduction effect.

To conclude, the imaginary part of SCoh (ImSCoh) is a connectivity measure that can help us uncover true connectivity patterns when stationarity is not guaranteed. In the next section a simulation scheme will be used to test and compare most of the connectivity measures mentioned and analysed in this chapter.
2.4 Simulations

In order to test the connectivity measures, a realistic scenario is needed in order to generate signals with the same properties as the natural signals. Moreover, the signals must be generated in a way that they follow a connectivity pattern that varies across time. This way, the ability to track the dynamical changes in the relations between brain regions can be tested.

For these reasons, a simulation scheme is designed to generate signals. In the first simplified case, a signal generation system with three nodes is used while one signal is propagating to these nodes. The system is shown in Fig. 2.1. The initial phase is to create a template segment out of Gaussian noise. This can be simply done by bandpass filtering the Gaussian noise over a specific frequency band. For this simulation, the band between 10 and 18 Hz is used as this band is close to the alpha/beta band we are interested in. In a specific experiment, one template signal of 200 samples is generated. In order to simulate a propagation scheme, a low level noise is added to the templates and the resulting signal sample is...
located between two segments of Gaussian noise. Thus, the signals are consisted of a segment of Gaussian noise followed by a noisy version of the template and another segment of Gaussian noise. The lengths of the starting and final segments of Gaussian noise (the relative position of the template in the signal) determine the propagation scheme. For this particular case the three signals contain the template signal at the sample numbers 200, 201, and 212 respectively. Therefore, the lags between signal 1 and 2 is one sample and between 1 and 3 8 samples.

For this simulation, three measures were tested, chosen by their suitability for our application. The method that was first tested was DTF. In the specific simulation, non-stationary signals are used and the connectivity patterns are expected to change through time. Therefore, the SDTF version of the method was tested, which uses a sliding overlapping window. For each window a MVAR model is fitted to the data and the result is used to compute the DTF coefficients for the specific time instant. For this experiment a model order of 12 and a sliding window of 40 samples with maximum overlap is chosen. To smooth the result, we take the average over 10 trials. The resulted connectivity map of the method is shown in Fig. 2.2(a).

In addition to SDTF, ADTF was also tested. The main difference between SDTF and ADTF is the fact that the MVAR computation is done by means of Kalman filtering. This helps for smoother tracing of the changes in the MVAR coefficients. On the other hand, the tracking ability needs some tweaking in order to avoid poor time resolution. In our experiments, the model order was determined with the help of the FPE criterion. The resulted connectivity map is shown in Fig. 2.2(b).

With a first look at the two connectivity maps of SDTF and ADTF, a few assumptions can be verified. First of all, the fact that SDTF is a more noisy in comparison with ADTF as on each time instant, we have a new estimation of the MVAR coefficients without taking into account the prior estimated models. On the contrary the Kalman filter tries to track any dynamic changes of the underlying model in an adaptive way. As it can be seen in Fig. 2.2, the two high peaks in both methods are between signal 1 to signal 2 and signal 1 to signal 3 at frequencies around 15 Hz as expected. The multivariate property of both these methods is causing the absence of any significant connectivity from signal 2 to signal 3 as the source is signal number 1. This property is missing in bivariate measures like ImSCoH, but in some applications this is not necessarily
Figure 2.2: Connectivity map of (a) SDTF and (b) ADTF averaged over 10 trials for the simulation scheme of Fig. 2.1. Both of the measures can detect a connection from node 1 to 2 and node 1 to 3 around 15 Hz.

Figure 2.3: Connectivity map of ImSCoh averaged over 10 trials for the simulation scheme of Fig. 2.1. A connection from node 1 to 3 and node 2 to 3 is detected. There is no detection from node 1 to 2 as connections with small lags are rejected by ImSCoh.

drawback.

The third measure that is applied to this particular dataset is the ImSCoh. Unlike previous measures the ImSCoh doesn’t need any specific parameter to set. As it can be seen in Fig. 2.3, the application of ImSCoh to this dataset offers a much clearer view of the propagation between signal 1 to signal 2 and signal 3. The main difference between the previous results of the model based methods is that there is no connection from signal 1 to signal 2. This is caused by the choice of keeping only the imaginary part of S-coherency and thus eliminating any interactions with small lags. In real data applications this will help eliminate
any connectivity caused by the volume conduction effect. However, as ImSCoh is a pairwise measure, there is an interaction between signals 2 and 3 as these signals have the same components between 200 and 400 samples with the lag of 7 samples. So, in this case causality is difficult to be determined, but we can assume the propagation scenario based on the lags between the channels that can be easily extracted from ImSCoh.

In order to test the above measures in a more realistic and complex scenario, another simulation scheme is used and presented in Fig. 2.4. There are two main differences compared with the previous scheme: two template signals are generated and also all the signals are filtered within the frequency band 1-45 Hz which is the usual band for EEG processing. Therefore, the generated signals which have similar spectral properties to those of EEG signals. There are two underlying propagations of templates 1 and 2. Particularly, signals 1 and 2 are based on the same template with frequencies that lie between 10 and 18 Hz and
Figure 2.5: Propagation scheme of the simulated signals.

Figure 2.6: Connectivity map of SDTF averaged over 10 trials for the simulation scheme of Fig. 2.4. It is difficult to identify the connectivity pattern from SDTF on this more realistic simulation.

signals 3, 4 and 5 are based on a different template within the same frequency band. Signal 6 does not contain any template of specific band and it is acquired directly by filtering the Gaussian noise. The propagation scheme that is created in this example is shown in Fig. 2.5.

In Figures 2.6, 2.7 and 2.8 the results of the three connectivity measures
Figure 2.7: Connectivity map of ADTF averaged over 10 trials for the simulation scheme of Fig. 2.4. Similarly with SDTF it is difficult to identify the connectivity pattern. Moreover, many false positive peaks are making the measure less reliable.

are shown. As it can be seen in the SDTF figure, the measure captures the interactions on the specific time and frequency ranges but it fails to describe completely the two propagations. This is mainly caused by the fact that the two template signals, although different, lie on the same frequency band. The MVAR model failed to pick up the difference between them and it can be seen as the source of the templates is signal 1 and 2. This is clearer from the ADTF figure where it seems that on the specific frequency band there is a causality from signal 1 to all the other signals. Both examples are common cases where the MVAR fit fails to represent the real interactions between the channels.

On the other hand in Fig. 2.8 the ImSCoh depicts perfectly the interactions between the channels. Moreover, the phase information is kept unaffected and can describe perfectly the propagation scheme of Fig. 2.5. In addition, ImSCoh is proved to be robust for different levels of noise. As it is shown in 2.10, ImScOh performs relatively well even for SNRs as low as -10 dB. Of course, the drawback of ImSCoh as a pairwise method can be seen in developing some peaks as in channel 3 to channel 2. This is caused by the common frequency components
2.4 ImScoh performance is proven satisfactory even on a single trial basis. Here, the connectivity pattern is clearly and accurately detected. Figure 2.8: Connectivity map of ImSCoH averaged over 10 trials for the simulation scheme of Fig. 2.4. Here, the connectivity pattern is clearly and accurately detected.

Figure 2.9: Connectivity map of ImSCoH for a single trial (simulation scheme of Fig. 2.4). ImScoh performance is proven satisfactory even on a single trial basis which is a main requirement for the proposed modelling method.
that the templates have but still these peaks are lower than the ones representing the propagation pattern. Another important feature of ImSCoh is the ability of producing reliable results even for single trials. In Fig. 2.9 the same peaks with the averaged version can be found amongst other inevitable maxima.

In conclusion, the requirements of the main method of this thesis lead to introducing ImSCoh as the most suitable connectivity measure. The reasons for this choice are the elimination of volume conduction effects, the reliability and robustness of the measure, and the ability to work effectively on a single trial basis. On the contrary, the model based techniques rely more on the success of the fitted model that usually lead to poor results especially for noisy and complex systems.

As future work, the idea of hyperscanning needs to be studied using the advances of this chapter. In EEG hyperscanning, the main idea is the simultaneous recording of the brain activity from more than one subject during a cooperative task [7][4][3]. The connectivity patterns are used to estimate the coherency between the subjects’ brain functions in the cases of cooperative and competitive tasks. Highlighting the similarities and/or differences can lead to deeper understanding of the human interaction in social applications.
Figure 2.10: Connectivity map of ImSCoh averaged over 10 trials for the simulation scheme of Fig. 2.4 for (a) SNR=-5dB and (b) SNR=-10dB. ImSCoh is robust even if the templates that are creating the propagation scheme are under high levels of noise.
Chapter 3

Modelling using Diffusion Adaptation

Nature is full of examples of complex systems, consisted of connected agents that are cooperating towards the achievement of a goal. Various examples of biological networks have been studied through the years such as fish schools, bee swarms and bird flocks [19][22][86]. The commonality of such evolving colonies are in the interactions among the members, their grouping, and more importantly, their self-organisation. Through cooperation they are able to complete tasks that they wouldn’t been able to complete if they were making the decisions on their own. In order to achieve this they need continuous communication with the agents in their proximity. However, the information processing happens in a decentralised manner which is central to our research in this thesis.

Diffusion adaptation is a novel method that uses a network of agents to perform decentralised optimisation and learning. The agents on this network have learning capability and cooperate to minimise a global cost function in an adaptive way. These agents are linked together locally, exchanging information and cooperating in order to solve an optimisation task. The advantages of this approach compared to the centralised one are noteworthy. In the decentralised version, any link failure between the agents doesn’t result in any serious problem in operation as none of the connections carries any crucial information. On the other hand, a failure in the centralised version can lead to collapse of the network’s operation. The same applies to the case of node failure. Moreover the collaboration between the agents provides us a better solution to the optimisation problem as we will see in a next section.
3.1 Motivation

The main objective of diffusion adaptation is to estimate a parameter vector \( \mathbf{w}^o \) by minimising a global cost function \( J_{\text{glob}}(\mathbf{w}) \),

\[
\mathbf{w}^o = \arg \min_{\mathbf{w}} J_{\text{glob}}(\mathbf{w}).
\]  

The information about the cost function \( J_{\text{glob}}(\mathbf{w}) \) is shared over a network of \( N \) nodes that are linked by edges that allow them to exchange information bidirectionally. Therefore, the cooperation between the nodes is valuable for the better estimation of \( \mathbf{w}^o \). The combination of the impartial information that every node can access and process lead us to convergence in a global estimate even if the information from individual local agents cannot lead to such optimisation.

In the next sections the focus will be on networks of agents that are connected with noiseless links. An example of a network like this can be seen in Fig. 3.1. In this specific network, the number of nodes is \( N = 5 \). Each node is connected with a number of other nodes of the network. The number of one node’s links to the others is defined as the node degree. For example node number 4 has degree 3 whereas for node number 1 the degree is 1. Also, each node forms a neighbourhood with the nodes that is connected to. A neighbourhood of a node \( k \) is denoted as \( \mathcal{N}_k \) and is defined by the set of the nodes that \( k \) is connected to, as well as \( k \) itself. For example, the neighbourhood of node number 4 is \( \mathcal{N}_4 = \{1, 3, 4, 5\} \) and for node number 1 is \( \mathcal{N}_1 = \{1, 4\} \).

Therefore the neighbourhood information gives us an insight of how well connected a node is. A link between two nodes is bidirectional which means that the information can travel in both directions. For example, a node \( k \) can receive information from node \( \ell \) as well as node \( \ell \) can receive from node \( k \). Moreover,
these links are weighted with non-negative values that quantify the level of reliability of the connection between the two nodes. These weighted links are denoted as $\alpha_{k\ell}$ for information flowing from node $k$ to node $\ell$. Generally, the weights $\alpha_{k\ell}$ and $\alpha_{\ell k}$ don’t need to be equal and a weight $\alpha_{k\ell} = 0$ represents the absence of interaction from node $k$ to node $\ell$. In addition, we can define $\alpha_{kk}$ as the amount of information of node $k$ will be considered on node $k$ for the optimisation process.

Diffusion adaptation can be used in many applications when an unknown vector $w^o$ needs to be estimated over a network of nodes. This vector can point to position of the target a group of agents want to approach (fish example) or just the parameters of a model that need to be estimated. In this thesis, the emphasis will be mainly on the modelling of the brain function during a specific task. However, even the fish example can serve as a similar scenario and it will be used in the following sections to make the objectives of this work clearer.

3.2 Diffusion Least Mean Squares

In this section we will formulate the diffusion version of least mean squares (LMS) for a network of $N$ nodes that are connected by links that allow them to exchange information bidirectionally. Our main objective is to estimate the unknown vector $w^o$ by having access to the time realisations $\{d_k(i), u_{k,i}\}$ of the data $\{d_k, u_k\}$ where $d_k(i)$ is the target scalar measurement at node $k$ and time instant $i$ and $u_{k,i}$ is a $1 \times M$ row regression input vector at node $k$ and time instant $i$. The input and the target data are related to each other by using the adaptive model parameter vector $w^o$ via the following equation [72]:

$$d_k(i) = u_{k,i}w^o + v_k(i) \quad (3.2)$$

where $v_k(i)$ is the noise vector for node $k$ at the time instant $i$. On a local basis our goal is to minimise the mean square error (MSE) of the following form:

$$J_k(w) = E[d_k(i) - u_{k,i}w]$$

Expanding this over multiple nodes, a global cost function can be achieved and written as:

$$J_{glob}(w) = N \sum_{k=1}^{N} J_k(w) \quad (3.4)$$
Therefore, the solution to our problem requires minimisation of the above global cost function in a distributed way across all nodes of the network [71]. Before solving this problem we go back to the Eq. (3.3) which can be rewritten in the following way:

\[ J_k(w) = E[d_k(i) - u_{k,i}w]^2 = \]
\[ = \sigma_{d,k}^2 - w^* \mathbf{r}_{du,k} - \mathbf{r}_{du,k}^* w + w^* \mathbf{R}_{u,k} w \]  

(3.5)

(3.6)

where \(\sigma_{d,k}^2\), \(\mathbf{r}_{du,k}\) and \(\mathbf{R}_{u,k}\) are the second order moments of \(\{d_k(i), u_{k,i}\}\) and can be expressed as:

\[ \sigma_{d,k}^2 = \mathbb{E}[|d_k(i)|^2] \]  

(3.7)

\[ \mathbf{r}_{du,k} = \mathbb{E}[d_k(i)u_{k,i}^*] \]  

(3.8)

\[ \mathbf{R}_{u,k} = \mathbb{E}[u_{k,i}^* u_{k,i}] \]  

(3.9)

If we want to express the optimal solution \(w^o\) using the moments above, both sides of Eq. (3.2) can be multiplied by \(u_{k,i}^*\):

\[ \mathbb{E}[u_{k,i}^* d_k(i)] = \mathbb{E}[u_{k,i}^* u_{k,i}] w^o + \mathbb{E}[u_{k,i}^* v_k(i)] \]  

(3.10)

\[ \mathbf{r}_{du,k} = \mathbf{R}_{u,k} w^o \]  

(3.11)

\[ w^o = \mathbf{R}_{u,k}^{-1} \mathbf{r}_{du,k} \]  

(3.12)

as we can assume that the covariance matrix \(\mathbf{R}_{u,k}\) is positive-definite and the noise \(v_k(i)\) is independent of the input \(u_{k,i}\). The expression of \(w^o\) coincides with the solution of the Wiener-Hopf equations that can be obtained by taking the gradient of the local cost function \(J_k(w)\). As it can be seen from Eq. (3.6) the cost function is strictly convex as the covariance matrix is positive. Therefore, to solve this problem locally a simple algorithm like LMS can be used:

\[ e_k(i) = d_k(i) - u_{k,i}w_{k,i-1} \]  

(3.13)

\[ w_{k,i} = w_{k,i-1} + \mu_k u_{k,i}^* e_k(i) \]  

(3.14)

where in the first equation the error for each time instant is estimated and on
the second equation this error is used to estimate the new filter coefficient \( w_{k,i} \). However, each node has its own noise profile with different variances. This means that each node will perform differently in respect to the convergence of the minimum difference between \( w_k \) and \( w^o \). On the contrary, in the diffusion case we are trying to collectively minimise the global cost function and therefore the performance improves across the network. This can be expressed as:

\[
\min_{w} J^{glob}(w) = \min_{w} \sum_{k=1}^{N} J_k(w) \tag{3.15}
\]

Using \( w^o \), the Eq. (3.6) can be rewritten as:

\[
J_k(w) = (\sigma^2_{d,k} - r_{du,k}^* R_{u,k}^{-1} r_{du,k}) + (w - w^o)^* R_{u,k} (w - w^o) \tag{3.16}
\]

In this equation the term in the first parenthesis is not related to \( w \) and the second term is equal to zero when \( w = w^o \). So, the first term can be replaced with \( J_{k,min} \) and rewrite the equation in a more compact form as:

\[
J_k(w) = J_{k,min} + \|w - w^o\|_{R_{u,k}}^2 \tag{3.17}
\]

where \( \|x\|_{R_{u,k}}^2 = x^* R_{u,k} x \). In order to introduce the concept of diffusion to our problem, the coefficients \( c_{k\ell} \) are introduced. These coefficients need to be non-negative and for the neighbourhood of each node \( k \), their sum must be normalised to one. For any node \( \ell \) that doesn’t belong to the neighbourhood of \( k \), the coefficient \( c_{k\ell} \) is zero. These properties can be expressed in the following equations:

\[
c_{k\ell} \geq 0 \tag{3.18}
\]

\[
\sum_{\ell=1}^{N} c_{k\ell} = 1 \tag{3.19}
\]

\[
c_{k\ell} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{3.20}
\]

The coefficients \( c_{k\ell} \) are parameters that can be defined later. They can be used to combine the local costs of the nodes in the neighbourhood of a particular node \( \ell \).

\[
J^{loc}_{\ell}(w) = \sum_{k \in \mathcal{N}_{\ell}} c_{k\ell} J_k(w) \tag{3.21}
\]

In this way the local cost for each node is estimated using the weighted sum of the
local costs of its neighbours introducing diffusion to our method. By replacing $J_k(w)$ from Eq. (3.17):

$$J_{\ell}^{\text{loc}}(w) = \sum_{k \in N_\ell} c_{k\ell} J_{k,\min} + \sum_{k \in N_\ell} c_{k\ell} \| w - w^\alpha \|_{R_{u,k}}^2$$  \hspace{1cm} (3.22)

$$J_{\ell}^{\text{loc}}(w) = J_{\ell,\min}^{\text{loc}} + \| w - w^\alpha \|_{R_\ell}^2$$  \hspace{1cm} (3.23)

where

$$J_{\ell,\min}^{\text{loc}} = \sum_{k \in N_\ell} c_{k\ell} J_{k,\min}$$

and

$$R_\ell = \sum_{k \in N_\ell} c_{k\ell} R_{u,k}$$

Now, to estimate the global cost we go back to Eq. (3.4) and rewrite it as:

$$J^{\text{glob}}(w) = \sum_{\ell=1}^N J_{\ell}^{\text{loc}}(w)$$  \hspace{1cm} (3.24)

Replacing $J_{\ell}^{\text{loc}}$ from Eq. (3.23) we get:

$$J^{\text{glob}}(w) = J_k^{\text{loc}}(w) + \sum_{\ell \neq k} J_{\ell,\min}^{\text{loc}} + \sum_{\ell \neq k} \| w - w^\alpha \|_{R_\ell}^2$$  \hspace{1cm} (3.25)

Finally as the middle term of the above equation does not depend on $w$ and the global cost can be expressed by the equation:

$$J^{\text{glob}}(w) = J_k^{\text{loc}}(w) + \sum_{\ell \neq k} \| w - w^\alpha \|_{R_\ell}^2$$  \hspace{1cm} (3.26)

The practical problem with the above equation is that one node cannot have access to all the other nodes. Moreover the matrices $R_\ell$ are not available to node $k$. Therefore, the global cost function is replaced by a local version which is closer to $J^{\text{glob}}$ than $J_k^{\text{loc}}$ is. At the same time the weighted norm is approximated by using another set of coefficients ($b_{\ell k}$) which will be defined later. The result is:

$$J_k^{\text{glob}}(w) = J_k^{\text{loc}}(w) + \sum_{\ell \in N_k \setminus \{k\}} b_{\ell k} \| w - w^\alpha \|_{R_\ell}^2$$  \hspace{1cm} (3.27)

As this version of the cost function only needs access to the neighbourhood of each
node $k$ (the only unknown vector is $w^o$ which will be replaced by an approximate later) we can apply a steepest-descent algorithm to formulate it. Therefore, a small step size parameter is introduced, denoted as $\mu_k$ (which at this stage will be considered constant through time), and the update equation can be formed as:
\begin{equation}
    w_{k,i} = w_{k,i-1} - \mu_k [\nabla \omega J^g_{k}(w_{k,i-1})]^*
\end{equation}
\begin{equation}
    w_{k,i} = w_{k,i-1} - \mu_k \sum_{\ell \in N_k} c_{\ell k}[\nabla \omega J^g_{\ell}(w_{k,i-1})]^* - \mu_k \sum_{\ell \in N_k \setminus \{k\}} b_{\ell k}(w_{k,i-1} - w^o)
\end{equation}
and from Eq. (3.16) we obtain
\begin{equation}
    w_{k,i} = w_{k,i-1} + \mu_k \sum_{\ell \in N_k} c_{\ell k}(r_{du,\ell} - R_{u,\ell}w_{k,i-1}) + \mu_k \sum_{\ell \in N_k \setminus \{k\}} b_{\ell k}(w^o - w_{k,i-1})
\end{equation}
Equation (3.30) can be divided into two equations splitting the two sums and introducing $\psi_{k,i}$ as an intermediate estimation of $w_{k,i}$ [71]:
\begin{equation}
    \psi_{k,i} = w_{k,i-1} + \mu_k \sum_{\ell \in N_k} c_{\ell k}(r_{du,\ell} - R_{u,\ell}w_{k,i-1})
\end{equation}
\begin{equation}
    w_{k,i} = \psi_{k,i} + \mu_k \sum_{\ell \in N_k \setminus \{k\}} b_{\ell k}(w^o - w_{k,i-1})
\end{equation}

The last step to reach a final form of the algorithm is to replace $w^o$ with an approximation of that. A suitable approximate of $w^o$ is the estimation $\psi_{\ell,i}$ which is already available to each node in the neighbourhood of $k$. This way the element of diffusion, is inserted in the algorithm as the estimates $w_{k,i}$ are computed using a weighted sum of $\psi_{\ell,i}$ (with $\ell \in N_k$) as the next equation shows:
\begin{equation}
    w_{k,i} = \psi_{k,i} + \mu_k \sum_{\ell \in N_k \setminus \{k\}} b_{\ell k}(\psi_{\ell,i} - w_{k,i-1})
\end{equation}
\begin{equation}
    w_{k,i} = \left(1 - \mu_k \sum_{\ell \in N_k \setminus \{k\}} b_{\ell k}\right) \psi_{k,i} + \left(\mu_k \sum_{\ell \in N_k \setminus \{k\}} b_{\ell k}\right) \psi_{\ell,i}
\end{equation}
Replacing the first and second parentheses with a new set of coefficients $\alpha_{\ell k}$ (the first term is replaced with $\alpha_{kk}$) the final form of the algorithm is achieved.
\begin{equation}
    \psi_{k,i} = w_{k,i-1} + \mu_k \sum_{\ell \in N_k} c_{\ell k}(r_{du,\ell} - R_{u,\ell}w_{k,i-1})
\end{equation}
\[ w_{k,i} = \sum_{\ell \in N_k} \alpha_{\ell k} \psi_{\ell,i} \quad (3.36) \]

The above equations can be described as the adapt-then-combine (ATC) scheme, as Eq. (3.35) is the adaptation step of the filter coefficients of node \( k \) using all the information from its neighbourhood and the Eq. (3.36) combines the filter coefficients of all the neighbourhood to estimate \( w_{k,i} \).

There are some variations of this scheme. The first one is swapping the two equations and come with the combine-then-adapt (CTA) scheme which can be written as \[ 72 \]:

\[ \psi_{k,i-1} = \sum_{\ell \in N_k} \alpha_{\ell k} w_{\ell,i-1} \quad (3.37) \]
\[ w_{k,i} = \psi_{k,i-1} + \mu_k \sum_{\ell \in N_k} c_{\ell k} (r_{du,\ell} - R_{u,\ell} \psi_{k,i-1}) \quad (3.38) \]

Both schemes use the information of the neighbourhood to perform the adaptation step. To make the algorithm simpler this can be eliminated by setting the coefficients \( c_{k\ell} = 0 \) for every \( k \neq \ell \) and \( c_{k\ell} = 1 \) for \( k = \ell \). For the ATC scheme we have:

\[ \psi_{k,i} = w_{k,i-1} + \mu_k (r_{du,k} - R_{u,k} w_{k,i-1}) \quad (3.39) \]
\[ w_{k,i} = \sum_{\ell \in N_k} \alpha_{\ell k} \psi_{\ell,i} \quad (3.40) \]

Finally, the autocorrelation and cross-correlation factors can be replaced with their approximates as practically they are not always available, especially in changing the environment that needs rapid adaptation. Then, the ATC scheme can be written as:

\[ \psi_{k,i} = w_{k,i-1} + \mu_k \frac{d_k(i) - u_k w_{k,i-1}}{\sum_{\ell \in N_k} \alpha_{\ell k} \psi_{\ell,i}} \quad (3.41) \]
\[ w_{k,i} = \sum_{\ell \in N_k} \alpha_{\ell k} \psi_{\ell,i} \quad (3.42) \]

There are specific advantages for the choice of diffusion adaptation in our method. Firstly, the diffusion LMS algorithm is proved to converge and more
importantly has improved performance compared to the non-cooperative version of the algorithm [49]. This is going to be demonstrated later using a simulated scenario. Moreover, the flexibility and robustness of this method is what it makes it so important. For example, imagine a network of nodes that cooperate to reach a goal. In the centralised example, the connections to the central processing node are crucial for the success of the system. In the decentralised example even if a node fails the system will not be affected since the other members of the neighbourhood will still be active. In addition to the flexibility, diffusion LMS (and other diffusion algorithms that will be discussed in the next sections) have the ability to constantly adapt and continuously learn, as the step size parameter is kept fixed as the time evolves. Also, the combination weights can be fixed or adaptively estimated across time in order to enhance the performance of the algorithm. In some cases these weights can be set by using a priori knowledge of the network’s structure and information sharing.

Apart from diffusion LMS there are alternative ways of solving a distributed optimisation problem, incremental and consensus strategies. The incremental strategies exploit the topology of the network and distributively estimate an unknown vector \([12][13][52][48][85]\). However, there is no collaboration step for the nodes which at each step they pass their estimate to the next node in a sequential manner. The main disadvantage over the diffusion strategy is that in diffusion adaptation each node has more data available from its neighbours (depending on the network topology). The consensus strategies, although they have various implementations \([24][27][53][74][99]\) can be expressed in a single equation as:

\[
\mathbf{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} \alpha_{\ell k} \mathbf{w}_{k,i-1} + \mu_k \mathbf{u}_{k,i}^* (d_k(i) - \mathbf{u}_{k,i} \mathbf{w}_{k,i-1}). \tag{3.43}
\]

The main difference between the consensus and the diffusion strategies lies on the fact that in consensus, the combined estimate is not used in the update. Therefore, diffusion strategies better incorporate the collective information from the neighbourhood \(\mathcal{N}_k\). Moreover, as proved in [87] diffusion strategies outperform consensus strategies. For our application, an important limitation of most of the consensus strategies is the non-constant step size. This makes the algorithm to lose its tracking ability as the step size will converge to zero when the iterations tend to infinity.

The combination weights \(\alpha_{\ell k}\) must be nonnegative and satisfy the following
conditions:

\[
\sum_{\ell=1}^{N} \alpha_{\ell k} = 1 \\
\alpha_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k
\]  (3.44)

The combination weights can be selected in various ways. Some of the strategies like Metropolis and Laplacian that are followed have been inspired by graph theory. In our first experiments a uniform scheme is used, where for each node \( k \) the combination weight is set as the inverse of the degree of the node \( n_k \) plus one \( (\alpha_{\ell k} = 1/(n_k + 1)) \). In the next section a numerical example of diffusion LMS for estimation of an unknown vector will be presented, using a network of nodes.

3.2.1 Simulations

In order to test the performance of diffusion LMS and compare it with the non-cooperative case, a simulation has been set. For this case, the application will be the estimation of the AR coefficients given a specific model. In more detail a set of data will be generated using a model of the following form:

\[
u_k(i) = u_{k,i-1}w^o + v_k(i)
\]  (3.45)

where \( u_k(i) \) is the scalar value of the signal at the time instant \( i \) on node \( k \) and \( u_{k,i-1} \) is the regression vector of \( u_k \) starting from \( i - 1 \) to \( i - M \). \( M \) is the model order and for this experiment \( M = 3 \) is chosen. The vector \( w^o \) (which represents the model coefficients) is set as \([0.7 \ 0.4 \ -0.2]\) for this experiment. Also \( v_k(i) \) is a scalar taken from a Gaussian noise vector with zero mean.

A network of \( N = 9 \) nodes is used with each node having its own noise profile. In this case, a different value of noise variance is chosen for every three nodes. We can choose any of the ATC and CTA schemes as they produce similar results. However, we choose the version of the algorithm that doesn’t include the information exchange between the nodes in order to have only the combination coefficients \( \alpha_{\ell k} \) to set. As we are looking into a simple stationary problem with no changes in the network topology and a single objective, we can use a fully connected network where each node is connected to any other node of the network. Therefore, the combination weights can take their simplest form which is the average over the neighbourhood (uniform). Also, as the neighbourhood is the
Figure 3.2: Performance of Diffusion LMS compared to the non-cooperative case. Diffusion LMS converge faster to a lower MSD as it benefits from the collaboration between the nodes of the network.

whole network, the combination step includes the averaging of the estimates $\psi_{k,i}$ from the nodes across the whole network.

The diffusion LMS was compared with the non-cooperative case where there are no connections between the nodes and each node estimates $w^o$ without sharing any information with its environment. In Fig. 3.2, the performance of both cases is compared in terms of the mean squared difference (MSD) between the estimates of $w_{k,i}$ and $w^o$.

$$\text{MSD}_{k,i} = E[\|w^o - w_{k,i}\|^2].$$ (3.46)

In order to generalise the result the average over the network is used for 100 repetitions. The result is clearly showing that diffusion LMS is outperforming the non-diffusion case.
3.3 Diffusion Kalman Filtering

In the previous section, diffusion LMS was analysed and its main advantages for the estimation of a linear model over a network of agents were presented. The LMS algorithm performs well in stationary environments and it is one of the most basic and celebrated methods of linear estimation theory. However, in real life scenarios, where the assumption of stationarity is weak, the method that is considered the best for estimating a linear model is Kalman filter \[70\]. Kalman filter is designed to be optimal for both stationary and non-stationary processes and it is widely used until today in various applications in communications.

Kalman filters are based on the state space representation of a system. As seen in the previous chapter, Kalman filtering was used in the application of MVAR estimation. The MVAR model was described with a state and an observation equation and it can be solved with accuracy as Kalman filter tracks the time evolution of the model in a fast and optimised way. In order to solve this problem in a distributed way, a diffusion version of Kalman filter was introduced in \[18\].

Let us assume that a state-space model over a network of \(N\) nodes has the following form:

\[
\begin{align*}
x_{k,i+1} &= F_i x_{k,i} + n_{k,i} \\
y_{k,i} &= H_{k,i} x_{k,i} + v_{k,i}
\end{align*}
\]  

(3.47)  

(3.48)

where \(x_{k,i}\) denotes the state vector, \(F_i\) is the the state transition model, \(H_{k,i}\) is the observation model and \(n_{k,i}, v_{k,i}\) are the state and measurement uncorrelated noises with \(Q_{k,i}\) and \(R_{k,i}\) as their covariance matrices respectively. The goal of Kalman filtering, for a non-diffusion case, is to estimate \(x_{k,i}\) and update all the required quantities based on the prior information i.e., at \((i - 1)\) of the model. \(\hat{x}_{k,i}\) is denoted as the estimate based on the prior information (a priori estimate) and set the initial state \(\hat{x}_{k,0|0} = E[x_{k,0}]\) which is usually set as zero. Also, the initial value of the covariance matrix of the estimation error can be set as \(P_{k,0|0} = E[(x_{k,0} - \hat{x}_{k,0|0})(x_{k,0} - \hat{x}_{k,0|0})^\ast]\) where, as long as \(\hat{x}_{k,0|0}\) is set to zero, \(P_{k,0|0}\) becomes the covariance matrix of the initial state \(\hat{x}_{k,0|0}\). The algorithm in its simple form can be written as:
**Measurement Update:**

\[ R_e = R_{k,i} + H_{k,i}^r P_{k,i} H_{k,i}^r \]

\[ \hat{x}_{k,i|i} = \hat{x}_{k,i|i-1} + P_{k,i|i-1} H_{k,i}^r R_e^{-1} (y_{k,i} - H_{k,i} \hat{x}_{k,i|i-1}) \]

\[ P_{k,i|i} = P_{k,i|i-1} - P_{k,i|i-1} H_{k,i}^r R_e^{-1} H_{k,i} P_{k,i|i-1} \]  \((3.49)\)

**Time Update:**

\[ \hat{x}_{k,i+1|i} = F_i \hat{x}_{k,i|i} \]

\[ P_{k,i+1|i} = F_i P_{k,i|i} F_i^* + Q_i \]

In order to move to the diffusion case, the network topology must be exploited. The goal of diffusion Kalman is estimating the unknown state \( x \) on each node \( k \) by sharing information between the node \( k \) and its neighbourhood \( N_k \). A simple solution to tackle the problem is to estimate the unknown state \( x_{k,i} \) by taking advantage of the information of node’s \( k \) neighbourhood on the measurement step. Then, the algorithm 3.64 can be rewritten as [18]:

For node \( k \):

\[ \psi_{k,i} = \hat{x}_{k,i|i-1} \]

\[ P_{k,i} = P_{k,i|i-1} \]

For each \( \ell \in N_k \) repeat:

\[ R_e = R_{\ell,i} + H_{\ell,i} P_{\ell,i} H_{\ell,i}^r \]

\[ \psi_{k,i} = \psi_{k,i} + P_{k,i} H_{\ell,i}^r R_e^{-1} (y_{\ell,i} - H_{\ell,i} \psi_{k,i}) \]

\[ P_{k,i} = P_{k,i} - P_{k,i} H_{\ell,i}^r R_e^{-1} H_{\ell,i} P_{k,i} \]  \((3.50)\)

end

\[ \hat{x}_{k,i|i} = \psi_{k,i} \]

\[ P_{k,i|i} = P_{k,i} \]

\[ \hat{x}_{k,i+1|i} = F_i \hat{x}_{k,i|i} \]

\[ P_{k,i+1|i} = F_i P_{k,i|i} F_i^* + Q_i \]

The above algorithm includes the notion of cooperation between the nodes in the measurement update phase. However there is no exploitation of the estimates of the unknown state that the other nodes carry. Following the same idea as in diffusion LMS, a set of combination coefficients \( \alpha_{\ell,i} \) is used to combine the estimates \( \psi_{\ell,i} \) (where \( \ell \in N_k \)) into the estimate \( \hat{x}_{k,i} \). Therefore, the diffusion
Kalman algorithm takes its final form [18]:

**Diffusion Kalman filter**

For every node $k$ and each time instant $i$:

$$\psi_{k,i} \leftarrow \hat{x}_{k,|i-1}$$

$$P_{k,i} \leftarrow P_{k,i-1}$$

For each $\ell \in N_k$ repeat:

$$R_e \leftarrow R_{\ell,i} + H_{\ell,i}P_{k,i}H^*_{\ell,i}$$

$$\psi_{k,i} \leftarrow \psi_{k,i} + P_{k,i}H^*_{\ell,i}R^{-1}_e(y_{\ell,i} - H_{\ell,i}\psi_{k,i})$$

$$P_{k,i} \leftarrow P_{k,i} - P_{k,i}H^*_{\ell,i}R^{-1}_eH_{\ell,i}P_{k,i}$$

end

$$\hat{x}_{k,|i} \leftarrow \sum_{\ell \in N_k} \alpha_{\ell,k}\psi_{\ell,i}$$

$$P_{k,|i} \leftarrow P_{k,i}$$

$$\hat{x}_{k,i+1|j} = F_i\hat{x}_{k,|i}$$

$$P_{k,i+1|j} = F_iP_{k,|i}F_i^* + Q_i$$

(3.51)

with initial values of $\hat{x}_{k,0|0}$ equal to zero and $P_{k,0|0}$ the covariance matrix of $\hat{x}_{k,0|0}$. Also, the combination coefficients satisfy the conditions from Eq. (3.44).

In the above algorithm after the initialisation of $\hat{x}_{k,i}$ and $P_{k,i}$, each node communicates with their neighbourhood and transfer the measurements $y_{\ell,i}$, the matrices $H_{\ell,i}$ and the covariance matrices $R_{\ell,i}$. After this step, the estimates of the unknown state are combined in the diffusion step. Finally, the new estimates of $\hat{x}_{k,i+1|j}$ and $P_{k,i+1|j}$ are computed for use at the next time instant ($i + 1$). Something that can be noted for diffusion Kalman algorithm is the fact that $P_{k,i+1|j}$ is not the covariance matrix of the state estimation error any more as the state estimators are affected by the diffusion step. In the next section, diffusion Kalman will be tested in different scenarios and will be compared with the LMS filter of the previous section.

### 3.3.1 Simulations

In this section, the performance of diffusion Kalman algorithm will be tested and will be compared with the diffusion LMS algorithm. In order to keep the consistency and compare the two algorithms with the same terms, the application
Figure 3.3: Performance of diffusion Kalman compared to the non-cooperative case and the LMS equivalents. Diffusion Kalman converges faster and at a lower steady state error compared to the non diffusion version and the LMS equivalents.

of estimating the AR coefficients will be used as before (see Eq. (3.45)). Moreover, as the interest is in the tracking the changes of the underlying models, another simulation will be presented. In that case the focus will be on the ability of the algorithms (LMS/Kalman) to converge fast enough to track the dynamic changes of the model. In order to estimate the AR coefficients using the Kalman filter, the generalised state-space model that is shown in Eq. (3.63) needs to be modified. The changes to Eq. (3.63) are:

1. Simplifying the model by replacing the matrix $F_i$ with the identity matrix.
   This leads to a simpler state equation where we assume that the vector of coefficients $x_{k,i}$ is following a random walk process.

2. Reducing the dimension of $y_{k,i}$, $v_{k,i}$ from vector to scalar and $H_{k,i}$ from matrix to vector in order to match the equation of an AR model (Eq. (3.45)).
After these changes, the state-space equations can be written as:

\[ x_{k,i+1} = x_{k,i} + n_{k,i} \]
\[ u_k(i) = u_{k,i-1}x_{k,i} + v_k(i) \]  \( (3.52) \)

In order to estimate the AR coefficients the same modifications must be applied to the diffusion Kalman algorithm:

For every node \( k \) and each time instant \( i \):
\[ \psi_{k,i} \leftarrow \hat{x}_{k,i|i-1} \]
\[ P_{k,i} \leftarrow P_{k,i|i-1} \]

For each \( \ell \in N_k \) repeat:
\[ r_e \leftarrow r_{\ell,i} + u_{\ell,i-1}P_{k,i}u_{\ell,i-1}^T \]
\[ \psi_{k,i} \leftarrow \psi_{k,i} + P_{k,i}u_{\ell,i-1}^{-1}r_e^{-1}(u_k(i) - u_{k,i-1}\psi_{k,i}) \]
\[ P_{k,i} \leftarrow P_{k,i} - P_{k,i}u_{\ell,i-1}^{-1}u_{\ell,i-1}^TP_{k,i} \]  \( (3.53) \)

end
\[ \hat{x}_{k,i+1|i} \leftarrow \sum_{\ell \in N_k} \alpha_{\ell k}\psi_{\ell,i} \]
\[ P_{k,i+1|i} = P_{k,i} + Q_i \]

In the first set of experiments, diffusion Kalman filter is compared with the non-cooperative case and the curve of the MSD is shown in Fig. 3.3. For the specific figure, the average over the nodes of the network after 100 repetitions is taken. The results clearly show the superiority of the diffusion Kalman over the non-cooperative case (where simply the weighted adjacency matrix \( A = \{\alpha_{\ell k}\} \) is set to the identity matrix). Diffusion Kalman converges faster and with better accuracy as the information from the neighbouring nodes is exploited in an optimum way. In the same figure, the comparison between Kalman and LMS is provided, for the same experiment and a set of simulated data.

For the second set of experiments, three different consecutive AR models are used. This way, the algorithm’s capability of adapting to a fast dynamically changing environment is demonstrated. This plays a significant role in the non-stationary environments to be studied later. For the specific simulation, three consecutive AR models were built with prediction order of 100 samples...
Figure 3.4: Performance of diffusion Kalman compared to the non-cooperative case and the LMS equivalents when the target dynamically changes. The improved convergence rate of diffusion Kalman compared to the non-cooperative case benefits the accurate tracking of the changing target.

As shown, diffusion Kalman is the best choice for modelling non-stationary signals in noisy environments. In the next session, a clustering method based on diffusion adaptation is presented, by extended the algorithm to multitask problems.

### 3.4 Diffusion Adaptation on Multitask Problems

In the previous sections, the problem of estimating an unknown vector $w^o$ over a network of cooperating nodes was discussed in detail. In this section, we will generalise this problem to systems that are not limited to only one desired solution. In some cases, as the real scenarios are analysed later, it is not realistic to assume
that there is a single solution to the optimisation problem. These problems are often referred to as multi-task when the problems with a unique solution will be referred to as single-task from now on.

In the case of a multi-task problem, there is a case where the set of the optimal solutions $w^o_k$ are close to each other. In this case, a simple diffusion LMS can solve the problem and converge to a solution that is close to the optimum. However, in the general case where there is no assumption about the similarity of the solutions $w^o_k$ diffusion LMS is proven to reach a Pareto optimal solution [21]. This means that there is no solution $w^e$ that can lower a local cost in Eq. (3.4) without increasing the other local costs. In other words, the Pareto optimal solution is a compromising solution for a multi-task network when using diffusion LMS.

However, in the cases where the solutions are not close to each other, the performance of diffusion LMS drops significantly as the diffusion step of the algorithm assumes that there is a single objective. Recently, there has been an effort to tackle this problem and generalise the diffusion LMS to work efficiently over multi-task networks. Under the assumption that there is a definite number of optimal solutions $w^o_k$, the new approach aiming at the clustering of the nodes so each cluster will correspond to a single solution [20]. In order to achieve this, we need to set rules for the combination weights $\alpha_{dk}$ in an adaptive manner that will ensure that the combination weights will reflect the degree of similarity between the objective of the nodes pairwise. Consequently, the nodes that estimate a single solution have larger combination weights and have smaller weights if the distance between the estimated values of $w^o_k$ is significantly large. This notion can be summarised in the following algorithm:
Diffusion LMS with clustering for multi-task problems

For each time instant $i$ and every node $k$:

$$\psi_{k,i} = w_{k,i-1} + \mu_k u_{k,i}^*(d_k(i) - u_k w_{k,i-1})$$

Update of the combination weights based on the estimated $\psi_{k,i}$

$$q_{k,i} = u_{k,i}^*(d_k(i) - u_k \psi_{k,i})$$

$$\alpha_{\ell k}(i) = \frac{||\psi_{k,i} + \mu_k q_{k,i} - \psi_{\ell,i}||^{-2}}{\sum_{m \in N_k} ||\psi_{k,i} + \mu_k q_{k,i} - \psi_{m,i}||^{-2}}$$

Combination of weights

$$w_{k,i} = \sum_{\ell \in N_k} \alpha_{\ell k}(i) \psi_{\ell,i}$$

The equation of the combination weights in the above algorithm is the result of the following optimisation problem:

$$\min_{\alpha_{\ell k}(i)} \sum_{\ell=1}^{N} \alpha_{\ell k}^2(i) ||w^o_{k,i} - \psi_{\ell,i}||^2$$

subject to $\sum_{\ell=1}^{N} \alpha_{\ell k} = 1$, $\alpha_{\ell k} \geq 0$

$$\alpha_{\ell k} = 0$$ if $\ell \notin N_k$  \hfill (3.55)

However, the cost function of the optimisation problem 3.55 contains the optimal solution $w^o_{k,i}$ which is generally not available from the nodes. A local approximation can replace this term by estimating the value of $w^o_{k,i}$ at each node as follows:

$$w^o_{k,i} \approx \psi_{k,i} - \mu_k \nabla J_k(w_{k,i-1})$$

$$w^o_{k,i} \approx \psi_{k,i} + \mu_k q_{k,i}$$  \hfill (3.56)

where we replaced the gradient of the cost function at $\psi_{k,i}$ with an approximation at this point which is $q_{k,i} = u_{k,i}^*(d_k(i) - u_k \psi_{k,i})$ and Eq. (3.54) takes its final form.
3.4.1 Simulations

In this section a series of simulated experiments are presented in order to compare the performance of diffusion LMS for multi-task scenarios. Initially, a set of three $w_k$s is defined and assigned to specific nodes $k$ of the network. Using these weights, we generate the signals $u_{k,i}$ through an AR process similar to the one used in Eq. (3.45). In our first series of simulations, a network of 9 nodes is used where a different $w_k$ is assigned to every three nodes. Three variations of the algorithm are tested: the diffusion LMS with the selection of the combination weights for clustering (Eq. (3.54)), the diffusion LMS with the combination weights set to $1/N$ (where $N$ the total number of nodes) and a non-diffusion LMS where there is no cooperation between the nodes.

For the first simulation scenario, three optima were selected in a way that they are close to each other in $\mathbb{R}^3$. In Fig. 3.5, the average MSD over 100 repetitions is presented. As it can be clearly seen the simple diffusion case performs well as
Figure 3.6: Performance of diffusion LMS with clustering compared to diffusion LMS and the non-cooperative case in the general case of three optima with relative large distances between them. In this case, simple diffusion LMS fail to converge to a low MSD as the combined information across the network of nodes with significantly different optima is not beneficial. On the contrary, diffusion LMS with clustering identifies the clusters and converges to a lower MSD compared to the non-cooperative case.

it converges to a Pareto optimum that is close to all the three optima we set a priori. So, diffusion LMS exploits the information from all the available nodes of the network and reaches to a lower MSD while the clustering version of diffusion LMS clusters the nodes via the combination weight setting and performs similarly but significantly better than the non-cooperative case.

However, in the general case where $w_0$'s are not similar to each other, the results are quite different. In Fig. 3.5, the average MSD over 100 trials is presented. As it can be clearly seen, in contrast to the previous example, the diffusion LMS it performs poorly, as it tries to converge to a solution taking into account all the data from the nodes that are generated based on three different weights $w_0$. On the contrary, the diffusion LMS with the clustering capability outperforms all the variations of the algorithm as it automatically combines the nodes that corre-
Figure 3.7: Values of the weighted adjacency matrix for a time instant $i$ for the clustering diffusion LMS case. Blue colour represents zero weight and red close to one. The neighbourhoods for nodes 1-3, 4-6 and 7-9 are clearly formed.

correspond to the same $w_k^i$. This can be seen in Fig. 3.7, where the nodes are grouped in groups of 3 in total compliance with the initial conditions of the experiment.

In the case of a network with more nodes, diffusion LMS with clustering performs even better as there is more information to be exploited. This can be seen in Fig. 3.8 where the MSD of diffusion LMS with clustering falls below -40 dB on a network of 60 nodes with the same three tasks as before.

In this section, a solution for multi-task networks was given. The results show that by clustering of the nodes for each task, the performance of diffusion LMS improves significantly. This implies that selection of the combination weights within the network affects the convergence and thus the performance of diffusion LMS. In the next section, the above diffusion adaptation methods will be exploited in order to build a method for the modelling the brain functionality during a specific task.
3.5 Modelling using Diffusion Adaptation and Brain Connectivity

In this section, the first steps towards modelling specific tasks using diffusion adaptation and EEG brain connectivity are made. The main motivation for this novel method is the exploitation all the physiological information that we acquire from EEG signals for building detailed models through time and space. These models will help describing, analysing or classifying brain responses to specific simple and complex tasks.

The core of the method is the diffusion adaptation algorithm that generates the model that connects two signals: the input of each node and the desired signal which can be locally set for each node or globally for every node. In other words, the adaptive filters estimate the linear model between these two signals through time and space. Thus, the choice of the desired signal plays a crucial role in changing the performance of the method. The important feature a desired signal must carry is the particular information that characterises a specific class.

Figure 3.8: Performance of diffusion LMS with clustering compared to diffusion LMS and the non-cooperative case on a large network of 60 nodes. As the network contains more nodes than the previous simulation, diffusion LMS with clustering performs better as there is more information to estimate the optima.
In most of cases in this thesis, the desired signal will be an average of the signals that corresponds to a specific class.

Techniques of modelling in a similar manner have been proposed in the past using simple Kalman filters [1]. The novelty of the contributed method is the use of diffusion adaptation in the modelling process as well as the exploitation of brain connectivity. Therefore, the diffusion strategy exploits the space-time characteristics of the measured signals more fully than the existing methods. Our goal is therefore, to create a model that corresponds to a specific motor or mental task by defining the network topology from the brain connectivity information.

Suppose we acquire $N$ (EEG) signals $\{u_k(i)\}$ from $N$ electrodes. We can form a network with $N$ nodes. Each node has access to time realisations $\{d_k(i), u_k,i\}$, where $\mathbf{u}_{k,i}$ is a $1 \times M$ row vector formed as:

$$
\mathbf{u}_{k,i} = \begin{bmatrix}
    u_k(i) & u_k(i-1) & \cdots & u_k(i-M+1)
\end{bmatrix}
$$

and $d_k(i)$ is a scalar taken from the local desired signal. If we store the data from all the nodes into two global matrices $\mathbf{U}$ and $\mathbf{d}$ the objective can be written as the following global optimisation problem:

$$
\min_{\mathbf{w}} E\|\mathbf{d} - \mathbf{U}\mathbf{w}\|^2
$$

Using diffusion adaptation, a new estimate for $\mathbf{w}_k^o$ is computed for each time instant, given the values of $\{d_k(i), \mathbf{u}_{k,i}\}$ as well as the neighbourhood information and the combination weights $a_\ell k(i, f)$. The latter two are computed by the chosen connectivity measure for every time instant for a specific frequency band. In order to test the method’s performance, two sets of simulations will be presented. The first attempt to solve this problem uses a simulation scheme to generate the data and diffusion LMS combined with DTF for the model building. The second scenario includes a more realistic signal generation mechanism (similar to the one that was used in section 2.4) and diffusion Kalman filtering which offers better performance for nonstationary cases.

3.5.1 First Scenario (Diffusion LMS-DTF)

In this scenario, we use a diffusion LMS paired with DTF in order to evaluate the method’s performance and robustness. The method was published in [29]. For that we generate two sets of signals that correspond to two different classes.
Figure 3.9: An illustration of a brain connectivity pattern. EEG signals collected at the marked electrodes are used to train a cooperative network to estimate a model for different tasks.

Figure 3.10: Dataset generation of two classes for the first scenario. Two trials are isolated and delayed with 5 different delays. The dataset is generated by adding Gaussian noise to the delayed signals.

The way that this can be achieved is illustrated in Fig. 3.11. Two trials \( s_1 \) and \( s_2 \) are isolated from the BCI competition II dataset \([64]\), one for each class. For the specific dataset the two classes are imaginary left and right hand movements. Then, each trial is delayed by five different time intervals simulating a propagation scheme. Gaussian noise is added to all the signals to add perturbation to the system. The resulted signals form a network of 5 nodes. Diffusion LMS is then applied to the nodes of the network using the signals \( u_{k,i} \) as the input and the signals \( s_1 \) and \( s_2 \) as desired outputs for each class accordingly. In order to train
the model a dataset with relatively low noise for each class is generated and we estimate $w_{k,i}$ which is an estimation of $w^T_k$. As an example, the combination weights of diffusion LMS are defined using the DTF measures. At each time instant, a MVAR model is fitted to the data in the following way:

$$x_i = \sum_{m=1}^{p} A(m)x(i - m) + e_i$$  \hspace{1cm} (3.60)

where $x_i$ is defined as the samples of the signal $u_{k,i}$ across all the network at time instant $i$:

$$x_i = \begin{bmatrix} u_1(i) & u_2(i) & \cdots & u_N(i) \end{bmatrix}$$  \hspace{1cm} (3.61)

The resulted connectivity values $\gamma_{kl}^2(f)$ from Eq. (2.21) are averaged over a frequency band of interest and used as the combination weights $\alpha_{\ell k}$. Moreover, we make sure to enforce the constraints of the combination weights ($\sum_{\ell=1}^{N} \alpha_{\ell k} = 1, \alpha_{\ell k} \geq 0$) in order to maintain the stability and convergence of diffusion LMS.

In addition, the stationarity of the specific simulation problem allows the use of non dynamic connectivity measure instead of using SDTF or ADTF.

Two sets of $w_{k,i}$s are estimated, one for each class of $s_1$ and $s_2$. Then, a testing dataset is generated in the same manner by adding different levels of noise. The estimated models $w_{k,i}$ for both classes are applied to the testing datasets one by one. The class of the testing trial is decided by the comparison between the values of the mean square error between the output of the above filtering process and the desired signal for each class ($s_1$ and $s_2$). If the mean square error between the output of the filtering process and the signal $s_1$ is less than the equivalent error of the output and the signal $s_2$ then the testing trial is classified as class 1.

The classification error for a range of noise levels is shown in Fig. 3.12. The
diffusion LMS results are compared with those of a non-cooperative version of LMS where each node has no access to its neighbours. In practice this is achieved by simply setting the matrix that contains all the combination weights $\alpha_{\ell k}$ as the identity matrix. The results clearly show that the diffusion case scores better in this task. In more detail, the diffusion LMS case exploits connectivity information to estimate a more detailed model that is proved to be more robust. Although this first scenario is not very realistic, it proves that exploiting all the space-time features and combining them in an elegant way using diffusion adaptation leads to an enhanced model that can indeed perform well for the EEG signals. In the next section, a more realistic scenario will be presented, taken into account the nonstationary nature of the EEG signals.

3.5.2 Second Scenario (Diffusion Kalman-ImSCoh)

In this section, an alternative scenario is analysed. In this more realistic approach there is a propagation scheme between a network of 6 nodes based on two different template signals. Moreover, the signals are nonstationary as the propagation...
appears only within one specific time interval. Therefore, the simulation allows more variability in signals’ dynamics compared to the previous case and the generated data resembles a real data scenario as this occurs in an EEG experiment. This simulation, along with the method that was followed, has been published in [28].

As this problem is multi-task and nonstationary, the undertaken approach needs to be changed. Although diffusion LMS performs well for nonstationary environments, diffusion Kalman performs better and the convergence doesn’t depend on any extra parameters like the step size of LMS. Moreover, in this case the single trial connectivity is needed. ImSCoh is a connectivity measure which performs well on single trial basis as it can be seen in Fig. 2.9.

The generation of the dataset follows the same scheme as in Fig. 2.4. Two template signals are created by bandpass filtering of the Gaussian noise and inserted in between two noise signals in different time intervals. By controlling the time instant, a propagation scenario can be simulated. The final signals are bandpass filtered between 1-45 Hz in order to have the same spectral features as the processed EEG data that are usually used in such applications. The propagation scheme for this experiment is the same as that we used for the ImSCoh simulation and it is shown in Fig. 2.5.

The method that is followed in this scenario is different from that in the previous section in three points: the type of the adaptive filter, the type of the connectivity measure and the way of classifying the two classes. In the adaptive filtering stage, we use Kalman filtering as its performance is proved better for nonstationary signals. The connectivity measure used in this scenario is the ImSCoh as it is parameter-free and can perform well for single-trial basis. Moreover, the classification stage is based on discriminating the classes in the filter coefficient space.

The core of the method is based on estimating the model coefficients using a state-space representation. The state-space model can be expressed as:

\[
\begin{align*}
   w_{k,i+1} &= w_{k,i} + n_{k,i} \\
   d_k(i) &= u_{k,i} w_{k,i} + v_{k,i}
\end{align*}
\]  

(3.62)  

(3.63)

where \( w_{k,i} \) denotes the state vector, \( d_k(i) \) is a scalar measurement and \( u_{k,i} \) is a \( 1 \times M \) row regression vector of the data \( \{d_k, u_k\} \) and \( n_{k,i}, v_{k,i} \) are the state
Figure 3.13: Outline of the method for the second scenario (Diffusion Kalman-ImSCoh). Here the connectivity values are obtained by ImSCoh and the classification is based on the application of LDA on the filter coefficients.

and measurement noises with $Q_{k,i}$ and $r_{k,i}$ their covariance matrix and value respectively. In order to estimate the model coefficients $w_{k,i}$ we employ the diffusion Kalman algorithm with the appropriate alterations. The algorithm can be expressed in the following equations:

**Diffusion Kalman filter for model parameter estimation**

For every node $k$ and each time instant $i$:

- $\psi_{k,i} \leftarrow \hat{w}_{k,i|i-1}$
- $P_{k,i} \leftarrow P_{k,i|i-1}$

For each $\ell \in N_k$ repeat:

- $r_e \leftarrow r_{\ell,i} + u_{\ell,i}P_{k,i}u_{\ell,i}^T$
- $\psi_{k,i} \leftarrow \psi_{k,i} + P_{k,i}u_{\ell,i}^Tr_e^{-1}[d_{\ell}(i) - u_{\ell,i}\psi_{k,i}]$ (3.64)
- $P_{k,i} \leftarrow P_{k,i} - P_{k,i}u_{\ell,i}^Tr_e^{-1}u_{\ell,i}P_{k,i}$

end

- $\tilde{w}_{k,i+1|i} = \sum_{\ell \in N_k} \alpha_{k,\ell}\psi_{\ell,i}$
- $P_{k,i+1|i} = P_{k,i} + Q_{k,i}$

The starting values are $\tilde{w}_{k,0|0} = 0$ and $P_{k,0|0}$, the a priori estimate covariance is the identity matrix of dimension $M \times M$.

The combination weights $\alpha_{k,\ell}$ are defined from the connectivity values of ImSCoh we estimate beforehand. The values of ImSCoh that are obtained from Eq. (2.51) are expressed as the functions of time and frequency. Therefore, the combination weights are dynamically changing over time and are expressed over
Connectivity Pattern for Class 1

1
2
3
4
5

Connectivity Pattern for Class 2

1
2
3
4
5

Figure 3.14: Connectivity patterns for the signal generation of the two classes. The generation scheme that is used is the same with the one described in Fig. 2.4.

a specific frequency range of interest as before. Moreover, the neighbourhoods $\mathcal{N}_{k,i}$ are related to the connectivity values between node $k$ and all the other nodes of the network. Setting a threshold and eliminating the connectivity values below this threshold leads to a more sparse connectivity matrix and therefore less populated neighbourhoods. The above modelling scheme is the main method to be used in the rest of this thesis for both simulated and real data. The benefits of using diffusion Kalman filtering combined with ImSCoh are the robustness in nonstationary environments and the ability to estimate single trial models. In order to test the efficiency of the proposed algorithm the simulation scheme of Fig. 2.4 is used. In order to create two different classes, two different propagation schemes are used. These propagation schemes are shown in Fig. 3.14. Thus, two datasets of 100 signals are generated, one for each class. The signals $d_k$ are class-specific and are obtained by averaging the trials of the generated signals $u_{k,i}$ over a class.

The connectivity values via the ImSCoh measure combined with diffusion Kalman filtering lead to a set of filter coefficients $w_{k,i}$ for every node $k$ and for each time instant $i$ for each trial. The sets $w_{k,i}$ represent the underlying model that connects the signals $d_k$ and $u_k$ on a single trial basis. In order to test the efficiency of the method and the representing ability of the model, a classification
step using only the filter coefficients $w_{k,i}$ and LDA is used. The classifier is trained using 80% of the dataset and the remaining 20% is used for testing. The resulted discrimination score is obtained by 10 times of random sub-sampling and is shown in Fig. 3.15. The time slot that the classes have significant changes in terms of connectivity patterns and template signals is from 200 samples to 400 samples. Class 2 is labelled using positive discrimination scores and class 1 negative. The method is compared with a non-cooperative method that was obtained by setting the matrix that contains all the combination weights $A = \alpha_{\ell k}$ as identity matrix.

The results in Fig. 3.15 can clearly show that the model obtained by the diffusion case can be easily and clearly discriminated in the time slot 200-400 samples. On the other hand, the non-cooperative case that is missing the connectivity information fails to discriminate the two classes as the discrimination scores are close to zero across all the time samples of the experiment. Therefore, the embedded connectivity helps the discrimination process and the obtained result proves the importance of the extra information in the classification. More-
over, the combination step of diffusion Kalman algorithm leads to highlighting the common information across the nodes and eliminates the randomness within the filter coefficient space. The results from both simulation experiments that were presented in this section validate the importance of exploiting connectivity information in modelling specific tasks.

In the next chapter, the challenge of modelling complex tasks using real EEG data is tackled. The main goal is to distinguish the part in time and space that characterises the specific task. This is not trivial as the important information might be hiding under other irrelevant activities. However, in the next chapter, a novel methodology will be described and tested where the significantly task-related information is highlighted over the coefficient space.
Chapter 4

Real Data Experiments

In this chapter, we focus on real data experiments. The application of diffusion adaptation combined with connectivity information is beneficial compared with the non-cooperative adaptive filter modelling as shown in the previous chapter. In real data experiments, modelling and classifying motor tasks is a challenging problem. The nature of the signals which are noisy and usually contain various artifacts can mislead the connectivity estimation and the modelling process. Moreover, the problem is assumed as a multi-task optimisation problem where not every task is related to the corresponding motor action. Also, as the system is dynamic and the signals generally non-stationary, the convergence to the optimum solution is converted to a continuously tracking problem. Therefore, the success of the modelling method is based on its ability to isolate the particular information that is related to the motor task through time and space.

In order to overcome these difficulties, a strategy for exploiting the relations between the filter coefficients $w_{k,i}$ in time, space and across trials is proposed. With the sufficient amount of data and trials, each task can be assigned to group of filter coefficients that follow specific trajectories at a specific time instant. Therefore, we use only these specific time instants when we want to analyse or classify a motor action. Thus, this feature gives a clear advantage compared to traditional BCI methods as only the representative to the task information is considered for the classification. This technique can also lead to valuable findings about the time evolution of the task-related brain activity as well as the corresponding propagation patterns. In the next sections, two sets of experiments will be studied. On the first set of experiments, the goal is to discriminate two circular hand movements (clockwise and anticlockwise) and on the second set, a dataset of Parkinson’s disease patients is examined.
4.1 Experiments on circular hand movement

The effectively coupled methods of diffusion adaptation and brain connectivity can help to build a detailed model incorporating dynamical changes both in space and time. In this section, first an attempt is made to model complex motor tasks and classify them using these models.

These experiments are motivated by both BCI requirements and more importantly the need for classification of complex and continuous movement related brain activities. Building a model that can discriminate these complex motor tasks is the main challenge in this section. The motor task that was chosen for these experiments is a circular hand movement. In particular, the subjects were asked to draw a circle with their right hand in clockwise and anticlockwise manners. However this is not a trivial task especially when the application is on real data. Even for a simple motor task, the information that can distinguish one specific movement lie in discrete point in time and space. Training a classifier across all the time points of the trials, as it is widely happens on most BCI methods, is not efficient and often results in random decisions. In order to overcome these limitations, the idea of trajectory voting is introduced in the next section.

4.1.1 Trajectory Voting

The idea of trajectory voting comes from the computer vision and machine learning field \([45][61][58]\). The main purpose of the method is to group similar trajectories together with respect to their relative motion through time and space. In an object tracking application, trajectory voting can help to identify the most frequent routes that the objects are taking through space and cluster them accordingly. In the application of modelling with diffusion adaptation, the trajectory voting is used to classify the coefficient space.

The trajectory voting is also motivated by what we learned in modelling a fish school discussed in the Introduction. Based on the assumption that the problem is multi-task, as there are a lot of concurrent events on different brain regions, the goal of the method is to cluster trajectories converging to a specific point at a specific time. The equivalent case in the fish example is that there are more than one nutrition point. Therefore, each agent or group of agents moves towards its desired goal (nutrition point) based on the information that it receives from its environment. Additionally, the environment must be considered sufficiently
dynamic to resemble the non-stationary nature of the EEG signals. Accordingly, the nutrition points in the fish example are considered moving at variable speeds that can be less or greater than the speed of the agents. Therefore, in order to assign each group of fish to a specific moving nutrition point the trajectory information of the fish is needed. In this way, the direction that a group of fish is travelling can be decided and corresponded to a nutrition point for within each time interval. This assignment is temporary as the environment (nutrition points) is constantly moving. Also, there is no assumption that the group of fish will reach the moving target as both their velocities are constantly changing too.

The above fish example can be equivalent to the problem of diffusion adaptation modelling. Each nutrition point represents a task in the multi-task problem of brain signal modelling considering inherent time varying tasks. The model coefficients \( w_{k,i} \), tracking the dynamical changes of the underlying model in fact resemble the fish behaviour in the fish school scenario. However, due to the nature of the problem, there is no a priori information about the number and position of the optima in the coefficient space. Therefore, a method like trajectory voting is needed to highlight the main routes the model coefficients are moving given that a number of trials of the same task are in hand. This way, specific trajectories that lie in the coefficient space can be assigned to the relevant tasks. Moreover, this provides the benefit of estimating the time intervals that the model coefficients best represent the task.

In order to describe the trajectory voting method, assume that a model is built that describes the relations between the input signals \( u_{k,i} \) and the desired signal \( d_k(i) \) which is class specific. The model is represented by the set of coefficients \( w_{k,i} \). For each time instant \( i \) a matrix \( W_i \) can be formed containing the vectors \( w_{k,i} \) for every \( k \) in the network. Thus, the size of this matrix is \( M \times N \) where \( N \) is the number of nodes/electrodes and \( M \) the model order. Then, \( W_i \) can be vectorised by stacking its columns and the result would be a column vector \( \phi_i \) with dimension \( MN \times 1 \). Therefore, all the model information from all the nodes, for time instant \( i \), is gathered in the vector \( \phi_i \).

Given the coefficients as points in an \( MN \)-dimensional space, a simple solution would be to cluster these points and try to assign them to their corresponding classes. However, this is non-trivial as the noisy environment and the high dimensionality of the problem make the classification performance inadequate. Therefore, a solution to this issue is to exploit the temporal information and ex-
Figure 4.1: Schematic description of the trajectory method representing the directions of the filter coefficients; (a) two classes (red and black) are presented by two families of different trajectories, (b) each trial is represented by one continuous curve with the points $w_{k,i}$ to be the filter coefficients. Between two consecutive points we define a segment $s^m_i$ (blue dotted line), (c) definition of the angular distance between two segments $s^m_i$ and $s^n_i$.

tend the points to segments. We define a segment $s^m_i$ between two points $\phi^m_i$ and $\phi^m_{i+1}$ of the same trial $m$ with the time difference between them $i_0$. The segment $s^m_i$ can be expressed as:

$$s^m_i = \phi^m_i - \phi^m_{i-i_0}$$  \hspace{1cm} (4.1)

Exploiting the temporal correlations between the model coefficients is an important advantage of this method. In the fish example, this is equivalent to defining the velocity of the fish and not only their positions. Their velocity and especially their directions provide an intuition to which nutrition they are heading. In the same way, the segments $s^m_i$ provide the information needed to choose the fraction of the information related to a specific task. In order to quantify the
relation between the two segments $s^m_i$ and $s^n_i$ of the trials $m$ and $n$, a distance metric must be chosen. The metric needs to be independent of the relative positions of the segments and consider only their directions in the space. This eliminates any dependence to the various scaling problems that may occur through time. Therefore, an angular distance metric is suitable for such a case. For each pair of trials $m, n$ we firstly compute the angle between the segments $s^m_i$ and $s^n_i$ as:

$$\cos(\theta_{mn}) = \frac{s^m_is^n_i}{\|s^m_i\|\|s^n_i\|}. \quad (4.2)$$

Then, the angular distance is calculated as:

$$\delta_{mn} = \begin{cases} \|s^n_i\| \sin(\theta_{mn}), & \text{if } 0^\circ \leq \theta_{mn} < 90^\circ \\ \|s^n_i\|, & \text{if } 90^\circ \leq \theta_{mn} \leq 180^\circ. \end{cases} \quad (4.3)$$

The angular distance is maximised for perpendicular vectors and minimised for vectors with the same direction. Vectors that have opposite directions are also treated as perpendicular and the angular distance between them is maximum. In order to make the metric more efficient, a Gaussian kernel is used:

$$\hat{\delta}_{mn} = e^{-\frac{(\delta_{mn})^2}{2\sigma^2}}. \quad (4.4)$$

where $\sigma$ is the standard deviation. The Gaussian kernel which is widely used in similar applications helps in scaling of the distances. For example, if the value of $\delta_{mn}$ is zero or close to zero, the value of $\hat{\delta}_{mn}$ is close to the maximum value of 1. When the value of $\delta_{mn}$ is high, the value of $\hat{\delta}_{mn}$ is closer to 0, especially for values greater than $3\sigma$. The Gaussian kernel offers better scaling for values close to 0 with a careful selection of $\sigma$. This way, the resulting value can be easily quantify the relative direction differences between the segments on the coefficient space and give a smooth and robust distance metric.

The pairwise distances $\hat{\delta}_{mn}$ need to be transformed in a meaningful way in order to detect the most representative trajectories for each segment $s^m_i$. A simple solution is to sum the values of $\hat{\delta}_{mn}$ for all the values of $m$ for each $n$:

$$v^m_i = \sum_{n \in C} \hat{\delta}_{mn} \quad (4.5)$$

where $C$ is a set that includes all the indices apart from $m$ itself. The values of
\( v_i^m \) represent the degree of common directionality of the trajectories based on the segment of trial \( m \) at the time instant \( i \). In other words, when \( v_i^m \) approaches its maximum value then the segment of trial \( m \) at the time instant \( i \) follows a trajectory that is common amongst all the trials at the specific time. By constructing a matrix \( V \) from the elements \( v_i^t \) and isolating the maxima, one can have a clear image for when the most common trajectories occur. This information is extremely valuable as it is justified in the next section.

4.1.2 Experimental setup

The experimental setup is consisted of two systems that work in synchrony. First, an EEG system is used with 28 electrodes that are positioned based on the 10-20 system (see Fig. 1.2). The system is a g.USBamp manufactured by g.tec. Active electrodes have been used and the sampling rate was set at 256 Hz. The signals were directly ported to Matlab without applying any filtering or pre-processing. The second system is a motion capture system manufactured by Xsens. One sensor was attached firmly to the subject’s hand to capture it’s movement (see Fig. 4.2). The sensor includes an accelerometer, a gyroscope and a magnetometer. The sampling rate that the data are acquired is 100 Hz. However, on the specific application we use only the information from the gyroscope in order to determine the time trials. For example, as shown in Fig. 4.2, the angular velocity in coordinate \( y \) is used to determine the time instants that the hand is on the starting point of the circular movement. Therefore, we can define the time intervals for a full circle and as the two systems (EEG and motion sensor) are synchronised we can split the EEG signals into trials with great accuracy. Each trial represents a full revolution of the hand from the top point of the circle back to the same point. The subjects were asked to repeatedly draw circles in a clockwise manner on a horizontally spaced flat paper. The first session lasted 4 minutes and then the subject was asked to repeat the experiment in a anticlockwise manner. From these two different tasks, two classes are formed by isolating 100 trials from each set of data.

In the next section, the modelling method developed for this particular experiment is described.
4.1.3 Description of the modelling and classification method

In this section, the steps of the modelling method (see Fig. 4.3) are discussed. The method starts with a preprocessing step for the raw EEG data. Then, the connectivity values are estimated using the ImSCoh and diffusion Kalman is employed to estimate the model coefficients $w_{k,i}$. The model coefficients are processed using trajectory voting and the resulted time instants are used for the classification of the two classes.

The data acquired from the EEG system are separated into trials, with the help of the data from the motion sensor, as described in the previous section. The raw EEG signals are filtered using a bandpass filter between 2 and 45 Hz. This way, we keep all the information related to the task (movement) and reject any DC coefficient as well as the 50 Hz mains interference.

The next step is to estimate the connectivity values $\alpha_{\ell k}$ using the ImSCoh measure as we discussed in previous chapter. The connectivity values are estimated in a pairwise manner for every combination of electrodes and for every trial. As previously mentioned, one of the strong advantages of ImSCoh is its robustness even on single-trial basis. In order to reduce the processing time of this demanding computation, a cluster of 20 vCPUs is used and the connectivity values of every combination of channels and every trial are estimated in a parallel way. In the next phase the diffusion Kalman filter is used to train and test the
The data are formed as $1 \times M$ row vectors in the following way:

$$
\mathbf{u}_{k,i} = \begin{bmatrix}
\mathbf{u}_k(i) & \mathbf{u}_k(i - 1) & \cdots & \mathbf{u}_k(i - M + 1)
\end{bmatrix}.
$$

\hspace{1cm} (4.6)

The choice of the desired signal $\mathbf{d}_k$ is quite important for the success of the method. The desired signal needs to be task-specific and represent sufficiently the class. Therefore, a good choice for the desired signal is to use the average of the trials of the EEG signals over a specific class. Consequently, two sets of signals are estimated, $\mathbf{d}_c^k$ for clockwise and $\mathbf{d}_a^k$ for anticlockwise hand movement. Therefore, using connectivity enhanced diffusion Kalman filtering from Eq. (3.64), the model coefficients $\mathbf{w}_{k,i}$ are estimated for every pair of trial $\mathbf{u}_{k,i}$ and its relevant desired signal ($\mathbf{d}_c^k$ or $\mathbf{d}_a^k$).

The connectivity information obtained from the ImSCoh connectivity measure is used to determine the combination weights in the diffusion Kalman algorithm. The connectivity values are obtained for every time instant for a specific frequency band. The frequency band is chosen based on the corresponding task and as it is shown later in this section, the selection affects the classification performance. Alongside with setting the combination weights, the connectivity values are used to set the neighbourhood of each node $k$. The neighbourhood of node $k$ includes all the nodes that have a significant connection with $k$ based on the connectivity measure. This is achieved by setting a threshold to the connectivity values after they have been normalised to sum to 1. If the ImSCoh value $\alpha_{\ell k}$ between two channels $k$ and $\ell$ is below the threshold, the value is set to zero and node $\ell$ doesn’t belong to the neighbourhood of $k$. In case $\alpha_{\ell k}$ is above the threshold, then, the
value is kept and node ℓ is combined with the estimates of node k in the combination step of diffusion Kalman. Keeping only the higher values of connectivity makes the modelling less sensitive to noisy interactions and at the same time the combination between the nodes become more robust and meaningful.

In order to set a classification experiment for this, we split the dataset into training and testing datasets. During the training phase, the model coefficients are estimated using the above procedure and their a-priori known class. Therefore, we have one set of model coefficients \( w_{k,i} \) for each trial of the training dataset for each class. These model coefficients evolve through time and space and their position in time-space characterises the underlying task. However, a simple distance metric between the testing sets of \( w_{k,i} \) and the equivalent training would not generally perform well. The multi-task character of the problem and the noisy environment makes the classification a non-trivial task. Moreover, the information that is related to the motor task physiologically is not expected to be prominent through the whole duration of the task.

Considering these limitations, the time instants when the classes are more distinguishable need to be identified. The trajectory voting method that was described before, can help in deciding these points in time and use them only during the classification phase of the algorithm (see Fig. 4.3). The two main advantages of this technique can be summarised as: the classification rates are improved significantly by doing the classifications at the selected time instants and the computational cost, on an already demanding algorithm, drops significantly as a small part of the time domain is used.

The trajectory voting plays a significant role in finding the timings when the classes are most distinguishable. Following the procedure of section 4.1.1 the values \( \hat{\delta}_{mn}^i \) are estimated. The maxima of \( \hat{\delta}_{mn}^i \) correspond to the trials \( m \) and \( n \) that belong to trajectories that follow the same direction at the time instant \( i \).

Collecting the values \( \hat{\delta}_{mn}^i \) into a 3D array \( V \), two arrays \( V_a \) and \( V_c \) can be formed, one for each class (anticlockwise and clockwise respectively). The maxima of the absolute difference of arrays \( V_a \) and \( V_c \) (\( \max|V_a - V_c| \)) corresponds to the most significant differences between the two classes. A significant number of these maxima are collected and the distribution of the maxima’s time instants are obtained. From the peaks of this distribution, the time instants \( t_d \) are defined. These time instants correspond to the points in time when the classes are more distinguishable in the filter coefficient space. Classifying the testing data only for
the time instants $t_d$ leads to better and faster classification results.

In Fig. 4.4, some representative results of the above procedure is shown. In Fig. 4.4 (a), the trajectory voting scores $v_i^m$ for the first trial are represented across time. The horizontal axis represents time and the vertical axis the trials that belong to the training subset. It is clear that at some time instants the trajectory voting score is high (red colour) across all the trials. At these time instants, there is high probability that the model coefficients $w_{k,i}$ follow the same trajectory across all trials. Therefore, at these time points, the class information is represented better using the model coefficients. To make sure that the classes are distinguishable, the histogram of the maxima’s time instants of the absolute difference of matrices $V_a$ and $V_c$ is obtained and shown in Fig. 4.4 (b). The resulted histogram is quite sparse, showing that the classes are distinguishable within some specific time intervals. So, the highest peaks of the histogram that are obtained will be used in the testing phase in order to improve our results and save the computation time.

In the testing phase the goal is to classify the trials of the testing subset using the training trials and the information obtained from the training phase. The method that is followed is relatively simple as each testing trial is classified based on the angular distances from the training trials of the corresponding class. For each testing trial the model coefficients $w_{k,i}$ need to be estimated in the same way as done in the training phase. However, there is no a-priori information for the task-specific desired signal $d_k(t)$. So, in order to overcome the ambiguity in selec-
Figure 4.5: All the possible combinations of the desired and input signals for the estimation of model coefficients.

The final step of the method is to establish a criterion to classify the testing trials in the two classes. The idea behind this step is to exploit the angular distances established in the trajectory voting step. As the models for the testing trial $x$ ($W_x^c$ and $W_x^a$) and those for all the training trials ($W_c^c$, $W_a^a$, $W_c^a$, and $W_c^c$) are available, the pairwise distances can be estimated using Eq. (4.3). The average of the distances between the model parameters of the testing trial and those of the overall training trials can reliably identify if the testing and training datasets have the same directions. The average of the angular distances between $W_x^a$ and the testing trials $W_a^a$ is denoted as $D(W_x^a, W_a^a)$. 

Konstantinos Eftaxias
Each testing trial \( x \) is classified to the relevant class after comparing the distances \( D \). Two conditions are followed to decide the class of each testing trial. The first condition for testing trial \( x \) being classified as clockwise can be expressed as:

\[
D(\mathcal{W}^c_x, \mathcal{W}^c_c) < D(\mathcal{W}^c_x, \mathcal{W}^a_c) \land D(\mathcal{W}^a_x, \mathcal{W}^a_a) < D(\mathcal{W}^a_x, \mathcal{W}^a_c)
\]

(4.7)

where \( \land \) denotes logical AND operator. Both inequalities are satisfied if the average angular distance between the testing trial and the clockwise training trials is smaller than that between testing and anticlockwise training trials. The difference between the two inequalities lies in the desired signals that are set for the estimation of the models. There is however the possibility that the previous inequalities are not met. In the case of classifying the testing trial \( x \) as clockwise the first inequality is most probably satisfied as we expect the distance between \( \mathcal{W}^c_x \) and \( \mathcal{W}^c_c \) to be minimum. However, there can be no reliable assumption for the second inequality (for the case of a clockwise trial). Therefore, to bypass this ambiguity, a second condition is used:

\[
D(\mathcal{W}^c_x, \mathcal{W}^c_c) - D(\mathcal{W}^c_x, \mathcal{W}^a_c) < D(\mathcal{W}^a_x, \mathcal{W}^a_a) - D(\mathcal{W}^c_x, \mathcal{W}^a_c)
\]

(4.8)

The concept behind the condition of Eq. (4.8) is the comparison of the difference between the relative distances. The right part of the inequality represents the comparison between the distances when the desired class is set as clockwise. The left part is the equivalent to the difference between the distances for the anticlockwise desired signal. For a testing trial \( x \) that belongs to the clockwise class, the difference \( D(\mathcal{W}^c_x, \mathcal{W}^c_c) - D(\mathcal{W}^c_x, \mathcal{W}^a_c) \) is expected to be negative as the distance between \( \mathcal{W}^c_x \) and \( \mathcal{W}^c_c \) is expected to be smaller than the distance between \( \mathcal{W}^c_x \) and \( \mathcal{W}^c_a \). At the same time, in case the difference between \( D(\mathcal{W}^a_x, \mathcal{W}^a_a) \) and \( D(\mathcal{W}^c_x, \mathcal{W}^a_c) \) is negative, the decision for the trial \( x \) is based on which of the two differences is smaller.

Moreover, the condition in Eq. (4.8) encompasses the condition in Eq. (4.7) and makes it redundant. Therefore, for the classification we first measure:

\[
\mathcal{D}_c = D(\mathcal{W}^c_x, \mathcal{W}^c_c) - D(\mathcal{W}^c_x, \mathcal{W}^a_c)
\]

(4.9)

\[
\mathcal{D}_a = D(\mathcal{W}^a_x, \mathcal{W}^a_a) - D(\mathcal{W}^c_x, \mathcal{W}^a_c)
\]

(4.10)
If $D_c < D_a$ the testing trial is classified as clockwise hand circular movement and if $D_c > D_a$ as anticlockwise. In order to test the efficiency of the method, we compare it to the case where there is no connectivity based combination of the nodes’ estimations. In the diffusion case, a frequency band from where the combination weights are estimated can be identified. The frequency bands have 2 Hz width over frequencies from 1 to 39 Hz. The dataset of the 100 trials for each class is split into 10 trial datasets, where 1 is kept as testing set and the rest as training. This 10-fold cross validation scheme provides the reliability and generality of the results. The experiments were performed for two healthy male subjects with ages between 20 and 30 years old. The movement was continuous, holding a pen and with their eyes closed to eliminate distractions and eye-blink artifacts.

<table>
<thead>
<tr>
<th>Frequency Band (Hz)</th>
<th>Subject #1 Classification Rate</th>
<th>Subject #2 Classification Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non Diffusion Case</td>
<td>0.53</td>
<td>0.705</td>
</tr>
<tr>
<td>1-3</td>
<td>0.605</td>
<td>0.67</td>
</tr>
<tr>
<td>3-5</td>
<td>0.61</td>
<td>0.7</td>
</tr>
<tr>
<td>5-7</td>
<td>0.65</td>
<td>0.7</td>
</tr>
<tr>
<td>7-9</td>
<td>0.635</td>
<td>0.735</td>
</tr>
<tr>
<td>9-11</td>
<td>0.6</td>
<td>0.75</td>
</tr>
<tr>
<td>11-13</td>
<td>0.595</td>
<td>0.765</td>
</tr>
<tr>
<td>13-15</td>
<td>0.575</td>
<td>0.76</td>
</tr>
<tr>
<td>15-17</td>
<td>0.59</td>
<td>0.755</td>
</tr>
<tr>
<td>17-19</td>
<td>0.56</td>
<td>0.755</td>
</tr>
<tr>
<td>19-21</td>
<td>0.55</td>
<td>0.735</td>
</tr>
<tr>
<td>21-23</td>
<td>0.575</td>
<td>0.745</td>
</tr>
<tr>
<td>23-25</td>
<td>0.625</td>
<td>0.745</td>
</tr>
<tr>
<td>25-27</td>
<td>0.62</td>
<td>0.75</td>
</tr>
<tr>
<td>27-29</td>
<td>0.605</td>
<td>0.74</td>
</tr>
<tr>
<td>29-31</td>
<td>0.63</td>
<td>0.735</td>
</tr>
<tr>
<td>31-33</td>
<td>0.585</td>
<td>0.73</td>
</tr>
<tr>
<td>33-35</td>
<td>0.585</td>
<td>0.74</td>
</tr>
<tr>
<td>35-37</td>
<td>0.595</td>
<td>0.725</td>
</tr>
<tr>
<td>37-39</td>
<td>0.575</td>
<td>0.73</td>
</tr>
</tbody>
</table>

The classification rates of the 10-fold cross validation technique for both subjects are shown in Table 4.1. The method was tested on two variations. The first was the one described above, with the connectivity values from ImSCoh
Figure 4.6: Clockwise and anti-clockwise hand circular movement classification success rates for different frequency bands using both diffusion and non-diffusion classifiers for subjects 1 (top figure) and 2 (bottom figure) respectively. The first box plot shows the scores of the non-diffusion case.
Figure 4.7: Mean differences between the classification success rates of diffusion and non-diffusion modelling for all the frequency bands. The diffusion case performs better around the bands that are related to motor tasks.

used from diffusion Kalman to model the movement task. For the second variation (non-diffusion) the matrix that contains all the connectivity values $\alpha_{lk}$ is set to identity. Therefore, there is no combination step in the Kalman filtering and each node tracks the optima independently. As it is shown in Fig. 4.6, the connectivity-based combination of the channels’ information enhances the method and the diffusion case outperforms the non-diffusion one. This can be seen in the majority of the frequency bands for both subjects. In Fig. 4.7, the average differences between the classification success rates of the two variations (diffusion and non-diffusion) are shown. The specific figure makes clear the benefit from the connectivity enhanced modelling in the classification rate. The combination step of the diffusion algorithm is to cluster the nodes and their estimations that are connected through a connectivity pattern. Consequently, the noisy information is eliminated and the common trajectories in the coefficient space are more prominent. The benefits of diffusion modelling are reflected on the time instant
selection during the training process as well as the testing process itself.

In addition, the method gives an insight into the frequency band that is more relevant to the motor task. In these experiments, the performance of the method is better in frequencies around 8 Hz and 22 Hz. This is justified by the relevant sensorimotor cortex activity and in particular the activity around the mu band (8-12 Hz) and beta band (12-31 Hz) that is related to motor tasks [55]. The classification success rate can approach 76.5% around the mu band for the second subject. This is a satisfactory result considering that the task is a single-trial single-arm complex motor task. However, the value of this novel method is not limited to the classification of tasks. It can be proved as a quite useful tool to analyse the neuro-dynamics in the brain by isolating the time instances that these measures are more prominent. Moreover, the proposed time-space modelling combined with connectivity information can make the study of prolonged motor tasks more reliable.

In the next section, a different experiment is presented. The flexibility of the method provides the opportunity to study a model that connects two different modalities. Diffusion modelling will be applied to a dataset of Parkinson’s patients that executing a simple motor task and the relation between their hand tremor and the EEG activity will be demonstrated.

4.2 Experiments using multi-modal datasets of Parkinson’s patient recordings

In this section, the diffusion adaptation based classification method that was established in the previous section is used in a different type of experiment. The dataset includes multi-modal signals acquired from 20 patients with different levels of tremor. In this experiment we capture the movement of a specific motor task through motion sensors attached to the patient’s hand and at the same time record the EEG signals from the patient’s scalp. In addition, the electromyogram (EMG) of the moving hand was recorded but it wasn’t used at this stage.

Before getting into the details of the experiment and the way the dataset analysed, some characteristics of Parkinson’s disease need to be mentioned. Parkinson’s disease is the second most common neurodegenerative disorder after Alzheimer’s disease [60]. It usually affects older people over the age of 50 and it causes tremor as the most common symptom. At late stages of the disease it can cause postu-
ral instability which can cause serious injuries to the patients. Apart from gait problems, Parkinson’s patients may suffer from neuropsychiatric disorders which in some cases are related to the excessive medication they receive in order to regulate the gait symptoms.

Parkinson’s disease is most affecting the area of the basal ganglia. The main cause of the disease is the loss of dopaminergic cells in the substantia nigra. Substantia nigra is the part of the basal ganglia that is partially responsible for motor action as well as other important brain functions such as learning. The factors that can lead to Parkinson’s vary. The most common cause that has been associated with the disease is the environmental factors, especially exposure to pesticides.

The symptoms of Parkinson’s disease do not become evident until most of the dopaminergic cells in the substantia nigra are gone. This makes the early detection a crucial matter for the treatment of Parkinson’s disease. When the condition is diagnosed the usual treatment is the intake of levodopa in order to eliminate the symptoms. Levodopa is used as the most common drug in recent years. The substance is converted to dopamine by the dopaminergic cells. However, a small percentage is absorbed by the brain and the rest can cause serious side effects. In patients that the high levels of levodopa cause serious implications in their quality of life, more drastic actions, like surgery, are required. In the case of surgery, deep brain stimulation is usually performed where an electric stimulation is applied to specific parts of the brain (mainly sub-cortex) in order to suppress the tremor effect.

4.2.1 Experimental setup

In this set of experiments, the patients were asked to perform a simple motor task. This task consists of four stages (see Fig. 4.8):

1. The patient is asked to pick up a cylindrical object with medium weight from a table and place it on a box with approximately 15 cm height located on the same table.

2. The patient then brings back the hand to its initial position (palm is touching the table) and rest for some seconds.

3. The patient raises his hand again and moves it towards the object.
The patient grabs the object from top of the box and returns it to its initial position.

The movement is voluntary and there is no specific cue for that. The stages of the movement are determined from the motion sensor output attached to the back of the patient’s hand. The motion sensor plays another important role as well. Using bandpass filtering it can quantify the level of tremor that is usually around the frequency of 5 Hz. At the same time, the EEG signals are acquired using a 10-20 EEG system of 28 electrodes as in Fig. 1.2. The third modality is the EMG acquired by special electrodes attached to the patient’s arm at three points. The one is positioned near the elbow and is acting as reference.

In this section, the EEG signals and the gait information from the motion sensor are used to build a model for the hand movement of the patients. The main purpose of this modelling is to uncover any correlations between the tremor intensity and the brain activity. The results, although in a preliminary form, verify the flexibility of the proposed novel method and its ability to elucidate specific aspects in a variety of applications.
4.2.2 Description of the modelling method

In this set of experiments, a different strategy is followed. In the previous section, the model was built based on a single modality. The EEG signals were acquired from the subject’s scalp during a motor task and they were used to build a model with the target created from the signals themselves. In the next experiment, the strategy is different as target signals are based on the gait information. Therefore, the model represents the relation between the cortical activity and the motor task itself.

For the Parkinson’s patients the importance of this relation is very significant. With this tool in hand, one can analyse the time evolution of the movement and relate it with the brain electrical activity. Consequently, some information can be uncovered about the correlation between the tremor and the brain activity. This correlation may be quantified and lead to a practical way of early diagnosis or categorising of the tremor type. However, this requires more input from clinicians particularly for selection of the patient and the data to be used which is beyond the scope of this thesis. Here, the main purpose is to establish the method and have some preliminary results to validate the assumption.

The modelling method differs from those used in the previous experiments in terms of some key points. The main difference is the choice of desired signal. In the previous case the desired signal was class specific and was estimated
from the trials’ average for each specific class. Therefore, each node had access to a node-specific desired signal and the model was built based on it. In this case, the signals from the motion sensor are counted as the desired signal (see Fig. 4.9). More specifically, a relevant signal (using one coefficient that best describes the movement) is processed and used as a global desired signal for all the nodes. However, the desired is trial specific and each trial is modelled using the corresponding gait information.

The other important point that differs from the previous application is the classification step. This example is not based on the classification of the trials but the analysis of the model itself and its relation with the physiological abnormalities. Therefore, there is no need for comparing the distances between trials or modelling using specific template signals as desired. However, a final manipulation step for the trajectory voting is needed to transform the outcome of the method to a meaningful result. The first step is processing of the gait information acquired from the motion sensor. The motion sensor captures a variety of information from the including accelerometer, gyroscope and magnetometer. For the movement along the z axis, the best best way to describe it is by considering...
the magnetometer z axis. This way, a representation of both the task and the tremor is available and is set as the desired signal.

The signal from the z axis of the magnetometer is firstly essential for determination and alignment of the trials through time. In order to achieve this, a simple one dimensional change point detection technique is used. The idea is based on the Canny method, where a Gaussian kernel is convoluted with the original signal \[16\]. The edges are simply defined by the maxima and minima of the resulted convoluted signal. The variance of the Gaussian kernel is determined by the level of noise in the data. Noisy data need less sensitive kernel to avoid false change points being detected. An example of the method’s performance is shown in Fig. 4.10.

For the preprocessing of the EEG data, there is need for removing eye blink artifacts despite the filtering used in the previous experiment. In order to achieve this, a common technique based on blind source separation (BSS) is used. BSS, normally based on independent component analysis, has been often used to restore EEG signals from noise and artifacts as in \[23\][76][91]. The Instantaneous BSS techniques such as JADE (joint approximate diagonalisation of eigen-matrices) \[17\], Infomax (information maximisation) \[10\], SOBI (second-order blind identification) \[11\], and fastICA \[38\] have been often applied achieving approximately the same results.

In this experiment we use fastICA in the preprocessing stage to restore the EEG and separate the components of the EEG signals for each subject. Then, the components with strong frontal projections are discarded (as they are likely to be the result of eye blinks) and the rest of the independent components are back projected into the electrode space to have the clear EEGs. The reconstructed artifact-free signals are then filtered and split into trials based on the edges of the gait sensor information. The movements that are isolated are: the grasping of the cylindrical object from the table until the placement on the box (task 1) and the similar task when the bottle is already on the box (task 3). Each epoch starts about 1 second before the start of each task and stops 4 seconds after. At some cases, as the movement is voluntary, the patients move faster than what they should and some epochs contain the movement of the hand back to the table as the epoch duration is fixed. However, the method exploits the information from all the epochs and these outliers do not affect the results. In some cases, the patients could not accomplish the tasks or part of the tasks. In such cases, the
trials are discarded by visual rejection.

The EEG signals are used to estimate the connectivity values the same way as before. The model is built using diffusion Kalman filter with the EEG signals set as input and the gait sensor signals as desired. Then, the model coefficients $w_{k,i}$ are used and via the trajectory method, described in the previous section the most common trajectories are generated over each subject’s trials. In the next section, some preliminary results are analysed and discussed, showing the valuable information that can be extracted from this type of representation.

4.2.3 Results

Given that the gait motion information is in hand, through a set of experiments we demonstrate how the developed diffusion adaptation approach in this thesis can model the pathways between the Parkinson’s neuro-generators and the resulting tremor. For this, three subjects were chosen who could best represent most of the Parkinson’s tremor cases. The first subject to be analysed had symptoms of action tremor through the whole task with relatively high amplitude. The second subject had symptoms of milder action tremor while the third subject had shown symptoms of mild tremor mainly on reaching out to grasp the cylindrical object. In the first round of experiments the desired signal is set as the raw motion sensor signal. The model coefficients $w_{k,i}$ are estimated and following the procedure of section 4.1.1, the values $\hat{\theta}_i^{mn}$ are calculated. As there is no need to keep the trajectory information for each trial, the average over trials $m$ and $n$ is estimated. The resulted time series depicts the level of common trajectories through time for a specific frequency band of the connectivity measure. This leads to a time-frequency equivalent representation that can easily expose the highlighted model similarities across the trials for each subject. Therefore, a strong common trajectory picked up from a representation means that there is a strong connection between the gait information and the brain activity at that specific time interval for a specific connectivity frequency band.

The results for the first subject are shown in Fig. 4.11. The top figure represents the trajectory voting across time for all the connectivity frequency bands and the bottom figure the data from the magnetometer of the motion sensor attached to the patient’s hand. It is clearly visible from the bottom figure that the subject suffers from tremor symptoms through the whole movement. However, the trajectory voting representation in the top figure does not depict any informa-
Figure 4.11: Results for subject 1 using the raw motion sensor signal as desired: (a) the trajectory voting scores across time and frequency and (b) the motion sensor signals that were used for the modelling. In this case, the low amplitude of the tremor components in the desired signal minimise the variance on the trajectory voting scores.

It actually shows the patient’s actual movement because of the existing baseline. Specifically, the first stage from 0 to less than 0.5 seconds the system pass the initialisation phase where the coefficients start from their initial value and track the first underlying target. The second part starts from 0.5
seconds to 1.5 seconds approximately where the system operates normally trying to track the underlying targets. Some peaks can be seen in lower frequencies where possibly the connectivity information groups the coefficients into common directions. However, in the next stage and until the end of this segment, the low variance baseline of the desired signal forces the system to reach maximum common directionality and lose its sensitivity to the tremor components.

In order to overcome these limitations, a filtered version of each gait signal can be used to highlight only the tremor components. Therefore, each trial signal from the magnetometer is bandpass filtered from 4 to 12 Hz and set as desired signal for the relevant set of EEG signals. The results for subject 1 are shown in Fig. 4.12. The first plot of Fig. 4.12 shows the scores of the trajectory voting technique for the model between the tremor and the EEG signals. The second plot is the filtered sensor data used for the modelling and the third is their corresponding time frequency representation (S-transform) averaged over the trials. The last plot shows the raw sensor data from the motion sensor in order to see the actual movement trajectory.

As seen from the time-frequency representation of the sensor data a tremor component around 5 Hz is present through the whole task and more intense on the arm extension stage. At the same time from the trajectory voting plot it is clearly shown that there are relatively high peaks in the frequency range around the tremor frequency. It is quite interesting to note that in the first stage of the movement the voting is quite high. This is mainly caused by the low variance of the desired signal at the beginning of the movement and consequently the trajectories are not varying too much and they all keep the same direction. This is also clearly visible in Fig. 4.14 in the subject’s 3 results.

In Fig. 4.13 the second patient’s result is shown. As it can be seen from the time-frequency representation of the motion sensor the relative power of the tremor at the corresponding frequencies is very low. This can be noticed in the first plot as well, where there are no distinct peaks that can justify the relation between the brain activity and tremor. In other words, in this case there are no tasks related to the tremor activity in the coefficient space and therefore no significantly common trajectories. The third patient’s symptoms are more severe comparing to the second patient’s symptoms and milder comparing to the first patient’s symptoms. Similarly to the first patient, some peaks can be seen around the relevant frequencies although the signal amplitudes in those frequency
intervals are not as high as those in the first case. This is justified as the symptoms of the third patient are milder than those of the first one as it can be seen from the time-frequency representation of the motion sensor.

In conclusion, the preliminary results for the Parkinson’s data prove the flexibility and efficiency of the novel method presented in this thesis. Of course, there are more steps to be taken in order to establish a reliable method for the study of such degenerative brain disease which link brain physiology to analytical methods. The method without any fundamental modification is shown that it can pick and quantify the relationship between the tremor and the brain activity. In the future, a more tailored solution can be developed and applied to this challenging problem to improve its clinically useful outcome.
Figure 4.12: Results for subject 1 using the filtered motion sensor signal as desired. (a) the trajectory voting scores across time and frequency, (b) the filtered motion sensor signals that were used for the modelling, (c) the time-frequency representation of the filtered motion sensor signals and (d) the raw motion sensor signals.
Figure 4.13: Results for subject 2 using the filtered motion sensor signal as desired. (a) the trajectory voting scores across time and frequency, (b) the filtered motion sensor signals that were used for the modelling, (c) the time-frequency representation of the filtered motion sensor signals and (d) the raw motion sensor signals.
Figure 4.14: Results for subject 3 using the filtered motion sensor signal as desired. (a) the trajectory voting scores across time and frequency, (b) the filtered motion sensor signals that were used for the modelling, (c) the time-frequency representation of the filtered motion sensor signals and (d) the raw motion sensor signals.
Chapter 5

Summary, Conclusions and Future Research

There is no doubt that there are numerous challenges and new areas to be elucidated in the field of brain activity modelling. There is also a growing interest in the scientific community in the field of brain connectivity and its relation to brain functions. A deep understanding of the brain’s functionality will help us solve several problems ranging from medical conditions \cite{79} to brain computer interfaces \cite{51}.

In this thesis, a novel method for modelling motor tasks was introduced. The method is based on diffusion Kalman filtering which is efficiently coupled with the connectivity measure ImSCoh. The method was tested on various scenarios: firstly, on simulated data generated from realistic schemes and then on two experiments with real data. The experiments on real data included a complex motor task executed by the right hand and a multi-modal dataset from Parkinson’s patients that contained both gait and brain signals. For the former set of experiments the goal was to classify two complex hand movements. The results clearly show that the method performs well and that the connectivity information enhances the model in a way that it makes the classification results even better. The latter set of real data is modelled based on the gait information acquired from the motion sensor attached to the patients’ hands. The results show that the method can uncover the relations between the tremor and the brain activity especially for patients with severe tremor symptoms.

The contributions of this work can be summarised as:

- A new connectivity measure using the imaginary part of S-transform co-
herency that gives clean and reliable results even in single-trial basis.

- The effective coupling of this connectivity measure with diffusion adaptation that its well performance was validated for various simulated and real data experiments. Moreover it is proved that the performance of the classification problems improved after inserting the connectivity information into the model.

- A trajectory voting technique which was used in order to make the classification of real data more effective by highlighting the time instances where the two classes were most distinguishable.

- A general and flexible modelling method that can be used in a number of various applications with minimum alterations.

However, there are a few open problems that their solutions can improve the performance of the modelling method and some applications that this method can be applied to:

- One important step for future research is by improving the coupling between diffusion adaptation and brain connectivity. In the modelling method studied in this thesis, the brain connectivity is inserted into the model in the form of combination weights without selecting the connectivity components that are relevant to the action. A more selective process based on the common information across trials of the same task would be beneficial for the modelling and would help increase the classification performance on the BCI application.

- Moreover, in the diffusion adaptation step, a better strategy for solving this multi-task problem is needed. A novel idea that can be applied to the specific application is the use of a regularised multi-task learning method based on diffusion adaptation. The regularisation will help the system learn to penalise any misleading information in order to cluster the nodes more efficiently based on their objective. The combination of a selective connectivity process and a regularised multi-task diffusion adaptation is the key to a more robust method, able to perform well for real data experiments.

- Many more applications can be considered for future research. First of all, the application of the multi-modal modelling on the Parkinson’s dataset can
be expanded to incorporate a quantitative analysis of the tremor. Moreover, a connection of the tremor and the connectivity pattern that is affecting it will be very valuable for the further physiological analysis.

- There is also room in expanding the motor task experiments in a multi-class classification problem by including more than two motor tasks. In this case, it will be challenging to maintain at the same levels or improve the classification scores in the multi-class scenario.

- Another interesting application which is considered as future research is the inter-subject EEG hyperscanning. In this type of experiments there are more than one subjects that are collaborating or competing during a common task. The simultaneous synchronised EEG recording can provide us with information about the change of connectivity patterns in relation to their collaboration or competition. The application of the connectivity enhanced modelling would be an interesting approach towards this goal.

Concluding, the main purpose of this project is to build a general method and prove its effectiveness for a variety of non-trivial problems. The motivation comes from the need of a modelling method that can incorporate efficiently the connectivity information during a specific task. The model is built through time and space and it effectively represents the brain dynamics. The learning properties of diffusion adaptation help towards this direction. Especially the choice of diffusion Kalman filtering which works efficiently with nonstationary signals. Moreover, the performance of ImSCoh on a single-trial basis (as demonstrated in the simulated examples in Chapter 2) is an important contribution to the method’s robustness. Finally, there is much space for improving the method, especially by introducing a regularised multi-task learning method and tailoring it for each specific application. These advances may lead to a more sophisticated model building which can overcome many limitations that modern systems have, and manage to create a more detailed view of task related brain activity. The dream of building a full model-based dictionary of motor and mental tasks is still alive.
Bibliography


Konstantinos Eftaxias 111


[64] G. Pfurtscheller and A. Schlögl. BCI Competition II dataset. Downloaded from http://www.bbci.de/competition/ii/.


