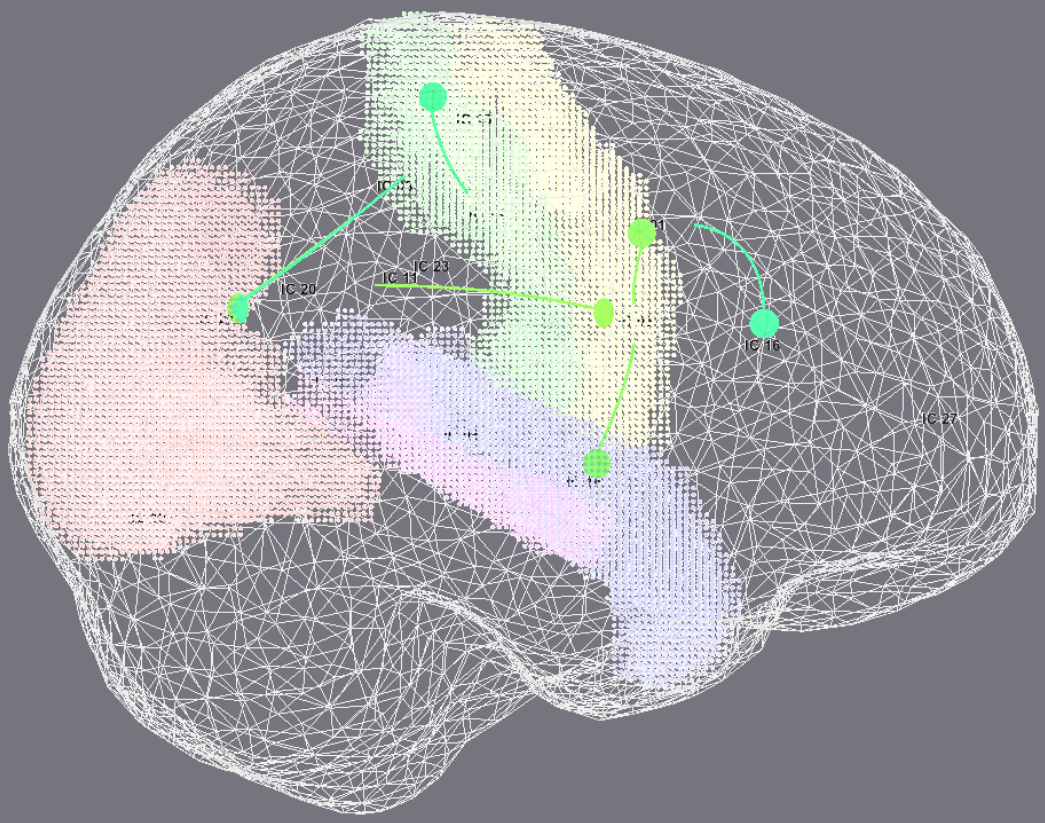


Mapping the Temporal Dynamics during Audio-Visual Speech Processing using Connectivity Analysis

Author: Caitlin Wouters

Submitted to the College of Science and Engineering in partial fulfilment of the requirements for the degree of Bachelor of Engineering (Biomedical) / Masters of Engineering (Biomedical) at Flinders University – Adelaide, Australia

Supervisor: Dr Trent Lewis

Declaration

I certify that this work does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text.

Caitlin Wouters

October 2017

Acknowledgements

I would first like to thank my thesis advisor Dr Trent Lewis of the College of Science and Engineering at Flinders University. Dr Lewis' office was always open when I had a question about my research or writing and allowed this paper to be my own work, while steering me in the right the direction whenever I needed it.

I would like to thank Mr Tyler Grummett who was involved in the experiments for the project, assisted immensely in the framework for initial processing of data and in use of the super computer colossus. I must also thank those in the brain signal laboratory at Flinders University for the accumulation of background knowledge and expertise in EEG work for a grounding to the project and laboratory practice.

Caitlin Wouters

October 2017

1 ABSTRACT

The project undertaken is 'mapping the temporal dynamics during audio-visual speech processing using connectivity analysis'. Speech processing relies on both auditory and visual sensory input. The visual stimulation observed during speech can influence what is perceived. This is an effect which is processed in well-defined regions of the brain.

The stimulus to be presented to the subjects was a series of individual stimuli of 'ABBA, 'AGGA', 'ATHA' or 'APPA'. The 'ATHA' is an artificial perception of auditory of 'ABBA' and the video of 'AGGA'. There were also static 'AGGA' and 'ABBA''s where there was the auditory sound but no visual stimulation. These were presented at three different volume levels. The data was recorded at a very high sampling rate of 9600Hz with 21 subjects.

Components were fit to the data for the connectivity measure. This component fitting was done across all subjects as a group to result in common components across all subjects for comparison and higher amount of data for stimuli. This resulted in a final set of components that are common for all subjects that were in the regions of interest.

Transfer entropy and conditional granger causality were used to provide a directed measure of connectivity. Transfer entropy was not the best measure, as none of the results were significant. Conditional granger causality was a better measure of connectivity providing some statistically significant results. To be a temporal analysis, the connectivity was taken in small 200ms windows across the post stimulus period.

An increase in the level of acoustic background noise caused there to be an increase in connection to the STS and more recruitment of the visual cortex, suggesting recruitment of STS in recognising speech in noisy conditions. There was obvious recruitment of the STS in the correct compared to little or no recruitment in incorrect. The ATHA stimulus caused there to be recognition of ABBA, as ATHA in high noise conditions, due to this pre-disposure to the sound with a different visual. This was then found to be a frontal activity that then recruited the STS and visual streams in recognising this from memory. In comparing a McGurk stimulus to a non-McGurk stimulus, it is obvious that in high noise conditions there is still this recruitment of the STS in understanding noisy speech, even when this does not change what is perceived. When there is no visual stimulation, there are still signals sent to the STS but just no output from it, however interestingly there are still signals sent from the visual cortex, which may be signals that are saying there is no visual input of significance. The STS recruitment was almost always a very initial process in the first 200-300ms, except when being accessed from memory – such as the case of guessing ATHA in a high noise ABBA.

2 TABLE OF CONTENTS

3 ADDITIONAL CONTENT

3.1 TABLE OF FIGURES

3.2 TABLE OF TABLES

4 INTRODUCTION

The processing of speech information is a complex process in the brain (McGurk & MacDonald, 1976; Van Engen, et al., 2017). The speech processing relies on both auditory and visual sensory input (Nath & Beauchamp, 2012; Poeppel, et al., 2008; Van Engen, et al., 2017). Visual stimulation can influence what is perceived (McGurk & MacDonald, 1976). This can be particularly observed when investigating the McGurk effect. The McGurk effect is when there are sounds being presented to the person with visual input, and this addition of the visual input changes the perception (McGurk & MacDonald, 1976). For example, when hearing the 'ba' syllable, with a 'ga' mouth movement, there is often a perception of 'tha' or 'da', due to the visual input changing the perception (Tiippana, 2014). This is a process which can also facilitate speech processing in noisy environments – often called the cocktail party effect.

The processing occurs in well-defined regions of the brain (Nath & Beauchamp, 2012). The auditory processing occurs in the primary auditory cortex and the visual processing occurs in the visual cortex (Barraclough, et al., 2005). There is evidence that the integration occurs in another region of the brain, the superior temporal sulcus (Nath & Beauchamp, 2012). These regions of the brain interact together, and with the prefrontal cortex, to integrate the information (Barraclough, et al., 2005). But this integration of sensory information is not well understood. These parts of the brain are known in their use, but the nature of the connection and timing is not. There is the question of which acts first and how they relay the information.

EEG is the best tool in order to investigate this problem due to its temporal resolution (Haufe, et al., 2013). EEG is recorded with electrodes on the outside of the head in order to record the electrical activity (Casson, et al., 2017). This data will be used to map the sources activating inside the head, then these sources will be used to observe the connectivity between them. There are many ways in which this connection can be investigated in the brain. One such way is using connectivity analysis (Friston, 2011). This investigated synchrony of brain sources with other sources can be analysed to determine whether there is a relation between them (Haufe, et al., 2013). This connectivity can then be mapped across the temporal domain in order to obtain this nature of the connection. These connections will be checked for statistical significance (Zalesky, et al., 2010).

5 BACKGROUND RESEARCH

5.1 ELECTROENCEPHALOGRAM

Electroencephalogram (EEG) is still used widely in brain research due to its low costs, non-invasiveness, portability and time resolution (Haufe, et al., 2013). Although EEG has very good temporal resolution, it has poor spatial resolution (Bell & Cuevas, 2012). The signals travel from their source through the brain and through the skull out to the electrodes (Casson, et al., 2017). The skull acts like a low-pass filter which retains and distorts brain signals (Bell & Cuevas, 2012). A dense array of electrodes could get better spatial resolution but can be very costly in obtaining and maintaining (Bell & Cuevas, 2012). A dense array of electrodes will be used in the project.

The other common measure to use in brain research is functional magnetic resonance imaging (fMRI) (Pittau, et al., 2011). However, fMRI can be even more costly and difficult to obtain (Sharon, et al., 2007). The signal measured is the blood oxygen level dependent (BOLD) changes occurring in the brain, which has a temporal resolution across a time frame of seconds, whereas the temporal resolution of EEG is in the milliseconds (Pittau, et al., 2011). In this study of mapping the temporal dynamics, it is important to have a good temporal resolution since the effect is to be measured across small time windows. This would be unable to be done using fMRI. While EEG has poor spatial resolution, there are still ways to work around this, and EEG has the appropriate temporal range for the study.

EEG has a number of other issues which need to be considered. There is the problem of muscle signals present in the recorded EEG (Casson, et al., 2017). These muscle signals have magnitudes much larger than the EEG signal, since it is not impeded by the skull and is a higher amplitude electrical signal (Duffy, et al., 1989). This can be reduced by the use of low frequency filters and other techniques (Duffy, et al., 1989). This brain signal is also conducted through the brain to the electrode, causing each electrode to contain a number of sources information with volume conduction errors, which makes it difficult to separate the scalp recorded EEG into individual brain sources (Kayser & Tenke, 2014). There is also electrooculography (EOG) signal from eye movement, eye blinks, electrode movement and incoming noise signals, such as 50Hz line noise (Duffy, et al., 1989). The EOG signal will be filtered by using electrodes around the eyes to pick up these signals and filter for them (Wyczesany, et al., 2015). Eye blinks, electrode movement and noise signals can be filtered out using filters within the frequency range of the noise but not EEG, by rejecting epochs with a high amount of noise or using independent component analysis to reject these sources (Wyczesany, et al., 2015).

5.2 MCGURK EFFECT

Speech processing is very complex process in the brain, which relies on both auditory and visual sensory input (McGurk & MacDonald, 1976; Van Engen, et al., 2017). This visual stimulation of what humans see can affects what is perceived (Nath & Beauchamp, 2012; Poeppel, et al., 2008; Van Engen, et al., 2017). This can be particularly observed when

investigating the McGurk effect (McGurk & MacDonald, 1976). McGurk and MacDonald (1976) found that if different acoustic speech signal was combined with a visual stimulation that is incongruent, then the listener heard a sound which was different from the sound or the visual stimulation, but a combination of the two (Tiippana, 2014; Nath & Beauchamp, 2012; Van Engen, et al., 2017). The illusion was termed as the 'McGurk effect', due to the discoverer of this phenomenon (McGurk & MacDonald, 1976). This phenomenon has had great impact in psychology and neuroscience field, as it shows the effect of the integration of information from both the visual and auditory senses into a unified, integrated perception (Tiippana, 2014; Gentilucci & Cattaneo, 2005; Stevenson, et al., 2012).

The most common use of this is to dub the auditory 'b', over a visual 'g', and the perceived is 'd' or 'th' (Tiippana, 2014; Nath & Beauchamp, 2012). There are a number of other 'McGurk' effects that can be used, such as the 'b' auditory and 'd' acoustic is heard as 'd' (Tiippana, 2014; McGurk & MacDonald, 1976; Nath & Beauchamp, 2012). One challenge with using this interpretation is that if the user responds 'd', it is impossible to determine whether they did this in response to the visual stimulation alone or the McGurk effect of the integration of the auditory and visual (Tiippana, 2014; Noesselt, et al., 2007). The audio 'b' and visual 'g' being perceived as 'd' is thought to emerge due to fusion of the features; since the place of articulation is bilabial in audition (front), in visual it is velar (back), and so alveolar is perceived (middle) (Tiippana, 2014). However, one must also take into account that the 'g' can also be easily mistaken visually for 'd' so the choice of 'd' does not guarantee that the perception was a McGurk perception (Tiippana, 2014). The McGurk effect is an excellent tool for investigating the multisensory integration in speech perception (Tiippana, 2014). This is a very useful research tool, as it reflects the strength of the integration (Tiippana, 2014; Noesselt, et al., 2007; Stevenson & James, 2009; Pearl, et al., 2009). One of the best choices is, therefore, the 'b' auditory with 'g' visual stimulus as the auditory visual integration of speech, as it has a differing result to record whether or not the participant perceived the effect. It is also one of the most used in literature and as such, a good choice, as it is very commonly used and accepted in the field (Nath & Beauchamp, 2012; Nahorna, et al., 2012; Boliek, et al., 2010; McGurk & MacDonald, 1976).

This effect is not experienced by all individuals with susceptibility noted to range from 26% to 98% (Nath & Beauchamp, 2012; McGurk & MacDonald, 1976; Gentilucci & Cattaneo, 2005). One study by Van Engen in 2017 claimed the opposite to many others; that the susceptibility to the McGurk effect does not predict audio-visual recognition. They state that the susceptibility to the McGurk effect is dependent on lip-reading ability and other external influences (Van Engen, et al., 2017). This should be taken into account in the study that if this is true then there may still be integration occurring in the McGurk non-perceived tasks and can not necessarily be used as a baseline for comparison.

The strength of the McGurk effect is taken to increase when the acoustic amplitude decreases (Tiippana, 2014; Nahorna, et al., 2012). This will be investigated in the study by making the stimulus of the study to have a differing level of volume, with a constant level of background noise of a 'cocktail party' effect, to investigate the effect of this noise level on

the McGurk effect. By using a constant background noise, the perceiver is not affected by the change in background noise as an effect as well as stimulus presentation. This has been investigated in a number of other studies; however, none have investigated the nature of the connection over time. This will be investigated in this study to have better perspective of the influence of this noise level on the nature of the connection.

The auditory processing occurs in the primary auditory cortex and the visual processing occurs in the visual cortex (Nath & Beauchamp, 2012). There is evidence by numerous fMRI, PET and other studies that the integration occurs in another region of the brain, the superior temporal sulcus (Nath & Beauchamp, 2012; Barraclough, et al., 2005; Beauchamp, et al., 2004; Stevenson & James, 2009; Noesselt, et al., 2007; Macaluso, et al., 2004). An fMRI study by Nath and Beauchamp in 2012 hypothesised that the left STS was the strongly active component in McGurk perceivers, and tested the hypothesis that the McGurk non-perceivers must have lesser activity levels in this area. They found that a weaker response in the left STS reflected lesser perception of the McGurk effect (Nath & Beauchamp, 2012). These regions of the brain interact together, along with the prefrontal cortex, to integrate the information (Nath & Beauchamp, 2012). But the nature of this integration of sensory information is not well understood, especially from the perspective of the timing of the interaction between these regions. There have been a number of hypothesis made about the way this connection works but, as of yet, it has not been investigated.

There are a number of similar EEG studies to the one to be undertaken. There was a study done by Simon and Wallace in 2017 that investigated the integration of temporal processing of asynchronous audio-visual speech (Simon & Wallace, 2017). They did a time-frequency analysis on the ERPs to investigate where the effect was occurring and when (Simon & Wallace, 2017). They found an auditory led suppression in ERP signal in audio-visual speech integration (Simon & Wallace, 2017). Another study by Kumar investigated the networks during audio-visual speech using EEG (Kumar, et al., 2016). They did a global time-frequency coherogram analysis on the event related potentials to investigate the reaction at the alpha, beta and gamma bands; pre and post stimulus (Kumar, et al., 2016). They found a heightened coherence in the gamma band and a decreased coherence in the alpha and theta bands during audio-visual integration (Kumar, et al., 2016). They stated that beta activity has been seen in top-down processing and therefore inferred that this may be what is occurring in this integration of audio-visual processing (Kumar, et al., 2016). There is a study by Wyczesany that used a similar method of investigation to this study to be undertaken but is on visual processing and the effect of emotional state (Wyczesany, et al., 2015). They used effective connectivity analysis between sources to investigate this effect (Wyczesany, et al., 2015). There are no studies in the audio-visual field that use the same level of connectivity analysis between sources acting across the temporal domain; only time-frequency analysis of ERPs of scalp electrodes to investigate the coherence across the temporal domain. There are some other studies that investigate the areas of the brain involved, as previously mentioned, which are all fMRI studies, instead of EEG, so are not in the temporal domain. This investigation would fill a gap in the field that has not been explored in this particular investigation of audio-visual integration in the temporal domain.

5.3 Connectivity Measures

EEG is often used for the studying of brain dynamics in humans (Haufe, et al., 2013). However, it is hindered by the signals from the brain being measured by numerous electrodes from other areas of the scalp and each scalp electrode measuring data from many sources within the brain (Haufe, et al., 2013). This causes finding the spatial position of the source of the signal within head difficult (Haufe, et al., 2013). Connectivity measures are methods of measuring the way sources in the brain or predefined regions of the brain interact with one another (Friston, 2011). The correct form of connectivity measure needs to be selected for the study, and as such, this is an area of interest which needs to be investigated.

The analysis of these networks is computed using different measures; structural connectivity, functional connectivity and effective connectivity (Lang, et al., 2012; Friston, 2011). Structural connectivity is when spatially known regions in the brain are used as the sources and the connectivity between these is measured (Lang, et al., 2012; Friston, 2011). Functional connectivity is the temporal dependency of patterns; it uses statistical dependencies of correlation, covariance, spectral coherence and phase-locking to determine the sources and the connections (Lang, et al., 2012; Friston, 2011). It is based on cross correlation of variance in the time or frequency domain or spectral coherence (Lang, et al., 2012). Effective connectivity describes the influence one neural system has on another and reflects the causal interactions (Lang, et al., 2012; Friston, 2011). Effective connectivity is better for a distributed system, since it investigates the influence each part has on one another, not relying on comparison of when the stimulus was presented. Functional connectivity can often be affected by the underlying neural networks of the brain; since these will be picked up in correlation (Lang, et al., 2012; Friston, 2011). Therefore, much study has been done to determine the default mode network, to determine only the effect due to the task being investigated and not just the resting state performance (Lang, et al., 2012). The default mode network is found from the resting state of a person when they are awake and alert but not actively involved in any task (Lang, et al., 2012). While performing cognitive tasks, there is a response where networks of synchronised activity occur, and they reorganise to have task orientated manner (Lang, et al., 2012). This is often referred to as the anti-correlated network (Lang, et al., 2012). When one is active, the other is less so (Lang, et al., 2012).

Using predefined areas of the brain would not be a very practical model in this study since there is no MRI or other means of accurately determining the predefined anatomical regions of each subject's brains. As such, it is best to use an effective or functional measure of connectivity, which does not need these predefined areas as input – it can be any source of activity. Both functional and effective connectivity have their advantages and disadvantages. These resting or default networks that are present all the time can be accounted for in the study by recording baseline data of a person with their eyes open and closed for a period of time to find this.

Within the field of effective and functional connectivity, there are still many different measures to be investigated. Many are looking at phase frequency, where they are all in phase then that region is assumed to be a region acting together at this point in time (Blinowska, 2011). Phase synchrony is when the phases are in the same synchrony, even if not in phase together (Blinowska, 2011). This assumes when a portion of the brain is being used, all the neurons are in phase in their function (Blinowska, 2011). A common phase synchrony measure is phase locking value (Aydore, et al., 2013). This measures the instantaneous phase of the signals and assumes connected areas work in the same phase (Aydore, et al., 2013). Phase locking value as with other phase synchrony measures, are only applicable across data which is approximately stationary over the time interval which is being measured (Aydore, et al., 2013). Correlation is a method based on using the Pearson correlation coefficient (Wang, et al., 2014). It considers time delays and can be directed or undirected measures (Wang, et al., 2014). It is a simple approach which can be directed but as a linear method also suffers the effects of false connections through indirect pathways (Ide, et al., 2007). Coherence is measured from the cross-spectral density obtained by the conjugate multiplication of the frequency domain of the difference in the signals. It is very difficult in linear predications. It is only a unidirectional interaction, and bilateral, but directional interactions are obtained by using linear predication and granger causality; such as coherent granger causality (Wang, et al., 2014). The correlation and coherence families are more susceptible to variations in structure across data (Wang, et al., 2014; Ide, et al., 2007).

The linear predications are an assumption of the time signal by a weighted value of the signal, which is derived by using the past signal; an assumption of the behaviour based in its own past. There can also be a bivariate model, which uses two signals and uses the past of both to predict each other. This takes the Granger causality of the two to predict whether they influence one another. Granger causality is an estimator of a signal based on the past of other signals in correlation. Some measures which use this estimator include Granger Causality Index (GCI), Directed Transfer Function (DTF) and Partial Directed Coherence (PDC). Granger causality index uses granger casualty to see whether the information contributed by the channel improves the prediction of the first (Blinowska, 2011). This can give misleading information, since the other channels may also have an influence on each other (Blinowska, 2011). This issue is resolved by the Granger causality measures using multivariate models; they use Granger causality across the connections to determine whether or not there is an established connection between every possible connection (Liu & Aviyente, 2012). These can only determine linear relations between each connection which can be misleading when there are nonlinear dependencies (Liu & Aviyente, 2012). This is good for noisy data but is a very computational heavy and therefore time-consuming calculation method (Wang, et al., 2014). Directed transfer function is one of these methods, which describe the causal influence of the channels on each other, taking into account the influence of others (Blinowska, 2011).

Some non-linear methods are mutual information, transfer entropy, generalised synchronisation, continuity measure, synchronization likelihood and phase synchronization

(Friston, 2011). Continuity measure, generalized synchronisations and synchronisation likelihood are very similar in their function, as they are based on the reconstruction of the signals in the phase space. Out of these measures, only transfer entropy allows for directionality (Blinowska, 2011). It is a non-linear measure of a multivariate method (Liu & Aviyente, 2012). It is used for effective connectivity in high density EEG data, which also accounts better for volume conduction (Olejarczyk, et al., 2017; Liu & Aviyente, 2012). It determines causality from a deviation of the observed data from the generalised condition (Liu & Aviyente, 2012). It can be seen as equivalent to Granger causality for Gaussian variables but requires much less computation time (Wang, et al., 2014). These non-linear methods require very large continuous segments of signal for processing (Friston, 2011). They are prone to systematic errors and very sensitive to noise (Friston, 2011). When nonlinear methods were compared with linear correlation with a noisy signal, the non-linear estimators were of poorer performance (Friston, 2011; Liu & Aviyente, 2012). Mutual information uses the time series to estimate the shared information between the sources (Wang, et al., 2014). In theory it can measure both linear and non-linear dependencies (Wang, et al., 2014). Mutual information is found to fail in some EEG data when determining the appropriate structure in tests where it is known (Wang, et al., 2014).

In comparison of bivariate and multivariate estimators, the bivariate estimators can give misleading information when the channels are interrelated (Blinowska, 2011). The signals of the sources could be acting at the same time and or a slight time delay in the reading of one of the signals and with bivariate estimators this causes dense disorganised structures of connections (Blinowska, 2011).

Many of these effective methods can have partial methods added in which partial results from the bivariate results are obtained, that are filtered by matrix inversion where the connectivity is evaluated between any two nodes while considering the influence of other nodes (Wang, et al., 2014). They suppress the connectivity when the two nodes receive input from the same source (Wang, et al., 2014).

From these many different measures, there are a number of potential measures that may be applicable. Conditional granger causality and transfer entropy were measures which are directed, commonly used and seem appropriate for this study. This takes into account the high-density data and nature of the connections required for the audio-visual speech processing. Some measures may be applicable, such as some of the other granger causality measures, but they are very time consuming and if other measures are sufficient, then they would be chosen over these. However, should the others be insufficient, they are very good for noisy or hard to define connections so would be used then. Methods to then evaluate would be multivariate granger causality, directed transfer function or possibly mutual information.

Once calculated, the graph of the connections can then be made by using a threshold from the connection value to only show those connections with a strength above a certain threshold (Wang, et al., 2014). For any of these methods, choosing the most appropriate strength threshold can be the problem (Wang, et al., 2014). Choosing a higher threshold can

show very few connections but choosing a smaller one can possibly show connections which are not statistically significant (Wang, et al., 2014). This will need to be considered when analysing the results.

5.4 ICA AND GROUP ICA

Source modelling needs to be done on the data to be able to investigate the connectivity between these sources. This takes the signals measured at the scalp and uses them in relation to one another to find where they originate from (Stone, 2002). This is done by assuming the sources are non-Gaussian signals that are statistically independent (Stone, 2002). It is called independent component analysis (ICA) (Stone, 2002). In EEG, a data driven model is used since it needs to be sources found from the data, not predefined areas (Stone, 2002). There are a number of assumptions made in this analysis in terms of propagation of the signals to the scalp (Onton, et al., 2006). It assumes a linear propagation with no refraction or time delays in signals (Onton, et al., 2006). Principal component analysis is also sometimes used, which finds temporally orthogonal directions of the data to define the sources (Onton, et al., 2006). But ICA also accounts for propagating back the data whose activity is independent from one another, allowing for a better separation of the sources (Onton, et al., 2006). There are a large number of algorithms possible in ICA and a number of main algorithms used are readily available through MATLAB toolboxes; Infomax (Bell & Sejnowski, 1995), fastICA (Hyvarinen & Oja, 1997), JADE (Cardoso & Souloumiac, 1993), AMUSE (Cardoso & Souloumiac, 1993), SIMBEC (Tong, 1991) and AMICA (Palmer, et al., 2011). Out of these it was found that AMICA was the most reliable, then Infomax and fastICA were reliable but not as well performing and SIMBEC, AMUSE and JADE did not perform reliably (Youssofzadeh, et al., 2012; Leutheuser, et al., 2013). AMICA will be the ICA used for the current study. AMICA uses a Newton method for maximum likelihood estimation of the ICA mixture model (Palmer, et al., 2011). This method accommodates for non-stationary environments and source densities in EEG by using a probabilistic mixture framework (Palmer, et al., 2011).

Most studies calculate a set of components on each subject individually using ICA and attempt to cluster ICA between subjects based on region or event responses (Kirschner, et al., 2012). To be able to get the connectivity as a mutual set of components across all the subjects, the sources need to be the same in all the subjects. This can be done by calculating and applying the ICA weight matrix across the subjects to get common sources (Huster, et al., 2015). The weights of these are then applied individually to the subjects to obtain the component time series in relation to the subject (Huster, et al., 2015). These then need to be analysed by a regression to determine if the component is present in the subject, otherwise it is discarded (Huster, et al., 2015). The common components between the subjects that are active are then obtained. This is useful in analysis as this can be used in comparison of the activity between subjects. This technique is recently used in the field, however, is not extensively used in the literature in this study of analysis of the audio-visual integration, providing a new perspective on this area of the field of research.

5.5 CONNECTIVITY STATISTICS

Once the sources have been found and the connections between these established, then the statistical significance between these need to be investigated in order to determine which are significant. This is done using network-based statistics (Zalesky, et al., 2010).

The network-based statistics can test, via some hypothesis, whether or not the observation is true. The mean of the data is computed, a standard deviation is found and from this, it is determined whether this is considered to be a valid hypothesis by performing a t test (Maris & Oostenveld, 2007; Zalesky, et al., 2010). This is inferential statistics (Maris & Oostenveld, 2007; Zalesky, et al., 2010). There is an experimental observation between two conditions. There will be a distribution of the data at one condition and the other (Maris & Oostenveld, 2007). Therefore, a t value can be found between the two conditions (Maris & Oostenveld, 2007; Zalesky, et al., 2010). This requires a known distribution of the test statistic and often the data in EEG is not normally distributed, making a t test or f test not necessarily valid (Maris & Oostenveld, 2007; Zalesky, et al., 2010). Another problem faced is multiple comparisons are needed since there are many frequencies at any point in time which need to be analysed (Maris & Oostenveld, 2007; Zalesky, et al., 2010). By having such a large number of tests, the chance of a false result is increased to a large probability, as there is an assumption of independence in the tests which is not a valid assumption (Maris & Oostenveld, 2007). There is an even higher probability of false positives, since these tests are across the difference sources or electrodes (Maris & Oostenveld, 2007; Zalesky, et al., 2010). The Monte-Carlo approximation is the randomisation of distribution using the maximum statistic (Maris & Oostenveld, 2007; Zalesky, et al., 2010). The Monte-Carlo approach takes the two sets of data and combines them in different ways to get the distribution and then uses this distribution for the statistical analysis (Maris & Oostenveld, 2007; Zalesky, et al., 2010). This can have different shapes of distribution but can still be used to analyse whether these falls in the distribution (Maris & Oostenveld, 2007; Zalesky, et al., 2010). This is non-parametric statistics, since it has randomisation of the independent variable and the hypothesis is about the data not a specific parameter (Maris & Oostenveld, 2007; Zalesky, et al., 2010).

The Bonferroni correction can be utilised here, which takes into account the number of tests done and alters the significance of the threshold accordingly by dividing it by the number of tests performed (Maris & Oostenveld, 2007). The problem with using this is with a large number of tests, this makes the threshold extremely small, so there will have to be an extremely strong effect to register, increasing a probability of a false negative (Maris & Oostenveld, 2007; Zalesky, et al., 2010). Another method is to use false discovery rate by controlling the expected proportion of false positives making it similar to Bonferroni correction but more sensitive (Maris & Oostenveld, 2007; Zalesky, et al., 2010).

To avoid the multiple comparison problem, rather than testing everything, only the most extreme condition can be tested (Maris & Oostenveld, 2007; Zalesky, et al., 2010). A randomisation distribution is done for the most extreme statistic (Maris & Oostenveld, 2007; Zalesky, et al., 2010). To increase the sensitivity, it is conventional to do a univariate

parametric approach where the data is considered at many channels, time points and frequencies (Maris & Oostenveld, 2007; Zalesky, et al., 2010). This increases sensitivity since the channel, frequency and time points are not independent as they show similar behaviour (Maris & Oostenveld, 2007; Zalesky, et al., 2010). Neighbouring samples will show similar behaviours, so incorporating all of them means that this is taken into account by accumulating evidence; cluster-based statistics (Maris & Oostenveld, 2007; Zalesky, et al., 2010). To avoid multiple comparison problems, the largest observed cluster can be compared to the randomised distribution of the largest clusters (Maris & Oostenveld, 2007; Zalesky, et al., 2010).

In the current study, controlling false positives needs to be considered, but also reducing the false negative rate (Maris & Oostenveld, 2007; Zalesky, et al., 2010). The solution is to have a multi-comparison problem, where testing one hypothesis per source, in time and at frequency level, and do this for a number of different comparisons (Maris & Oostenveld, 2007; Zalesky, et al., 2010). But it is one hypothesis per all the data to be tested (Maris & Oostenveld, 2007). To increase the sensitivity the study should be using cluster-based statistics or multivariate analysis (Maris & Oostenveld, 2007; Zalesky, et al., 2010).

5.6 Purpose of Study

The deeper understanding of brain networks and the way the brain integrates information, helps to be able to understand and classify the usual behaviour of the brain for audio-visual integration. This could then be able to be used in further research to determine how it differs in disease or disorder cases. This could be used as both a diagnosis tool and to be able to better understand the disorder or damage done and what this causes to happen. Audio-visual integration has been shown to identify or investigate many clinical groups, such as those with autism spectrum disorder (Irwin, et al., 2011; Woynaroski, et al., 2013; Boliek, et al., 2010; Ujiie, et al., 2014), schizophrenia (de Gelder, et al., 2003; Pearl, et al., 2009), stroke patients (Hamilton, et al., 2006), and dyslexia (Blau, et al., 2009)

6 METHODS AND ANALYSIS

6.1 STIMULUS

The stimulus to be presented to the subjects was a series of individual stimuli of 'ABBA', 'AGGA', 'ATHA' or 'APPA'. The 'ATHA' is an artificial perception of auditory of 'ABBA' and the video of 'AGGA'. There was also a static 'AGGA' and 'ABBA', where there was the auditory sound but no visual stimulation.

The 'ATHA' stimulus was presented to the subjects as the McGurk stimulus to test the perception and whether there was a change in perception and to ensure they were attending both the auditory and visual stimuli. A 'correct' perception of 'ATHA' suggests that the participant has fused the information from both modalities. The 'ABBA' and 'AGGA' were therefore presented in order to challenge this stimulus. 'APPA' was included as an opposition to 'ABBA', so that the person had to pay attention to the sound. With only 'ABBA' and 'AGGA', the subject would be able to tell which was coming just by seeing the lips come together for 'ABBA'. Then the static sounds were added in to able to later test the effect of visual or no visual on the connection.

There was a background 'cocktail party' noise, which was made of vague background talking sounds and other background noises. This was chosen to have some syllable like speech sounds in it, to further concentrate on the stimulus, but also have some conflicting background non-speech to make it more difficult a task. The stimulus was be presented at three different volumes with a constant background noise; low noise of 100% volume, medium noise of 75% volume and low noise of 50% volume, while the background noise was at a constant 50% volume. This choice was made to alter the stimulus level, rather than the actual background noise level, to keep the background as constant and not add another stimulus in there that the subject may be reacting to along with the actual stimulus presented.

This was presented in 10 blocks. There were 72 stimuli per block; the number of sound levels by the number of stimulus by 4; to results in at least 4 of each stimulus types, per block. This results in 40 of each stimulus, per person, across the experiment. The duration of the stimulus presentation was around 45 minutes. This was chosen to be the longest amount of data to be able to be taken since a lot of data is needed for connectivity. However, any longer than this and it is unreasonable to ask for the participation and concentration of subjects. The stimulus was presented using the software SNAP (Kothe, 2013), written in Python.

6.2 EXPERIMENT

The experiment was conducted in the multimodal recording facilities at Tonsley. The recording of EEG signal was done in the faraday cage, Figure 1, while the presentation was controlled outside, Figure 2. LabRecorder (Stenner, et al., 2018) was used to record the

different layers of incoming information for the test; EEG data, triggers, and eye tracker data. This information can be seen on the screen in Figure 2.

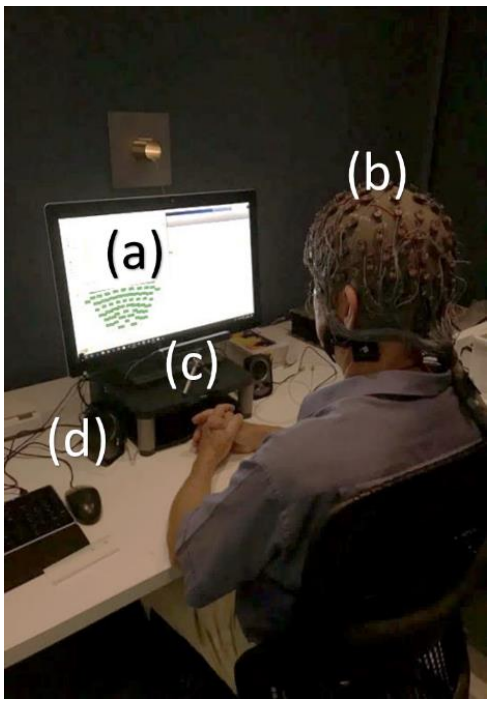


Figure 1: The equipment set-up within the faraday cage; (a) presentation screen currently showing the impedance of electrodes, (b) EEG cap with 128 active electrodes, (c) eye tracker, (d) presentation speakers

This allows for the data and output triggers and spectra to be monitored during the experiment, while having as little influence in the cage to affect the data. An active electrode cap by g.tec was used; g.GAMMAsys. There were 128 electrodes, with four EOG electrodes, a ground and a reference included. The data was recorded at a very high sampling rate of 9600Hz using a g.HIamp amplifier. This very high sampling rate was chosen in order to have enough data for connectivity. As discussed in part 5.3, connectivity needs a large amount of data to obtain results, as the more data there is the better the results will be. 9600Hz was the highest frequency possible to record at on the hardware so was chosen as the rate for recording. This way the time resolution of the connectivity can be made to have as high number of samples as possible to best observe the nature of the connections.

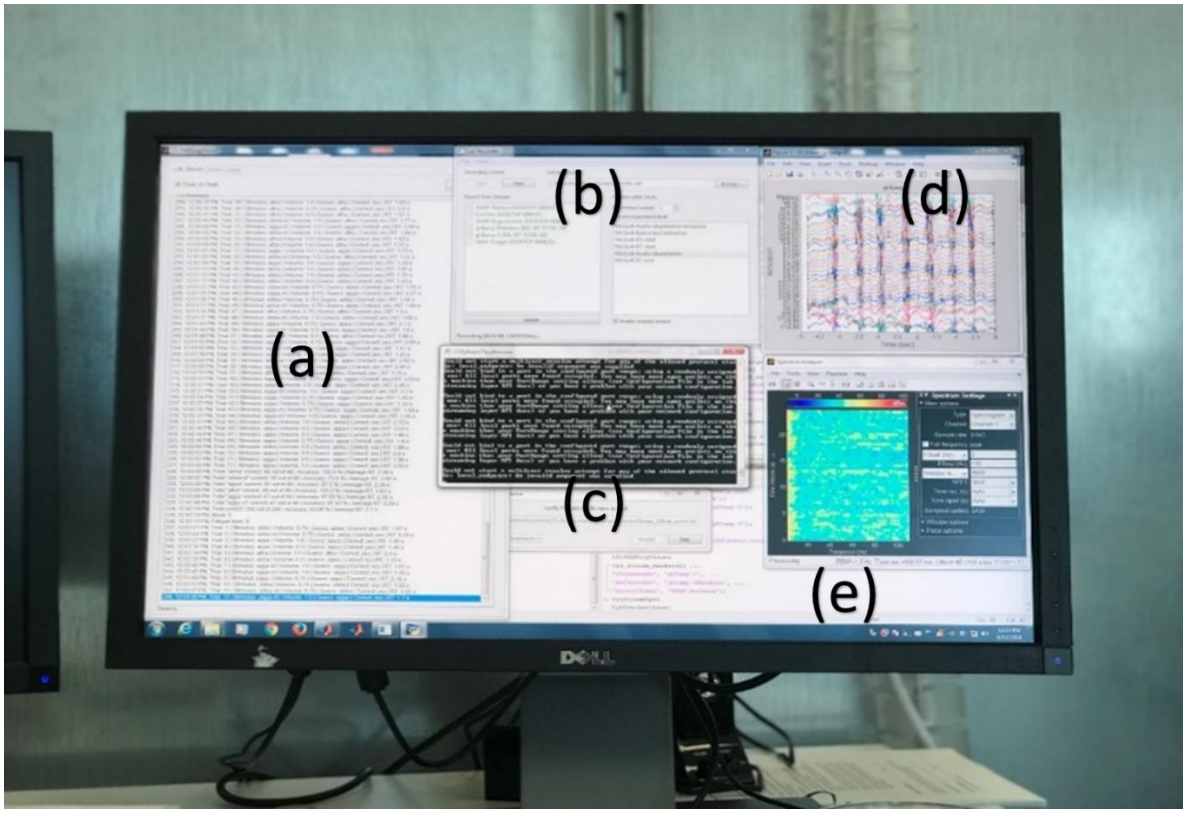


Figure 2: The different outgoing information from cage that can be monitored during the experiment; (a) output of markers from SNAP, (b) control interface for presentation to participants screen, (c) output of network connection to screen, (d) EEG data stream, (e) frequency of EEG data stream

The study received ethics approval from the Social and Behavioural Research Ethics Committee (SBREC), Flinders University (Project no. 6452). There was a consent form for the participant to fill to agree to the experiment and use of data. The participant was presented with a practice trial before the task started, to give them an opportunity to trial the experiment and ask questions before everything was set up. Then the entire head was rubbed down with alcohol, for removal of any residue, to lower the impedance between skin electrode interfaces. The placement of Cz was found using 50% of the distance from nasion to inion, and 50% between the preauricular of the ears and marked. The cap is then placed on head and Cz put in the appropriate place, and orientation of cap corrected. The EOG electrodes were placed on the face around 1 cm to the left of both eyes and 1 cm from the top and bottom of eye. The electrodes were then electronically located in 3D space using Polhemus hardware. This involved placing placeholders on the head at the temples and inion, then locating the ears and nasion with the 3D locating pen or orientation of the head. Once this was calibrated, each electrodes placement is input using the locating pen until all are in the system. This is to be used for more accurate spatial awareness of the electrodes for better connectivity analysis. To get good sources, the accurate location data is important since it uses back propagation from the electrode placement to predict the source in the head. Then the locators are taken off and the gel is put into the electrodes in the cap. The impedance is tested, and the impedance reduced until most of the impedance is below 5 kΩ. Then the eye tracker is set up and calibrated.

The participant is explained the tasks once more and instructed on how to conduct the experiment. The participant is then closed into the faraday cage to begin the experiment. Firstly, the eye tracker calibration is tested to correlate where they are looking in terms of eye tracker coordinates, to where on the screen they are looking. Next an eyes-open test is conducted for a minute. Then an eyes closed task for 5 minutes to record baseline data. Then the audio-visual degradation task is begun, and the 10 blocks presented. The subject is advised that should they need a rest they can take this in the breaks between blocks. After the completion of this there was another eyes-closed task for 5 minutes.

The timing of the stimulus was recorded with a marker to show when the audio started and when the syllable of interest was presented. The presentation of the start of selected stimulus can be seen as the spikes in red in Figure 3, and the green line the presentation of the start of the articulation of the target part of the stimulus. This target part is the start of the mouth movement of the 'BBA', 'THA', 'PPA' or 'GGA' section of the 'ABBA', 'ATHA', 'APPA' or 'AGGA'.

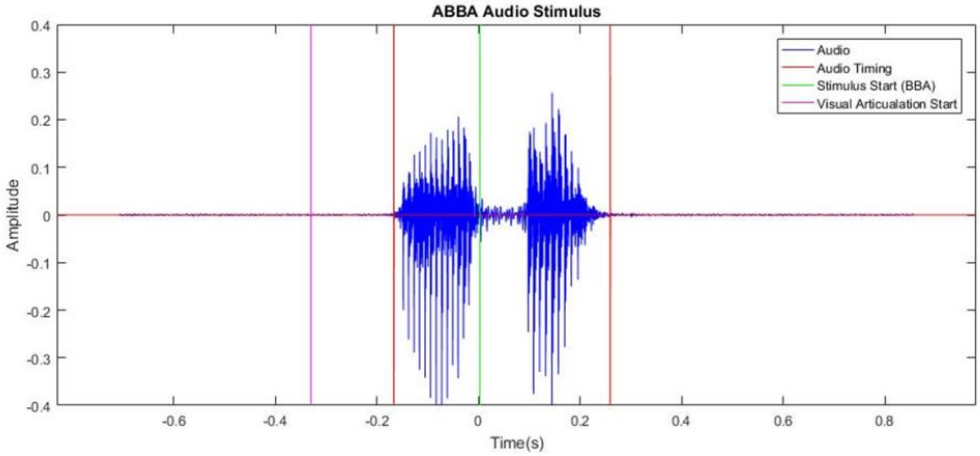


Figure 3: Presentation of audio of stimulus and timing

The audio marker was produced by using the left channel as the audio signal and the right channel as timing signal. This stereo signal was feed into custom built trigger box (Kleiss, Engineering Services) that split the signal by converting the audio signal into stereo output by bridging the left output channel to the right and the trigger channel was converted to a 1V, 0.5ms pulse and sent directly into the EEG amplifier digital input. The video marker was inserted by adding a white square to the trigger frame of the video and this square was identified by a light sensor attached to the screen (detecting the transition from black to white). The output from the light sensor is feed into the custom trigger box and converted to a 1V, 0.5ms pulse and sent directly into the EEG amplifier digital input. This allowed for precise timing of the stimulus presentation as software and hardware presentation on modern computers cannot always be guaranteed to deliver precise timing in a complex experiment setup. The visual articulation start marker was found using video images of the frames of each of the stimulus and selecting visually when the beginning of the articulation of the target sound was, as demonstrated in Figure 4. These video images in Figure 4 were also how the pink line representing the visual articulation start was found, by using the

video to find when the mouth first moved for visual stimulation. These markers will be later used in order to define the baseline region before any stimulation.

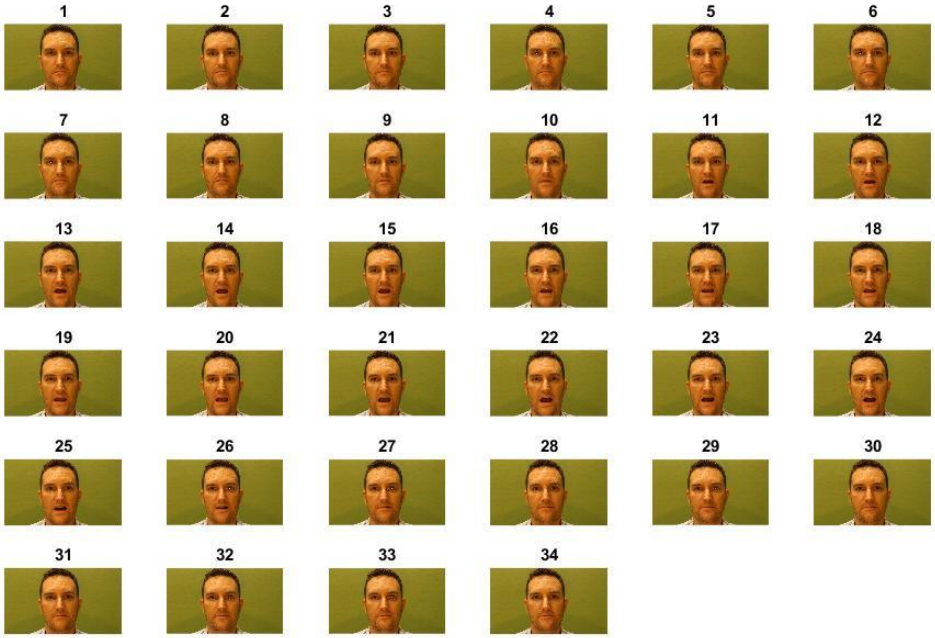


Figure 4: Frames of video for identification of the articulation of target syllable

There were 21 subjects tested on between the ages of 18 and 35, 9 female and 12 males. There was one subject whose data was not used in the analysis, due to high noise levels in the recording from a broken electrode in the active cap affecting the data. They were used for the statistics of the responses, but the EEG data was not included leaving 20 subjects for the analysis.

6.3 Data Processing

Once the data was obtained the data then needed to be cleaned and used for the connectivity between sources. The event markers were found, and the data established into the 10 different blocks of data. This removes the data in between which is not needed since the data at such a high sampling rate, it is very high in data size and so time-consuming to process. The blocks are large enough to be enough continuous data for cleaning of baselines in continuity. The analysis pipeline was as seen in Figure 5.

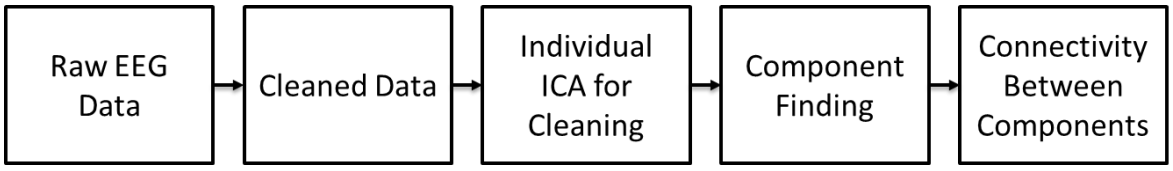


Figure 5: Analysis Pipeline

These steps will be shown in the following sections.

6.3.1 Behavioural Data Analysis

The responses were investigated to see any anomalies in responses and investigate the response times. The accuracy of the response at the different noise levels and stimulus was investigated. The response time can be investigated to see the difference between the speed to process and act upon the information. This was done using MATLAB to extract out the markers for each stimulus and noise level and then write these out and find the response time. To get the exact response time, there were markers input by the system when the screen showed the stimulus at the time of articulation. There was a response marker when the subject responded. These were used to get an exact timing of the response with no delay due to different stimulus length timings.

Statistics were then done on these results to determine the accuracy of each sound and sound level. This was done using MATLAB, by summating the target and actual response into a matrix, to create a confusion matrix out of the responses with a recall and precision measure. Precision is a measure of how many were correct out of the guesses that were the target. This is a useful measure to consider if subjects got the selection they should have. Recall is a measure of when they selected this choice, including the times that that answer was guessed that was not the right answer. So, this shows when they selected the sound for a different sound included. Together they give a better overview of the results by being able to investigate what was answered when they guessed incorrectly.

6.3.2 Cleaning of Data

The data was then converted into a format for a MATLAB based toolbox EEGLAB (Delorme & Makeig, 2004). The data was cleaned using a number of steps. The equipment was cutting out, due to a bad connection in some experiments, so the data was padded with zeros instead of NaNs when there was no input. This data will then be thrown away when it is epoched, rather than immediately, so that if half a block is still viable, it is used rather than throwing away a larger chunk of data than needed. The data was then converted into EEGLAB format to use the toolbox. The channels were added using the exact locations from the Polhemus data. The data was then tested for any bridged electrodes using the custom eBridge function in MATLAB to throw out any of the bridged electrodes which are giving the same information due to connection via conductive gel. The data was then de-trended using the function from EEGLAB to apply a linear, piecewise normalisation with segment length of 0.33 and step size of 0.0825. The EOG channels that were recorded were cleaned using the basic function of clean_rawdata from EEGLAB to remove artefacts from the EOG and convert it to an eeg3 format to be saved.

6.3.3 ICA for data cleaning

The data was then cleaned by using an ICA on each of the subjects. This will be done using a measure of independent component analysis (ICA). There are a number of algorithms, but the best choice would be an AMICA algorithm, as it was found to be reliable. The algorithm fits the subject's data into sources that this data is acting from. This uses the input parameters of the number of ICA models to be trained, the number of mixture components

to be assumed in the input data, the learning rate for the newton method and the initial learning rate gradient (Leutheuser, et al., 2013).

These components are then analysed by EEGLAB by inspecting the spectra, the activation in trials and the heat map representation on the head to automatically reject any sources that are not classified as brain activity. This can include muscle activity, outside influences or any other sources of noise. The sources are then fit with dipoles to check if any are situated outside the head to reject those as they are also not brain activity. The data is then projected back to sensor space to reconstruct the EEG without sources that were considered non-brain activity.

6.3.4 Source Analysis

Source modelling needs to be done on the individual participants reconstructed EEG data to be able to investigate the connectivity between these sources. This will be done using a measure of independent component analysis (ICA). There are a number of algorithms, but the choice used was the same as in the individual ICA; the AMICA algorithm. The process to fit sources for the connectivity analysis is outlined in Figure 6. To be able to get the connectivity as a mutual set of components across all the subjects, the sources need to be the same in all the subjects. This can be achieved by completing the independent components analysis across the subjects to get common sources. This is to obtain the same sources across subjects for comparison when performing the connectivity analysis as a group.

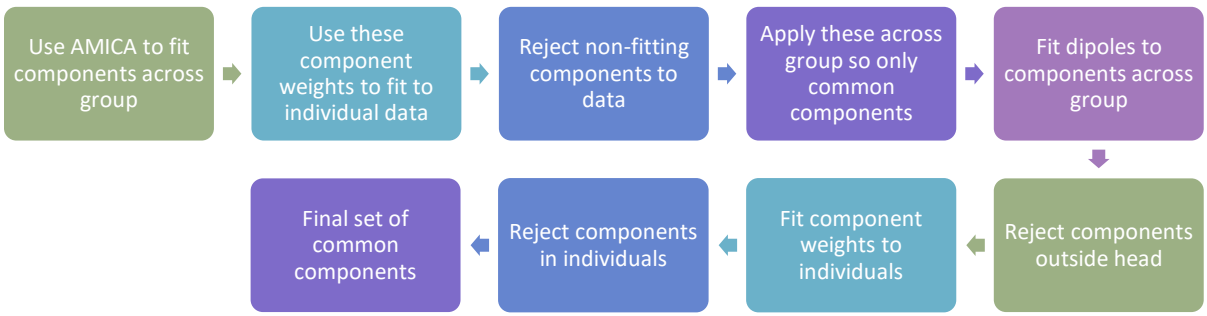


Figure 6: Outline of the group source analysis method

The components are fit using the valid data from all the subjects. Any sources that are classified by EEGLAB to be non-brain activity are rejected. These are classified using the spectra, trials and head map. These sources are then fit with dipoles for the location and orientation of the sources. These dipoles can be used in rejecting any sources that are outside the head as they are not brain sources. These component weights then applied to the individual subjects. These are analysed in the individual subjects and rejected for any individuals that they are not present in. These are then applied across the whole group to end up with a common set of components that are acting in all individual and are classified as brain activity.

6.3.5 Connectivity

To map the brain dynamics, there needs to be an analysis of the EEG data to process the connections in the brain. This was obtained using connectivity analysis. There are a number of different methods of measuring connectivity but an effective or functional measure of would be the best choice. Of these measures' coherent granger causality and transfer entropy were measures which are directed, commonly used and seem appropriate for this study. Some other measures that could be used if these fail are multivariate granger causality, directed transfer function or possibly mutual information; however, these are very time consuming to measure and would be left as a last resort.

Transfer entropy can be defined as computing the different joint and marginal probability distributions as seen in Equation 1 (Vincente, et al., 2011).

Equation 1: Transfer entropy; sum of four Shannon entropies (Vincente, et al., 2011)

$$TE(X \to Y) = S\left(\mathbf{y}_t^{d_y}, \mathbf{x}_t^{d_x}\right) - S\left(y_{t+u}, \mathbf{y}_t^{d_y}, \mathbf{x}_t^{d_x}\right) + S\left(y_{t+u}, \mathbf{y}_t^{d_y}\right) - S\left(\mathbf{y}_t^{d_y}\right).$$

Transfer entropy for two observed time series x_t and y_t can be written as Equation 2 where t a time-index value, u shows the prediction time, $y^{dy}{}_t$ and $x^{dx}{}_t$ are dimensional delay vectors (Vincente, et al., 2011).

Equation 2: Transfer entropy; two observed time series (Vincente, et al., 2011)

$$TE(X \to Y) = \sum_{y_{t+u}, \mathbf{y}_t^{d_y}, \mathbf{x}_t^{d_x}} p\left(y_{t+u}, \mathbf{y}_t^{d_y}, \mathbf{x}_t^{d_x}\right) \log \frac{p\left(y_{t+u} | \mathbf{y}_t^{d_y}, \mathbf{x}_t^{d_x}\right)}{p\left(y_{t+u} | \mathbf{y}_t^{d_y}\right)},$$

Therefore, the reconstructed state space of a pair of time series as a sum of four Shannon entropies can be calculated as Equation 3 (Vincente, et al., 2011).

Equation 3: Transfer entropy; two observed time series as a sum of four Shannon entropies (Vincente, et al., 2011)

$$TE(X \to Y) = S\left(\mathbf{y}_t^{d_y}, \mathbf{x}_t^{d_x}\right) - S\left(y_{t+u}, \mathbf{y}_t^{d_y}, \mathbf{x}_t^{d_x}\right) + S\left(y_{t+u}, \mathbf{y}_t^{d_y}\right) - S\left(\mathbf{y}_t^{d_y}\right).$$

Conditional granger causality is defined by a linear vector autoregression (VAR) model with two stationary time series that are explained by their own past using linear modelling (Franciotti & Falasca, 2018). This is then made as a lagged model to account for the difference in conduction of signals (Franciotti & Falasca, 2018). This results in the series in Equation 4, where ε_1 and ε_2 are the error estimators, m is the maximum number of lag observations, b_j and d_j are the gain factors, Y(t) is the influencing the signal and X(t) is the recipient signal (Franciotti & Falasca, 2018).

Equation 4: Conditional granger causality; lagged linear VAR model (Franciotti & Falasca, 2018)

$$X(t) = \sum_{j=1}^{m} a_j X(t-j) + \sum_{j=1}^{m} b_j Y(t-j) + \varepsilon_2(t)$$

$$Y(t) = \sum_{j=1}^{m} c_j Y(t-j) + \sum_{j=1}^{m} d_j X(t-j) + \eta_2(t)$$

As shown in some of the previous studies of EEG, mentioned in Section 5.2, there were some differences in activity at different frequency bands (Kumar, et al., 2016; Simon & Wallace, 2017). Kumar in 2016 found that there was an increase in coherence in the theta and gamma bands and a decrease in the alpha and beta bands (Kumar, et al., 2016). Therefore, analysis at different frequency bands, instead of across the spectra, might lead to different brain processing. The best bands to investigate would be delta (1-4Hz) theta (4-8Hz), alpha (8–14Hz), beta (14–30Hz), and gamma (30–45Hz) (Kumar, et al., 2016).

6.3.5.1 Evaluating Connectivity Measures

Before using the measures, an analysis was done on them to prove their validity. This was done by creating some artificial brain signals in MATALB that are simple sinusoids, as seen in Figure 7. These were made to have a high sampling rate equal to that of the actual data; 9600Hz. The lengths of the epochs were 2.5 seconds, since this is the length of most of the epochs that will be used in the connectivity. Channel 1 and 2 were made to be the same frequency, 6.1Hz, but with a 0.1 second offset for channel 1. These two signals are the two that are in phase and therefore should show connectivity between them with channel 1 as the leading channel. Channel 3 was made as an out of phase channel at 15.793Hz; this should not be a resonant frequency. Therefore channel 3 should be classified as no connectivity to 1 or 2.

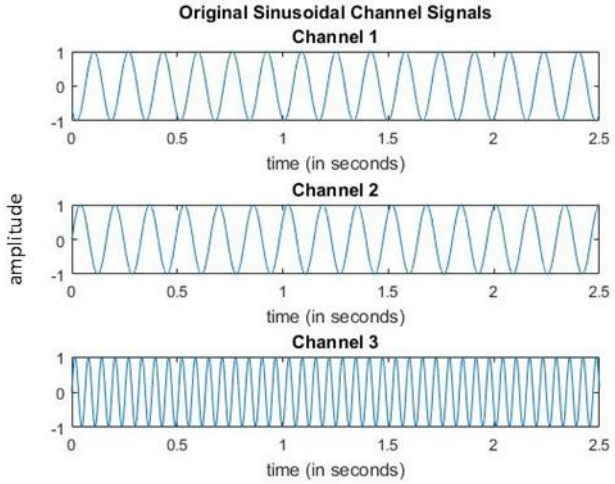


Figure 7: Artificial signals for testing; raw signals of 6.1Hz offset 0.1s, 6.1Hz and 15.793Hz respectively

The channels do not resemble EEG data as EEG data is very noisy. So, two conditions of noise were made; low noise and high noise. This is to be able to test the effect of noise on the connectivity measure. The low noise condition was made to have a signal to noise ratio

of 10 with just white noise added to the signal. This can be seen in Figure 8. Then a high noise condition was created with a signal to noise ratio of 0.1. This can be seen in Figure 9.

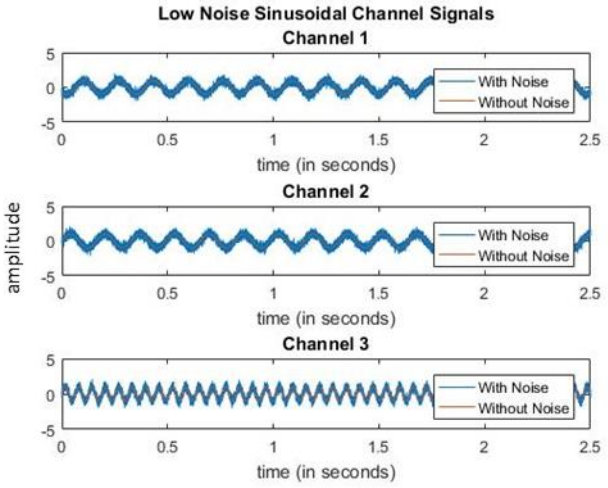


Figure 8: Artificial signals for testing; low noise added to a ratio of 10

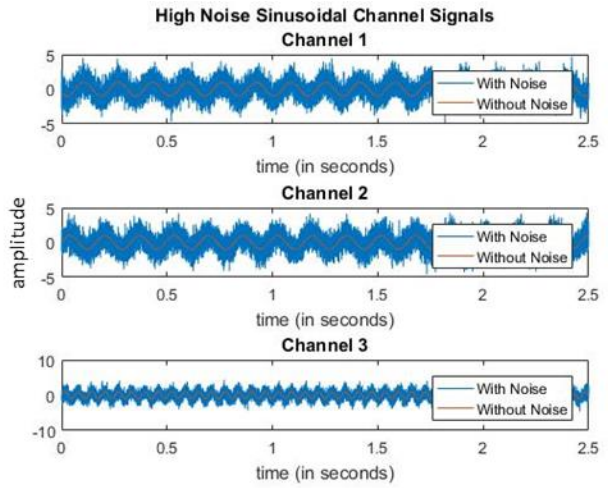


Figure 9: Artificial signals for testing; high noise added to a ratio of 0.1

These were then used in the connectivity measures of transfer entropy and conditional granger causality. There was also a frequency banded conditional granger causality used in order to test if this is valid. The output was an adjacnenvy matrix showing the connections of the signals.

6.3.5.2 Analysis Methods

The analysis is in the temporal domain, so the timing of the events is important. The zero point of the epochs was defined to be when the target sound of the stimulus was presented to the subject, green line in Figure 10. This was the 'ba', 'ga' or 'pa' sound. A baseline region needs to be defined to have an accurate representation of resting state of the brain before a stimulus is presented. If the resting state is not accurately defined, then the connections present during this resting state may be found instead of the audio-visual connections. If the time right before the zero point was used, then this would be within the first section of

the audio, the ‘a’ sound. Before this would still have an initial mouth movement that would be interpreted by the subject as a stimulus. Therefore, as seen in section 6.2, the region of the video before any mouth movement was found as represented by the pink line in Figure 10. Therefore, the baseline region is before this at -0.6 to –0.4 seconds before the stimulus presentation. This is shown as the blue region in Figure 10. The region of interest is defined as after the stimulus until after the decision has been made which from 0 to 0.8 seconds. This is the purple region in Figure 10.

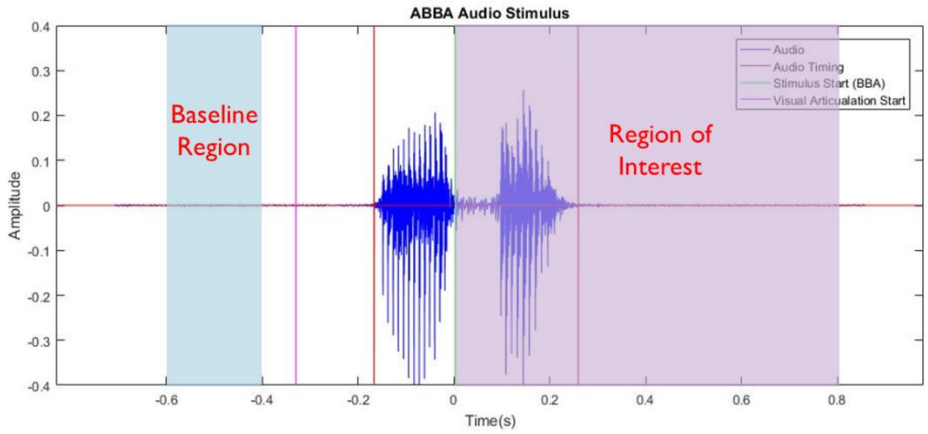


Figure 10: Audio of stimulus to define the regions for analysis

As a temporal analysis, the connectivity is measured in small time windows across this region of interest. 200ms windows were used in order to have enough data for temporal analysis. The more data that is present the more accurate the connectivity results. This means there is 200ms of data at 9600Hz, which is 1920 samples for connectivity per window. This is the best compromise between enough data for an effective connectivity analysis and temporal resolution. By slightly overlapping the windows, there can be a very small change recorded over time with the better temporal resolution. This is done across the entire region of interest as seen as an example of the first three windows of connectivity in Figure 11. See Appendix B – Code, Section 12.1 for the script that was used in colossus for the connectivity analysis.

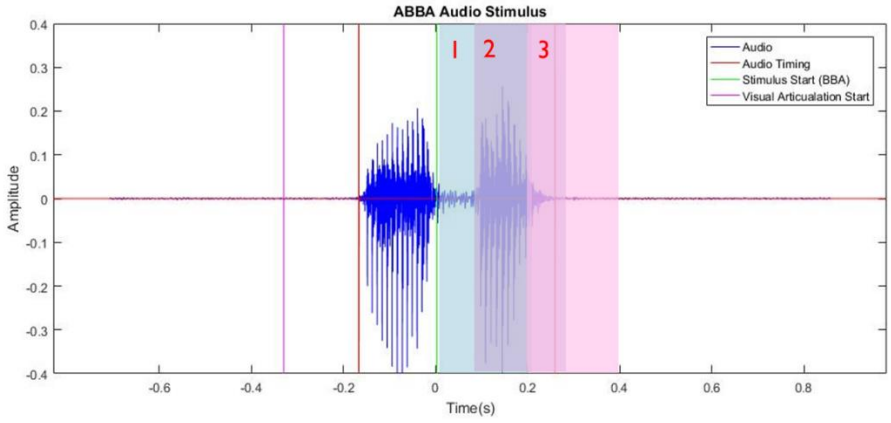


Figure 11: Audio of stimulus to show the method of temporal connectivity analysis

6.3.5.3 Connectivity stats

Once the components are found, then the validity of these connections needs to be statistically significant. This was done using cluster-based statistics or multivariate analysis using inferential statistics. The choice of method would be network based statistics using a method to control false discovery rate. It was calculated in MATLAB using the Brain Connectivity Toolbox. This has a network-based statistics package for the testing of hypotheses about the human connectome (Zalesky, et al., 2010).

The data for each stimulus type and comparison was all collated for comparison by looking at each stimulus type, noise level and correct or incorrect answer were analysed. This breakdown of each of the types is in Figure 12. This put all of the combinations of each of the subject's data into categories and analysed whether the changed between the reaction and the baseline were significant or not.

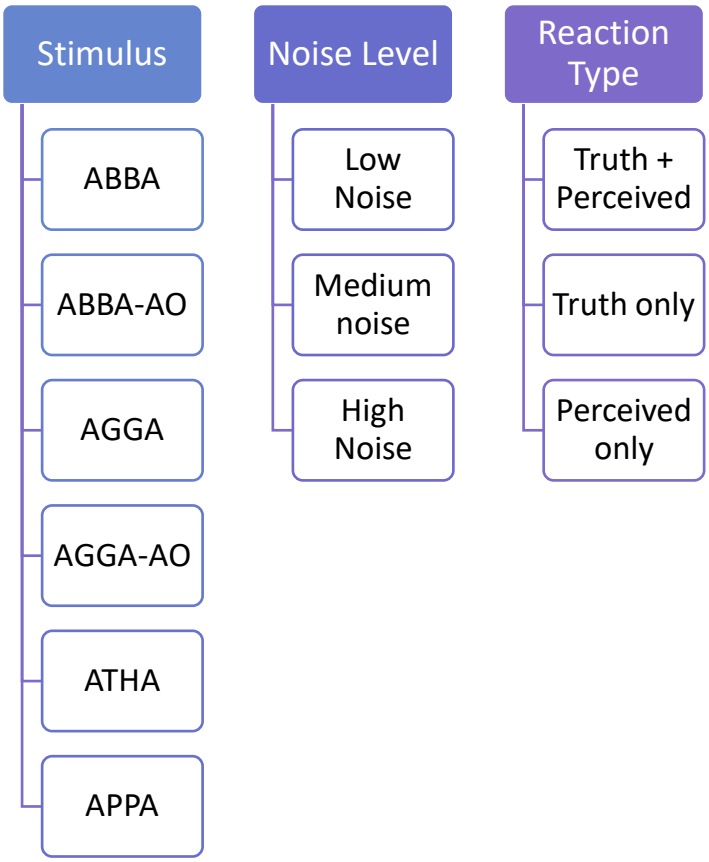


Figure 12: Breakdown of the compartmentalisation of data

The network-based statistics threshold was 0.3 and the alpha value 0.05. It was run as a t test not an f test, and for false discovery rate method. This then took in all the subjects' epochs for each type and compared it to the region baselines to test for statistical significance, taking a eyes-open no task base off of both of these to eliminate this base network in the functioning. It resulted in a 6 by 3 by 3 matrices of the statistical results. This included p values and matrices of significant connections. The results noise level and reaction type were also taken since this will also be investigated, so then this was just taken

across the stimulus. See Appendix B – Code, Section 12.3 for the script for calculating the connectivity statistics.

6.3.5.4 Connectivity Visualisation

Once the connections are checked to be significant or not, then the adjacency matrix of the connection can be determined to be valid. This adjacency matrix is obviously quite hard to visualise, so tools were used in MATLAB plotting to be able to visualise and investigate this mapped onto the head, as seen in Figure 13.

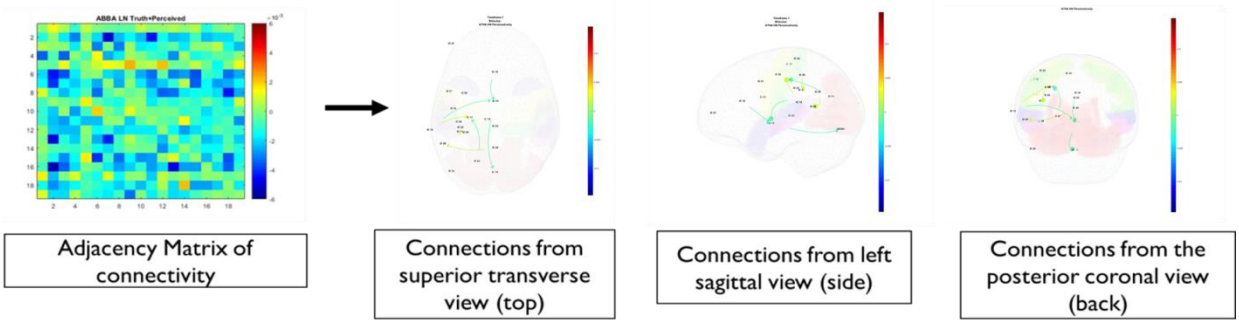


Figure 13: Visualisation method of connectivity

This was done using a number of steps. Firstly, the head mesh was created using Fieldtrip toolbox (Oostenveld, et al., 2011). This plots a mesh from an example MRI brain. Then the dipoles were plotted on the head by using their coordinates in the brain. Then the connectivity matrix is used to plot arrows between the chosen connections using base MATLAB functions. Then the Brainnetome atlas was used to map the brain regions that were found from MRI scans of the brain (Fan, et al., 2016). Appendix B – Code, Section 12.2 for the script for plotting the results of conditional granger causality results.

7 RESULTS

7.1 RESPONSE STATISTICS

The percentage of each response type being correct or incorrect can be seen in Figure 14.

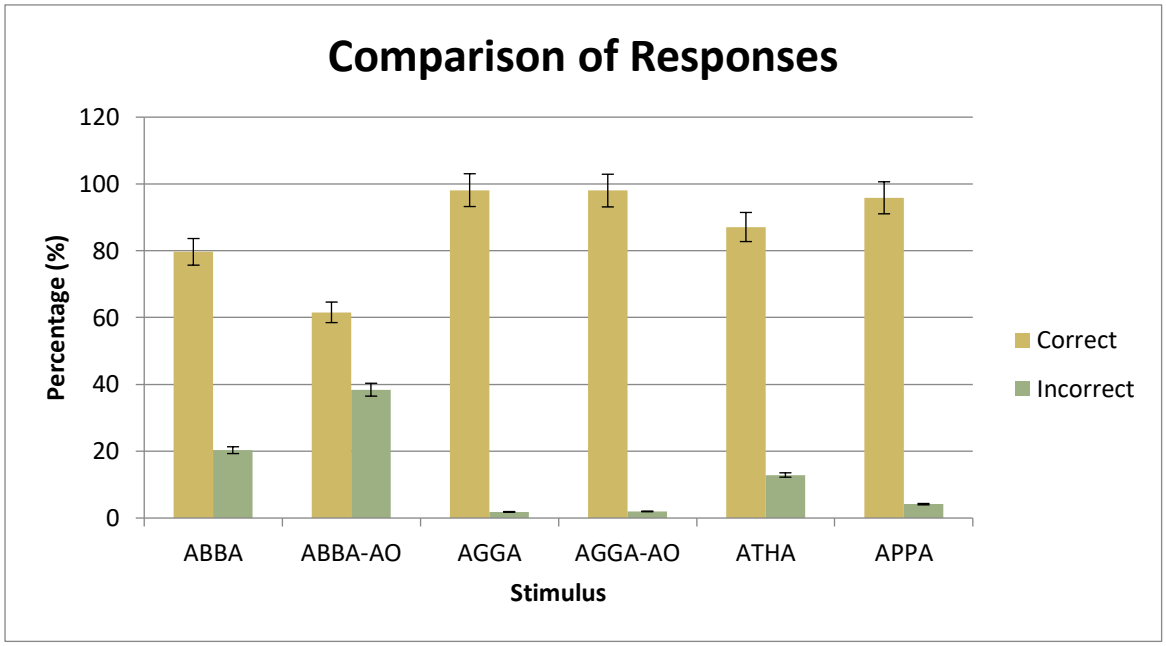


Figure 14: Comparison of correct or incorrect responses across stimuli with 5% error

There was very high accuracy on most of the results. AGGA and APPA were almost always correct, the ATHA around 90% correct. ABBA audio only presentation where there was no visual stimulus had the highest amount incorrect. This can be broken down further to the different noise levels to investigate the effect of the differing noise levels on the response, as seen in Figure 15.

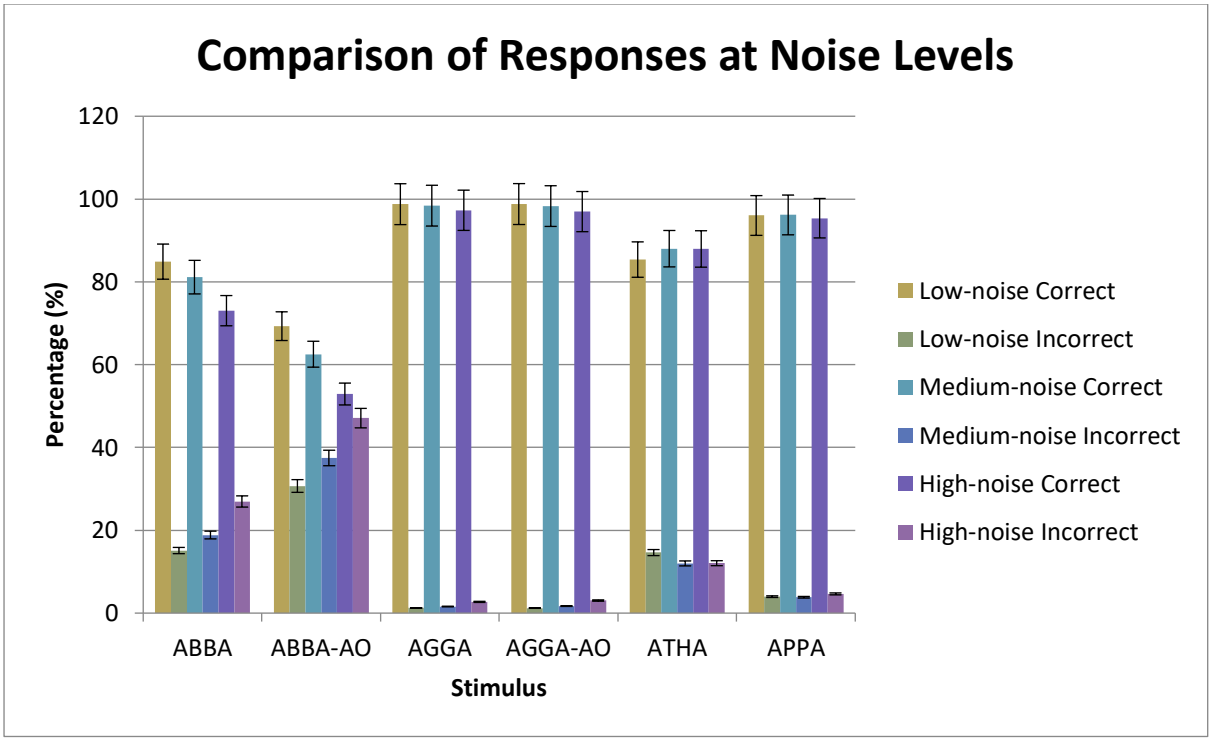


Figure 15: Comparison of responses at different noise levels with 5% error

It can be seen from Figure 15, there is a slight decline in accuracy across noise levels for AGGA, AGGA-AO and APPA as the noise increases. This is to be expected as it is a slightly more difficult task in the higher noise. However, the ATHA accuracy improved as the noise increased and there was a very large decrease in the accuracy of the ABBA and ABBA-AO as the noise increased. This can be further investigated by using a confusion matrix.

The confusion matrix with the precision and accuracy for the low noise results can be seen in Figure 16. The answers were all generally precise, except for the few exceptions. Not all subjects perceived the ATHA; two did not so that is only at 70%. This is higher than expected as compared to many studies, showing a relatively good ATHA stimulus since the values can range from 26% to 98% (Nath & Beauchamp, 2012; McGurk & MacDonald, 1976; Gentilucci & Cattaneo, 2005). There was also confusion with ATHA where ABBA audio only was identified as ATHA due to preconditioning.

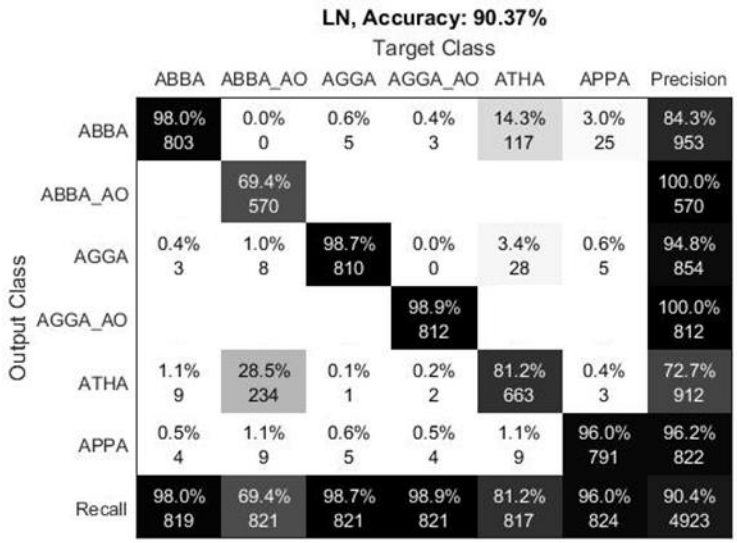


Figure 16: Confusion matrix of responses with precision and recall for low noise results

The medium noise and high noise confusion matrices are seen in Figure 17 and Figure 18 respectively. They can be compared to Figure 16 and each other to see the effect of an increase in noise. As can be seen this misconception of ABBA audio only for ATHA increased as the noise increased. So as the subjects could not hear the sound as clearly, they were more likely to identify ABBA as ATHA without the aid of visual stimulation. There was also an increase in incorrect answers overall, particularly in ATHA, since this is audio driven.

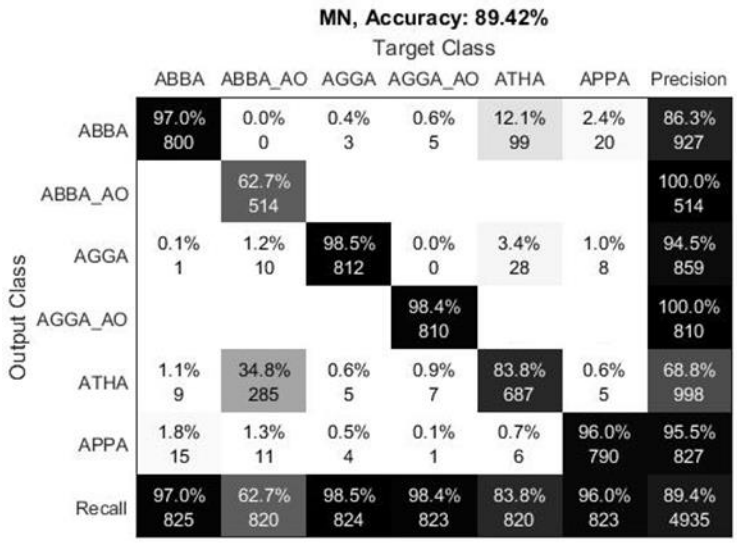


Figure 17: Confusion matrix of responses with precision and recall for medium noise results

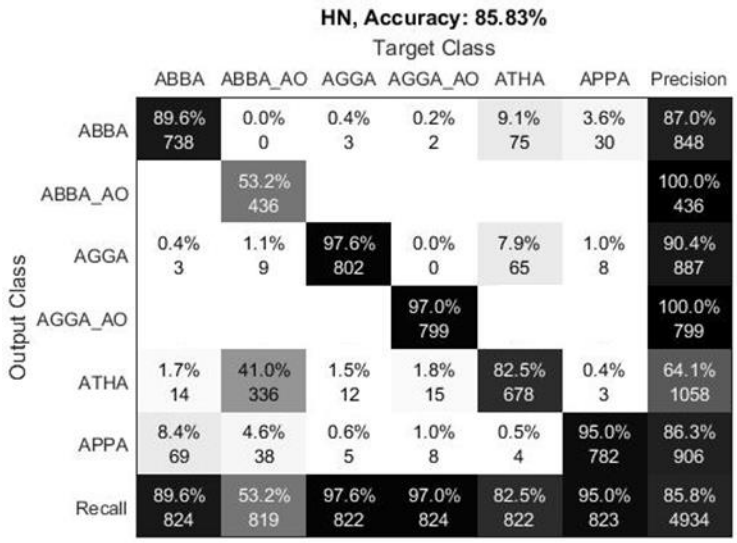


Figure 18: Confusion matrix of responses with precision and recall for high noise results

The response times of the data were extracted from the data to be able to investigate the effect of stimulus and noise level on the timing of the response, Figure 19.

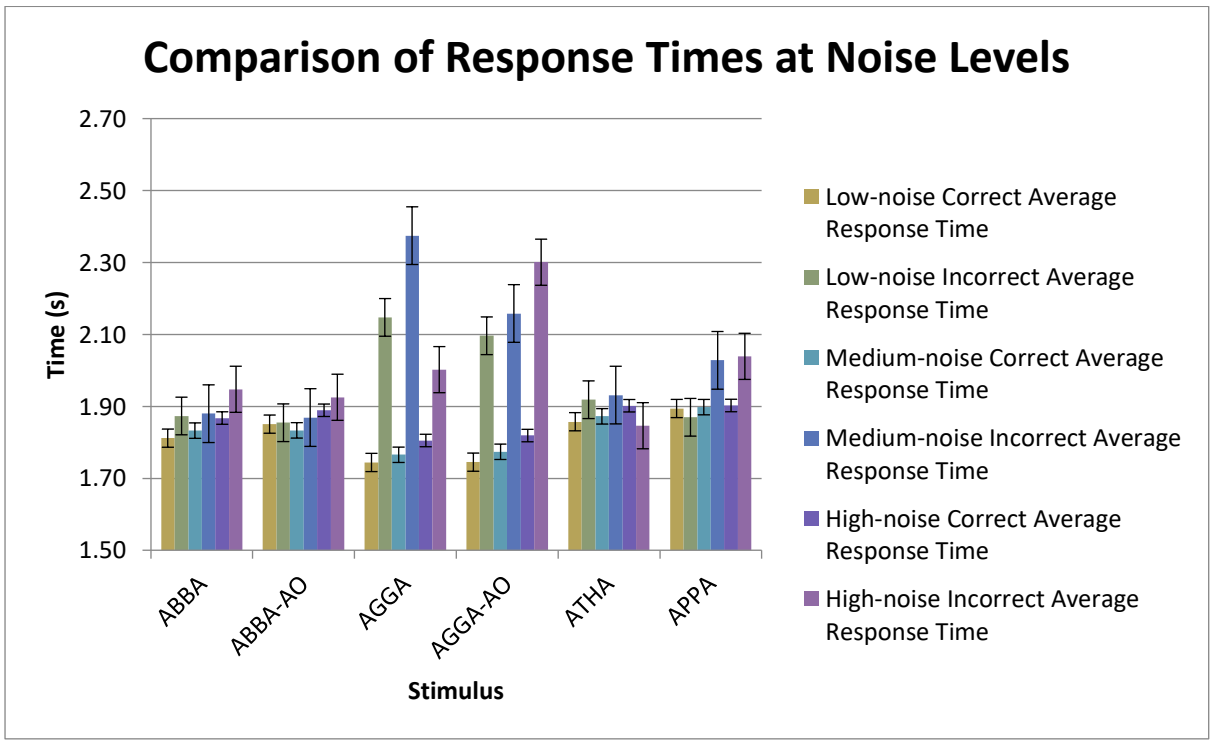


Figure 19: Comparison of the response times at different noise levels with standard error

From the response times of the data, the response time of incorrectly identified AGGA and AGGA-AO (AGGA with no visual), were quite long. This could be correlated to the fact that this is one of the easier stimuli to determine so the subjects may have not heard the stimulus and taken a bit longer and taken a guess at which one it was resulting in a longer time, but a very quick response time when it is properly recognised. Most of the other

responses were relatively insignificant though there was a slight increase in timing as the noise increase as it was more difficult a task.

7.2 Cleaning of Data

There were 21 subjects, however the actual data of one subject was removed, since there was a bad fitting of the cap and lots of the electrodes were not against the head, along with a failure of recording equipment, resulting in not enough data to include the subject. However, an extra subject was recorded to achieve the goal of 20 subjects. An example of the original data was as seen in Figure 20. This data has a lot of baseline drift and is extremely noisy.

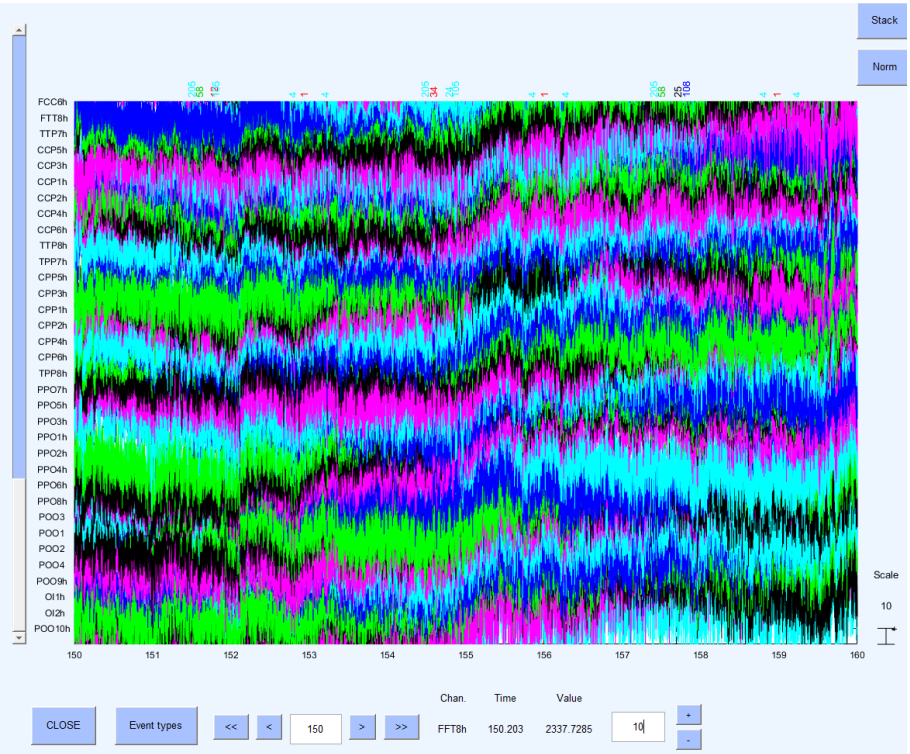


Figure 20: Raw data of subject 1

As can be seen in Figure 21, this is much cleaner data in comparison to the raw data after the cleaning methods have been applied. It has a straight baseline with much clearer signals.

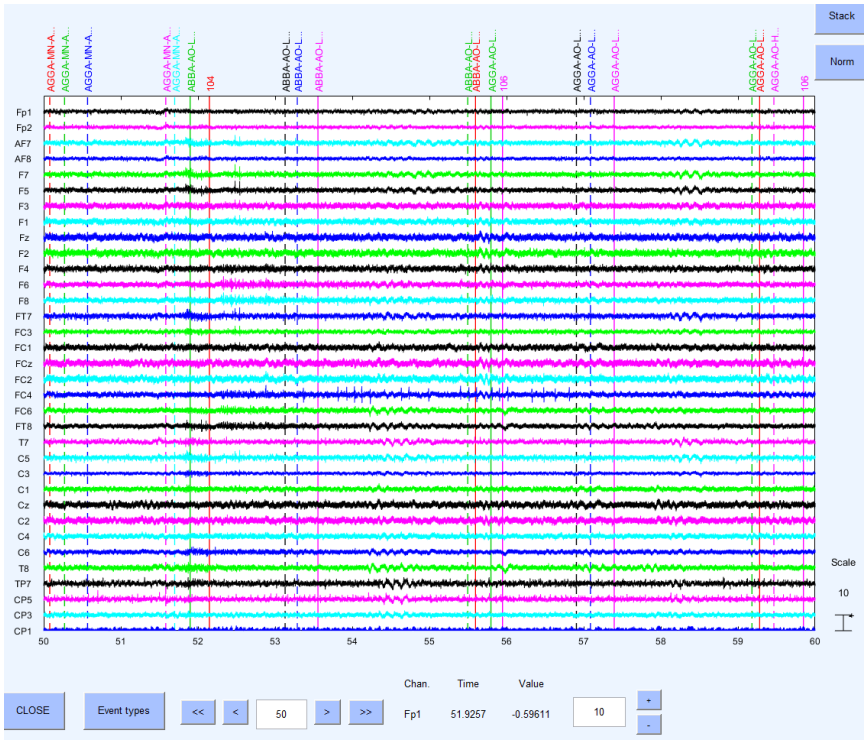


Figure 21: Cleaned data for subject 1

Each of the subject's data was fitted with components to isolate for muscle or other non-brain activity to be able to further clean the data. As can be seen in Figure 22, the good component has a good frequency spectrum and activation pattern, if they do not they are rejected. This then results in the more refined EEG activity data in Figure 23.

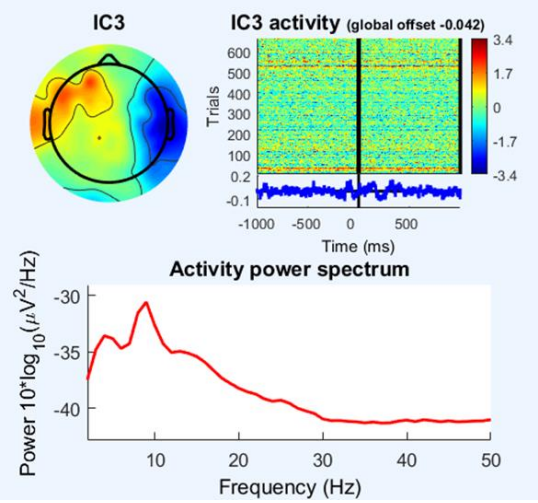


Figure 22: Good components kept, subject 1 example

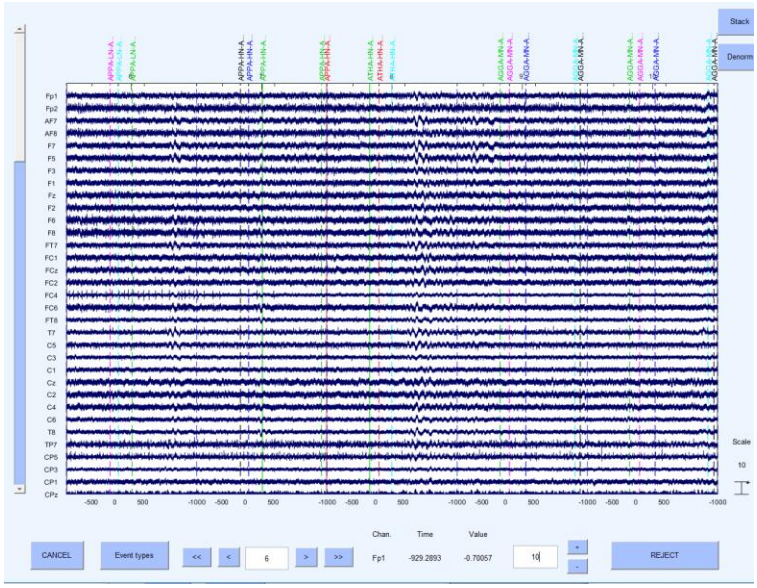


Figure 23: Data further cleaned using ICA

7.3 Source Analysis

The results for the final set of components that are common for all subjects can be seen in mapped in Figure 25. The ERPs of these components can be seen in Figure 24. They have good ERP shape to assess they are valid. They have the usual shape of when a stimulus presented with the P300 present. There were 27 components left. They are fit in one side of the head since brain processes are often unilateral and the region of interest was the left side of the brain.

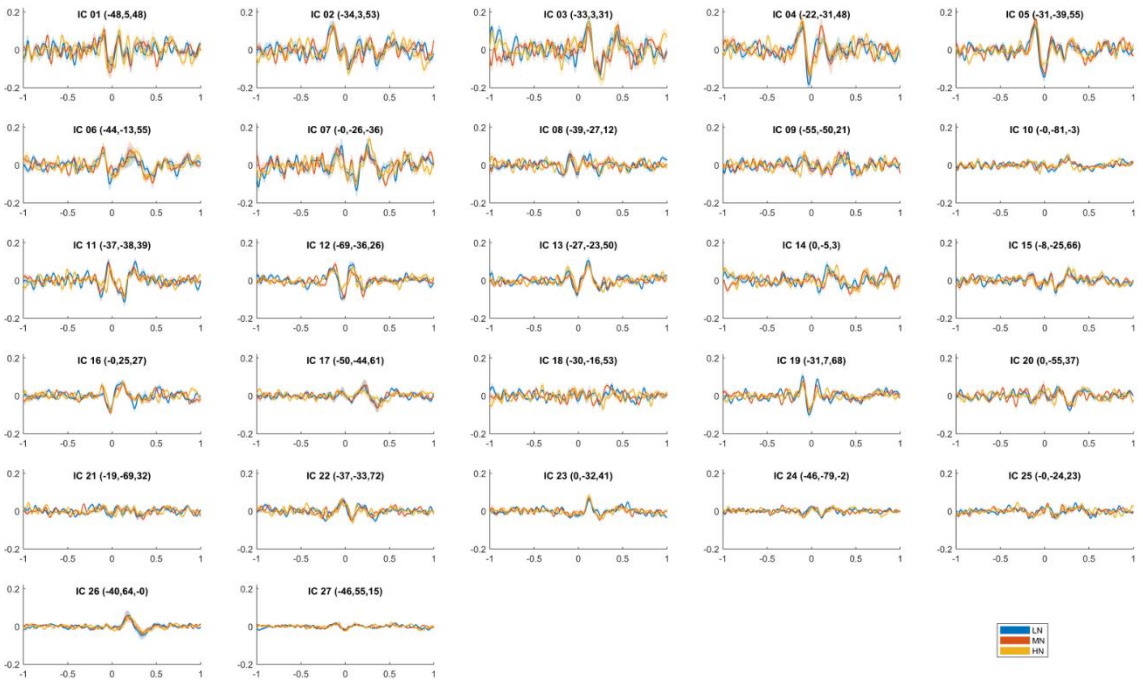


Figure 24: ERP components for the final sources

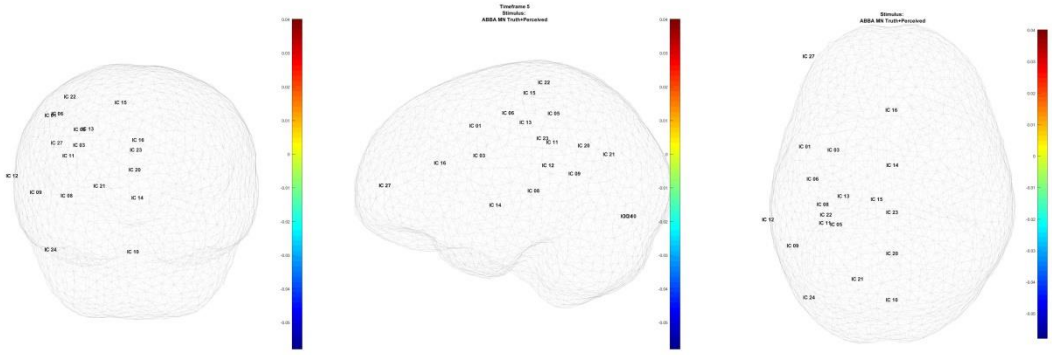


Figure 25: The dipole locations mapped of final components

The components can be analysed to see whether they are relevant. As seen in Figure 26 the components are in the regions of interest of the areas which were identified to be working in the MRI studies.

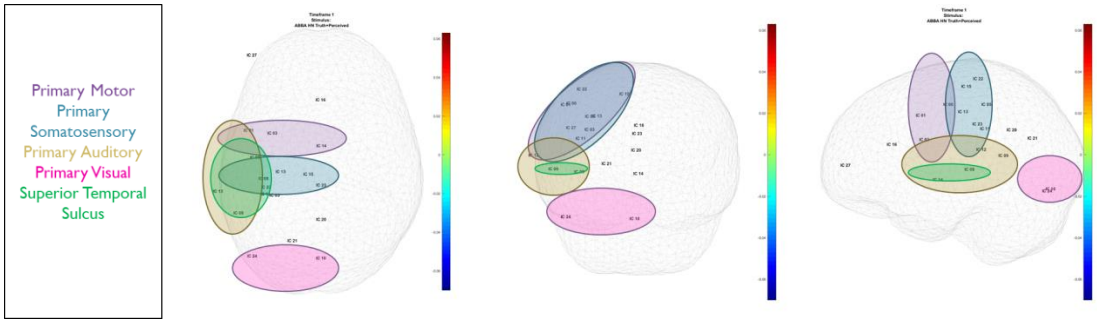


Figure 26: Components mapped with relevant areas of interest

The components were classified as brain regions to be able to determine the activity that is occurring in these components, Table 1.

Table 1: List of components with corresponding relevant brain regions

Brain Region	Component
Primary Visual Cortex	24
	21
	10
Primary Auditory Cortex	12
	8
Superior Temporal Sulcus	14
	9
Primary Somatosensory Cortex	6
	22
	5
	13
	15
Primary Motor Cortex	1
	3
	11

7.4 CONNECTIVITY

7.4.1 Testing Measures

The ideal directed and ideal undirected adjacency matrix shows the expected result of a perfect solution, Figure 27.

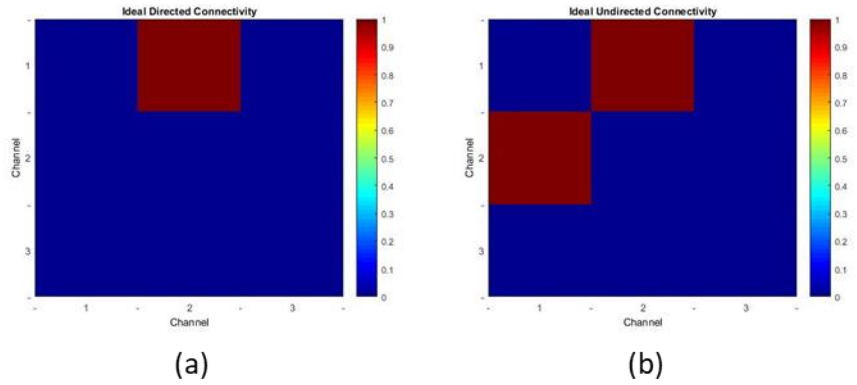


Figure 27: (a) Ideal directed adjacency matrix, (b) Ideal undirected adjacency matrix

As can be seen in Figure 28 (a), transfer entropy is not a very reliable measure and is relatively undirected in the results. Granger causality is a good measure of connectivity and is directed, Figure 28 (b). The frequency banded results were a good measure but less able to determine direction of the connection, Figure 28 (c).

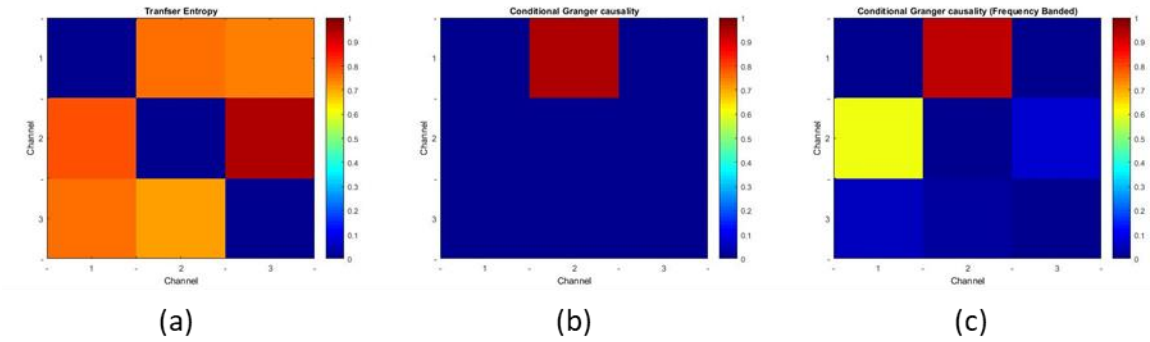


Figure 28: (a) Transfer entropy result at low noise artificial signal, (b) Condition granger causality result at low noise artificial signal, (c) Condition granger causality frequency banded result at low noise artificial signal

As seen in Figure 29 (a), as the noise increased transfer entropy did improve slightly to get better results, as compared to Figure 28. One of the reasons it is used in this study is that it is good for noisy data. However, granger causality was still a better measure even at higher noise, Figure 29 (b). The frequency banded results, Figure 29 (c), got worse in directionality with the increased noise but were still a relatively good measure.

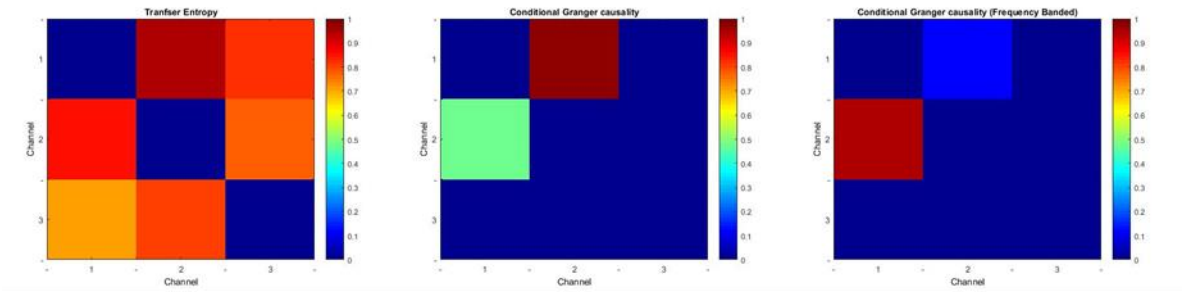


Figure 29: (a) Transfer entropy result at high noise artificial signal, (b) Condition granger causality result at high noise artificial signal, (c) Condition granger causality frequency banded result at high noise artificial signal

7.4.2 Transfer Entropy

Transfer entropy was predicted to not be the best measure of connectivity in the trial of the connectivity measures. However, the advantage of transfer entropy algorithm used was the speed of calculation of the measure. So, a small investigation was done using transfer entropy while the calculation of the other results was processing.

7.4.2.1 Connectivity Statistics

As expected from the small trial of artificial data, the results were all not statistically significant. However, the results will still be examined for any patterns even if they are not significant changes in connectivity.

7.4.2.2 Results

To visualise the connectivity the parts of the brain were mapped. The small investigation was on the effect of noise level on the connectivity. There are arrow heads on the connections which indicate the direction of the connection. The strength of connection is shown by the colour of arrow; zero is the centre of colour bar so warm colours are increased connection and cool colours are a decrease.

From Figure 30, it can be seen that at low noise it can be seen that in comparison to the baseline there was no STS activation and a decrease in activation between the auditory and visual cortex.

Figure 31 shows that at medium noise there was an increase between the STS and the visual cortex. There was a decrease between the somatosensory and visual cortex. There was also a decrease between auditory and visual cortex.

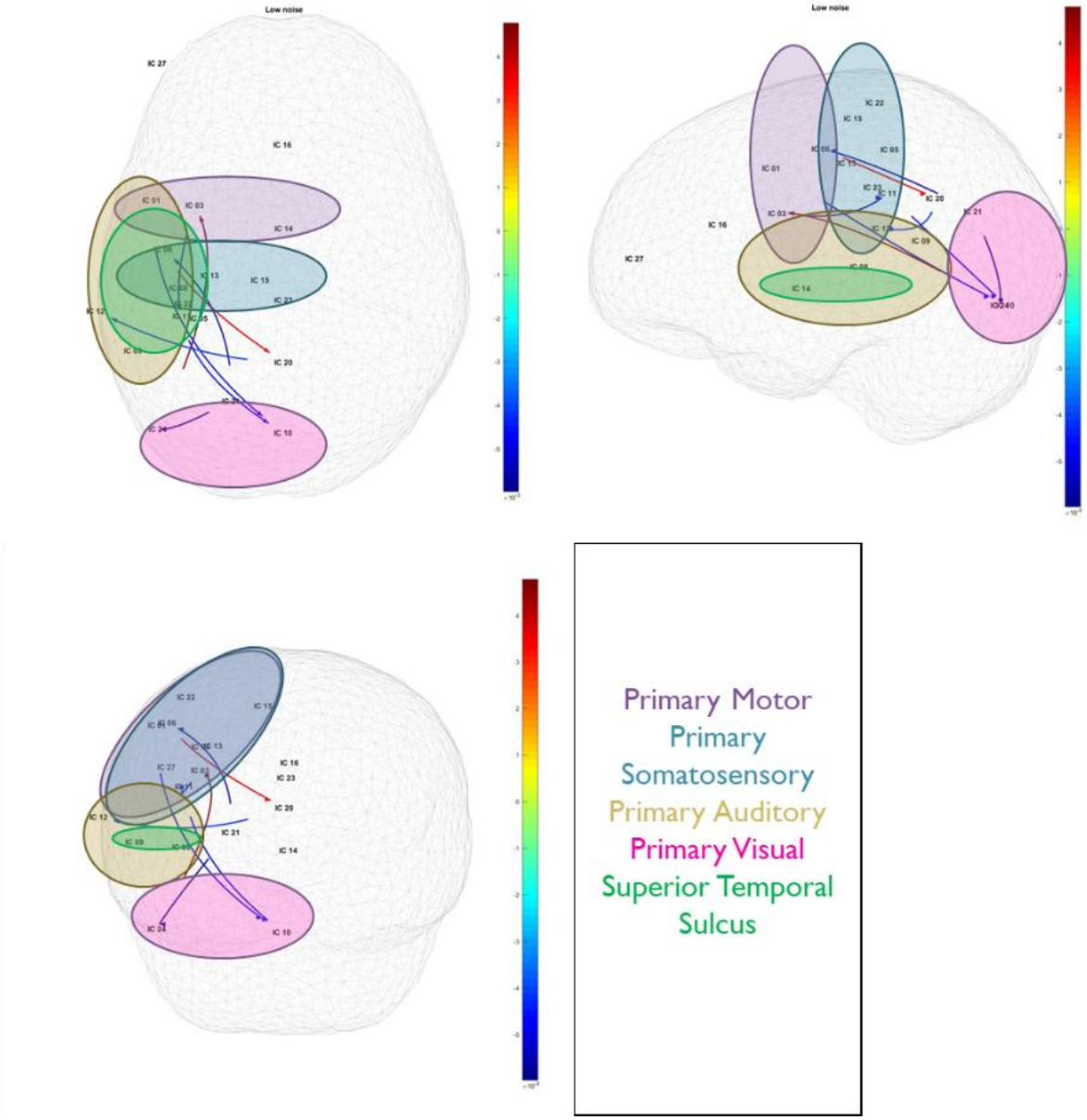


Figure 30: Low noise transfer entropy connectivity results

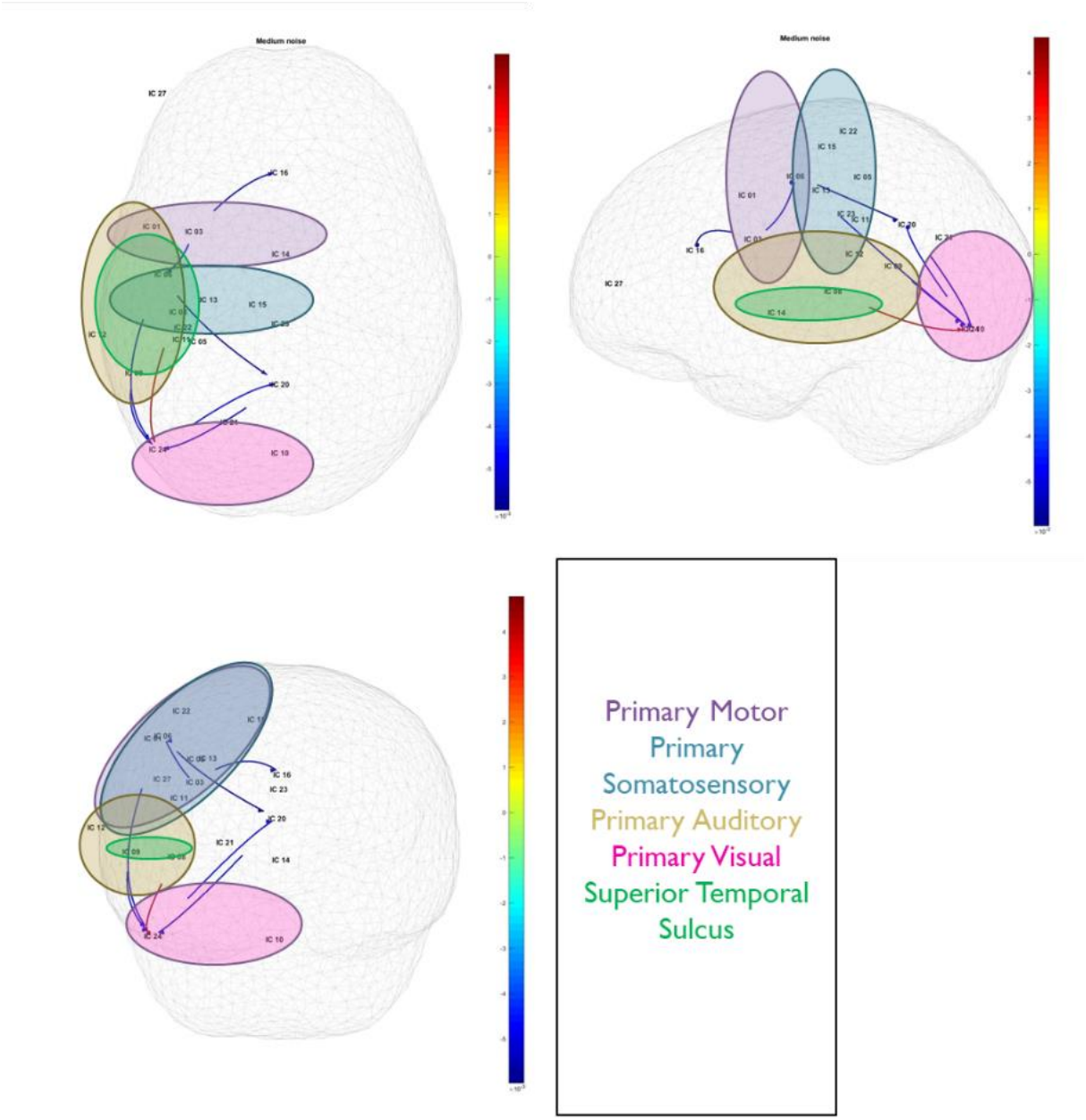


Figure 31: Medium noise transfer entropy connectivity results

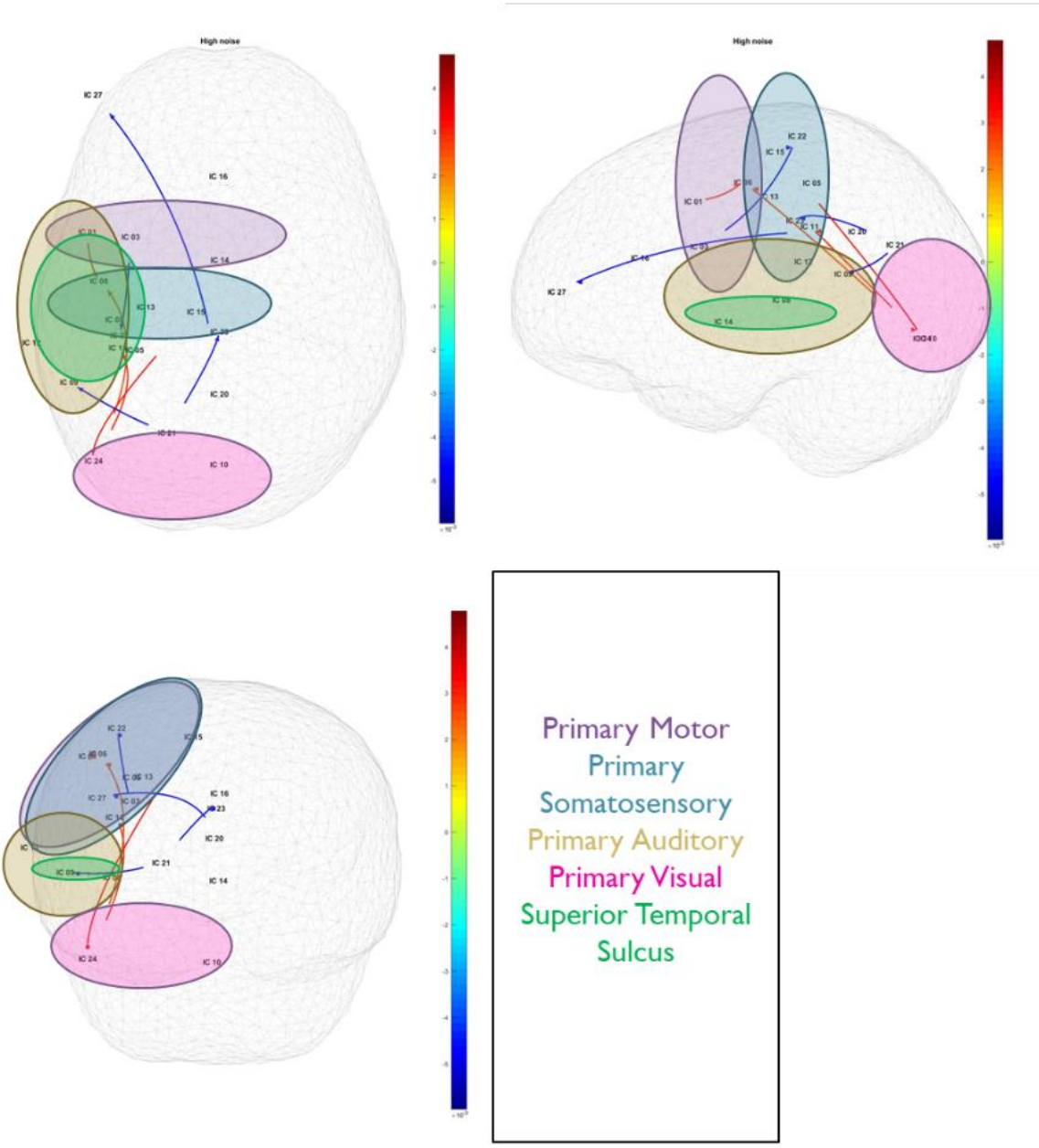


Figure 32: High noise transfer entropy connectivity results

Figure 32 shows that at high noise there was an increase in activation between sensory to visual and auditory cortex. There was also an increase in connection between visual and auditory cortex, but no apparent STS connection increase or decrease which is unexpected. This is considered the 'hard task' which could explain the connectivity to the frontal cortex as it is recruited in the decision making. Transfer entropy did not have the best results as expected. The comparison of time intervals of correct vs incorrect was chaotic and did not appear to show a pattern.

7.4.3 Condition Granger Causality

A better method of mapping the regions was found to be able to automatically have an anatomical region using the Brainnetome brain atlas for the region colours as seen in Figure 33.

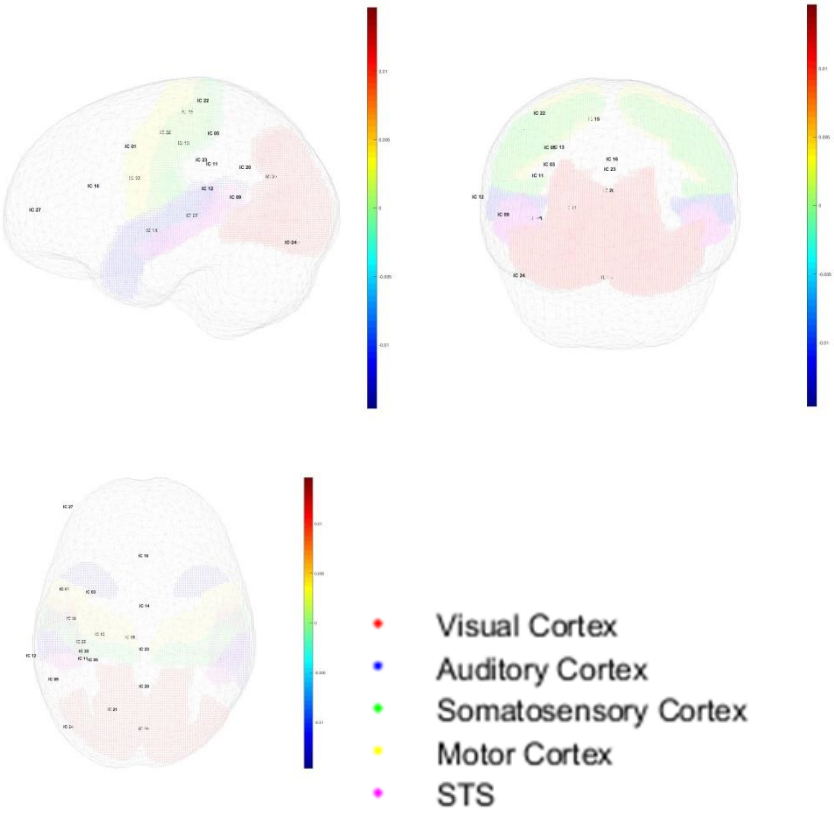


Figure 33: Brain regions mapped using the Brainnetome atlas

One downside of this method of imaging is that although the dots are made transparent to see the connections through, the dots themselves are not that transparent in respect to one another and as such the regions that are behind another do not show through well.

In the visualisation of the results there are arrow heads on the connections which indicate the direction of the connection. The strength of connection is shown by the colour of arrow; zero is the centre of colour bar so warm colours are increased connection and cool colours are a decrease.

7.4.3.1 Connectivity Statistics

The results had some statistically significant outcomes. The comparison of the truth and perceived results were all significant as seen in

Table 2. However, from

Table 2, not all the truth only or perceived only were statically significant. This is not particularly surprising since this is when subjects got incorrectly this perceived response would include when they were not paying attention and other such times. From Table 3 across stimulus but at different noise levels, all the noise levels were significant. Table 4 shows that the correct vs incorrect stimulus comparison was not statistically significant.

Results that were not statistically significant will still be investigated however, as they are still interesting results.

Table 2: p values showing statistical significance of results across the stimulus and perception (purple significant)

	Truth + Perceived	Truth Only	Perceived Only
ABBA	0.0058	0.1232	0.2846
ABBA-AO	0.0484	0.0020	0.0164
ATHA	0.0110	0.6948	0.4308
APPA	0.0094	0.2824	0.4460
AGGA	0.0280	0.0124	0.0110
AGGA-AO	0.0024	0.0152	0.0026

Table 3: Noise level effect on p values (purple statistically significant)

	Low Noise	Medium Noise	High Noise
All Stimulus	0.0466	0.0818	0.0152

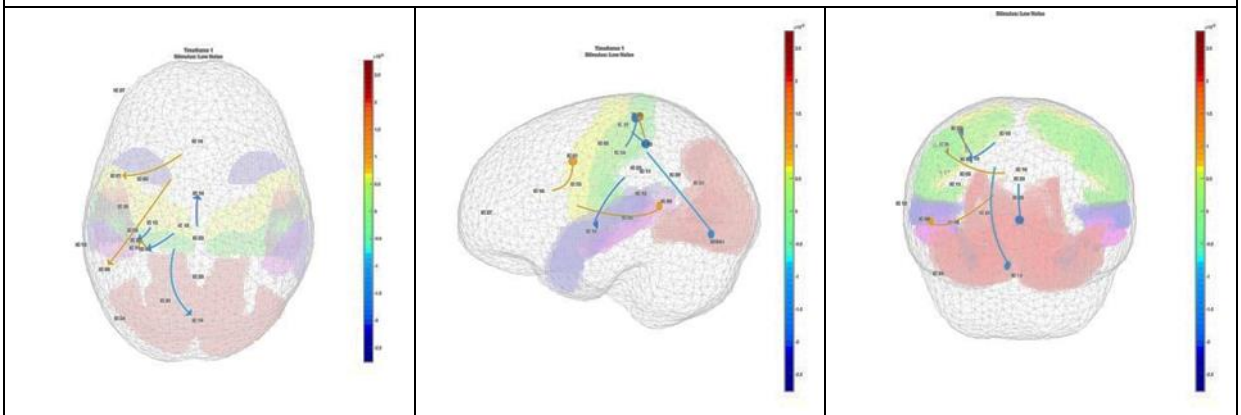
Table 4: Correct vs incorrect effect on p values (purple statistically significant)

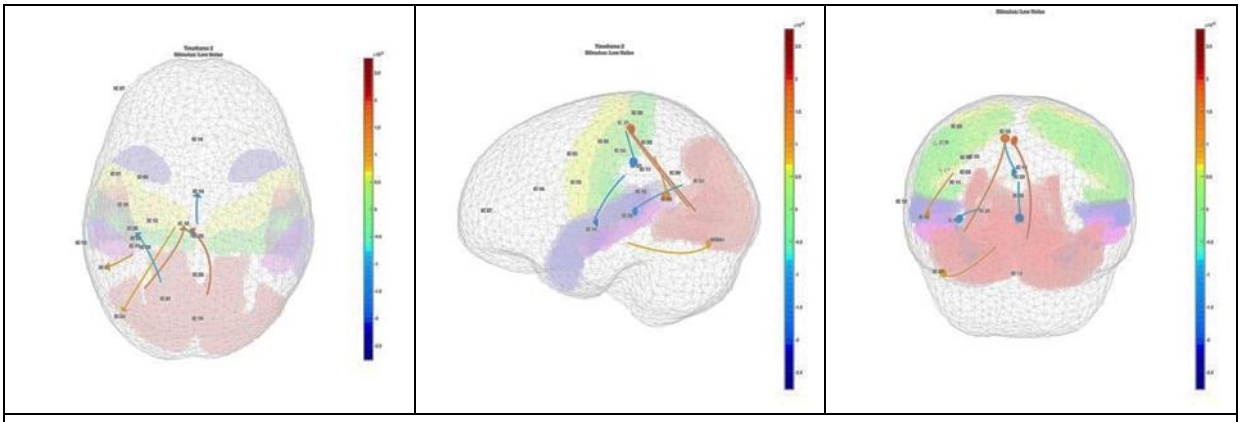
	Correct	Incorrect
All Stimulus	0.2736	0.4546

7.4.3.2 Noise Level

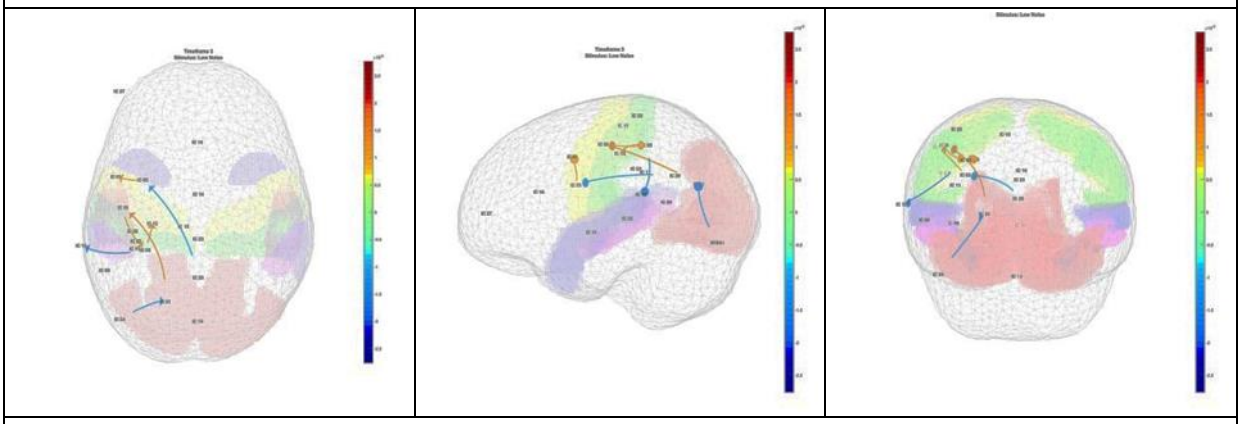
To investigate the effect of the noise level on the connectivity, this was the average across the different stimulus but at the different noise levels.

*Table 5: Low noise connectivity results (scalebar -2.5*10^{-3} to 2.5*10^{-3})*

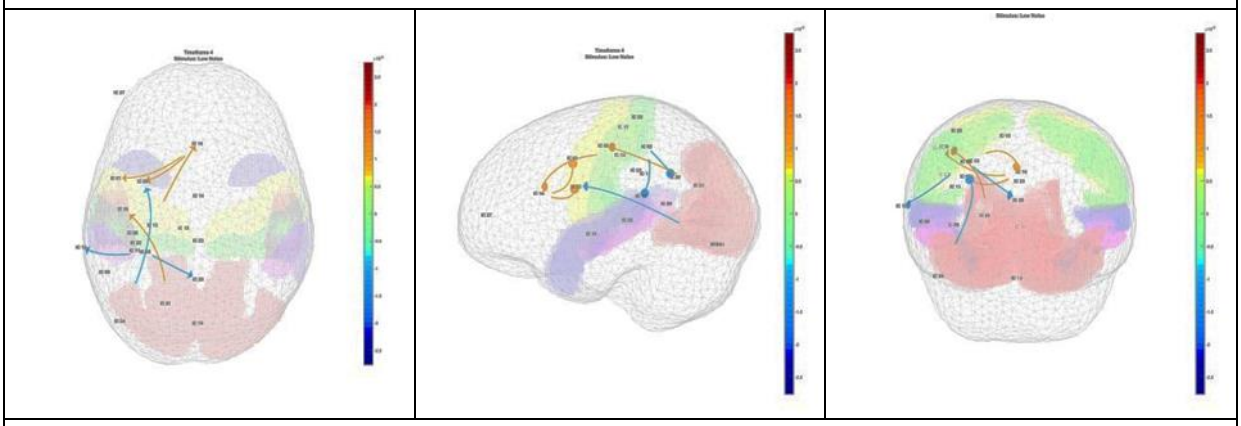
Superior transverse view	Left sagittal view	Posterior coronal view
0-200 ms		
		
100-300 ms		



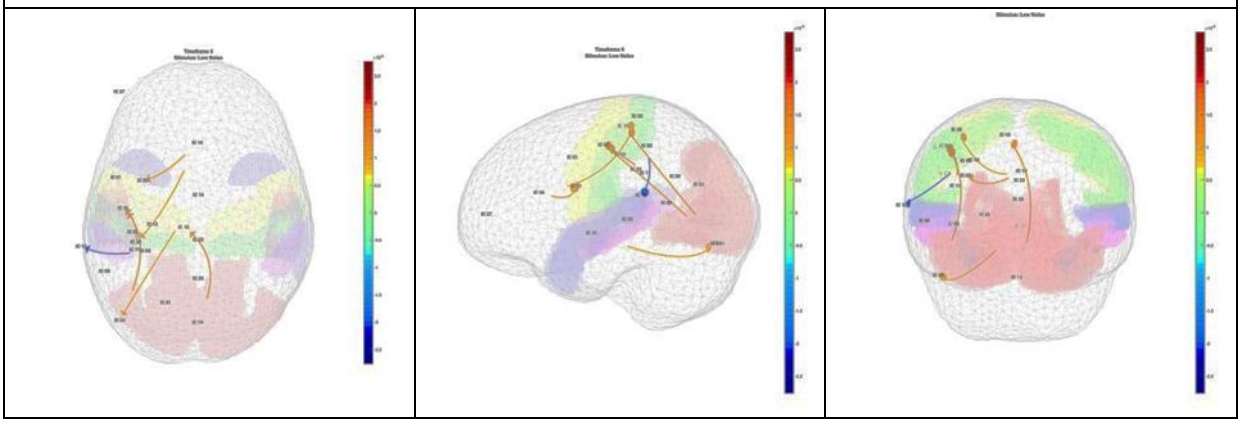
200-400 ms



300-500 ms

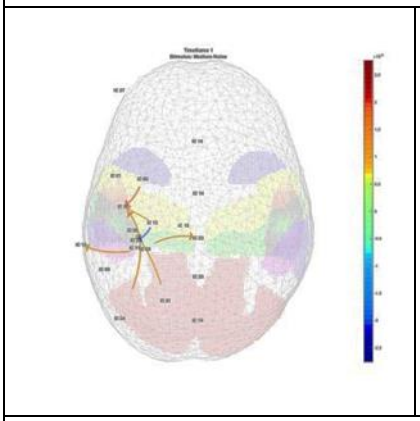
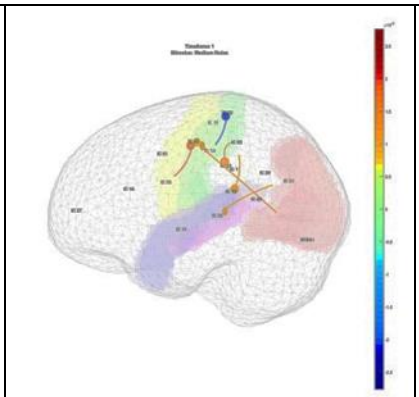
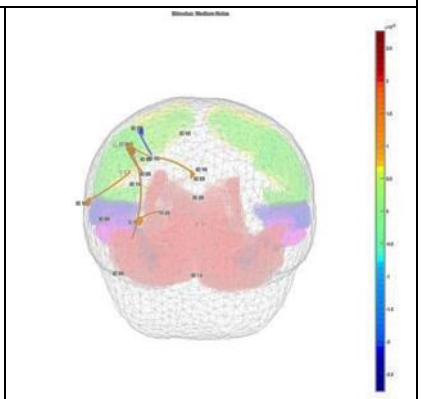
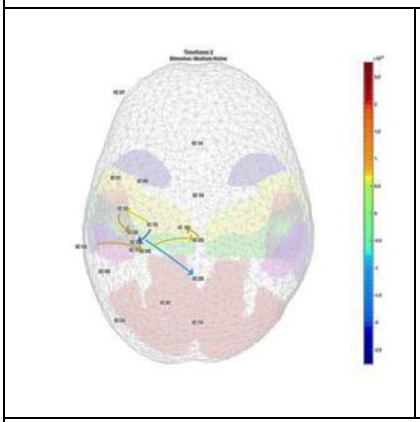
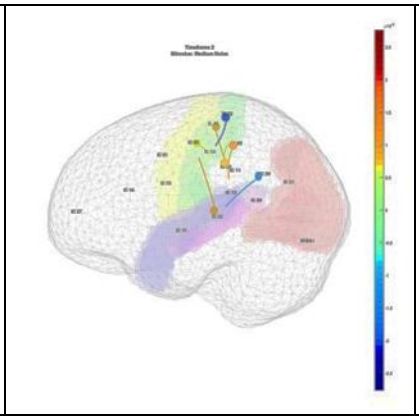
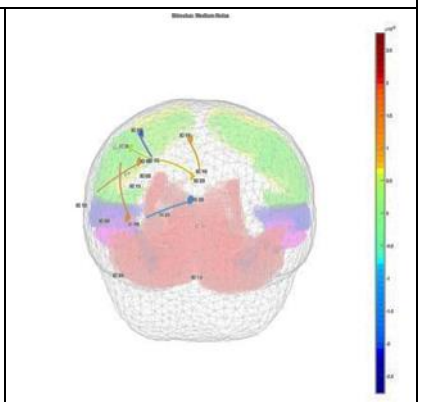
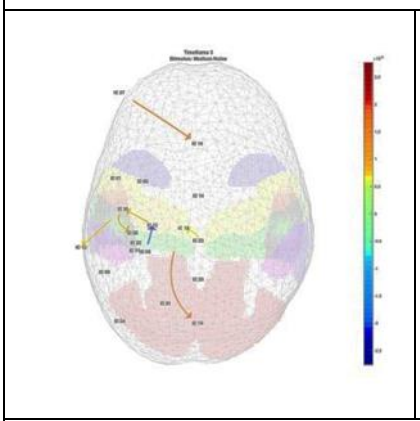
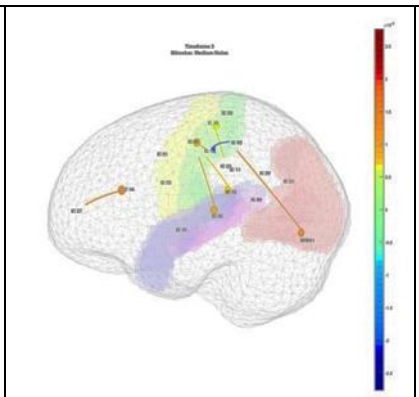
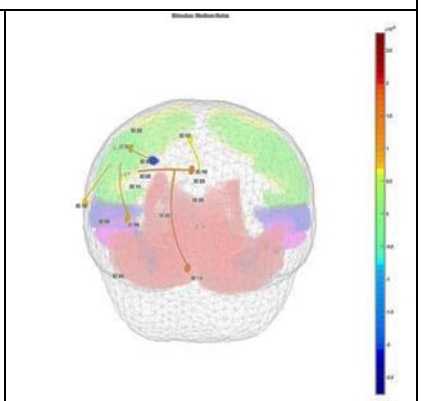


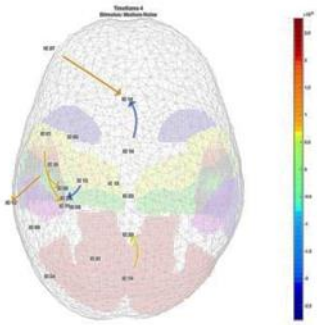
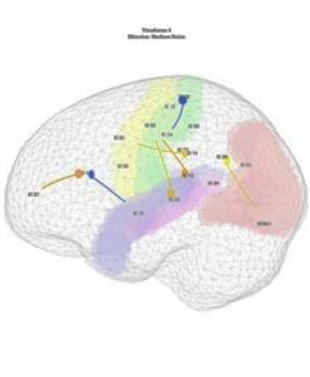
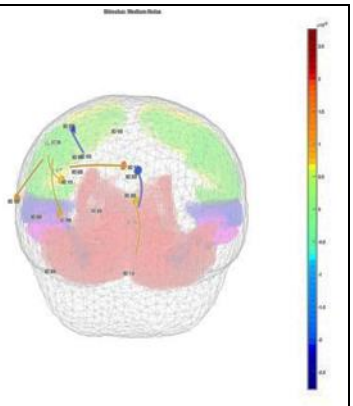
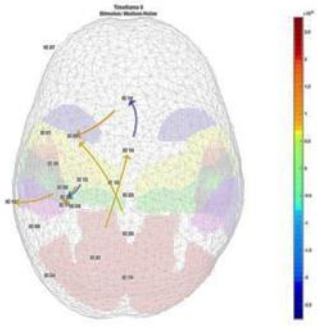
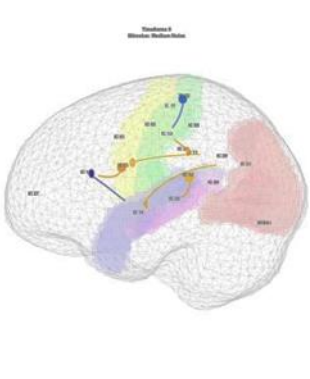
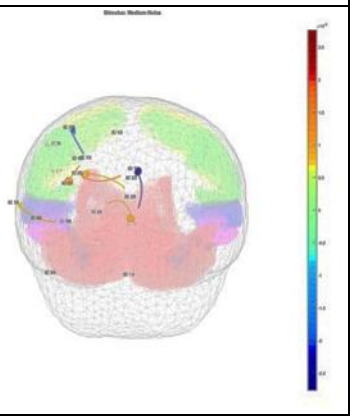
400-600 ms



As seen in Table 5, at low noise the stimulus is easy to hear and therefore not a difficult task meaning there is little frontal activity occurring. Initially, there is an increase in connection between the sensory components to the STS. There are signals sent to the visual cortex, and some return from it. There are no signals from the STS.

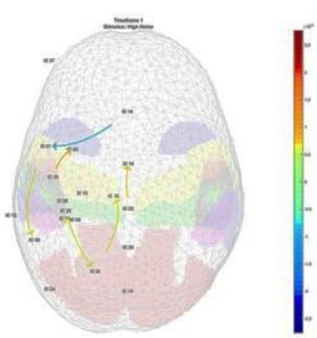
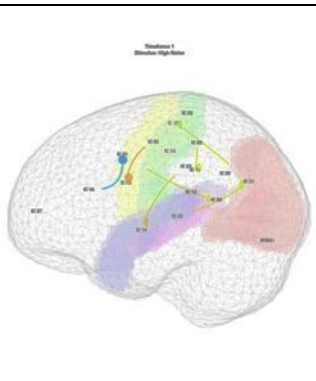
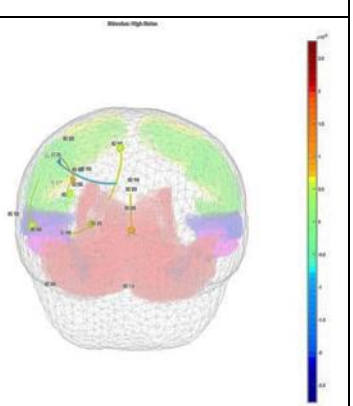
*Table 6: Medium noise connectivity results (scalebar -2.5*10^{-3} to 2.5*10^{-3})*

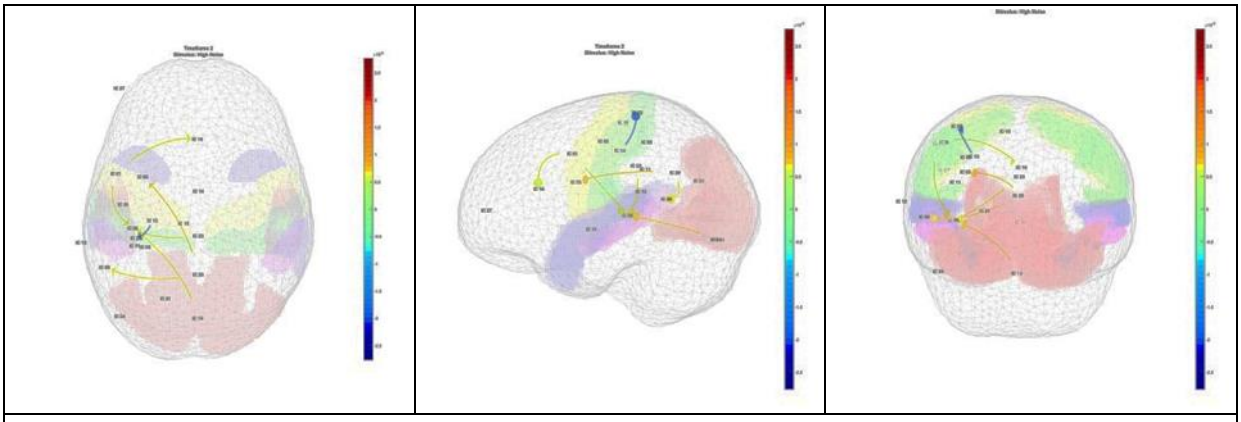
Superior transverse view	Left sagittal view	Posterior coronal view
0-200 ms		
		
100-300 ms		
		
200-400 ms		
		
300-500 ms		

		
400-600 ms		
		

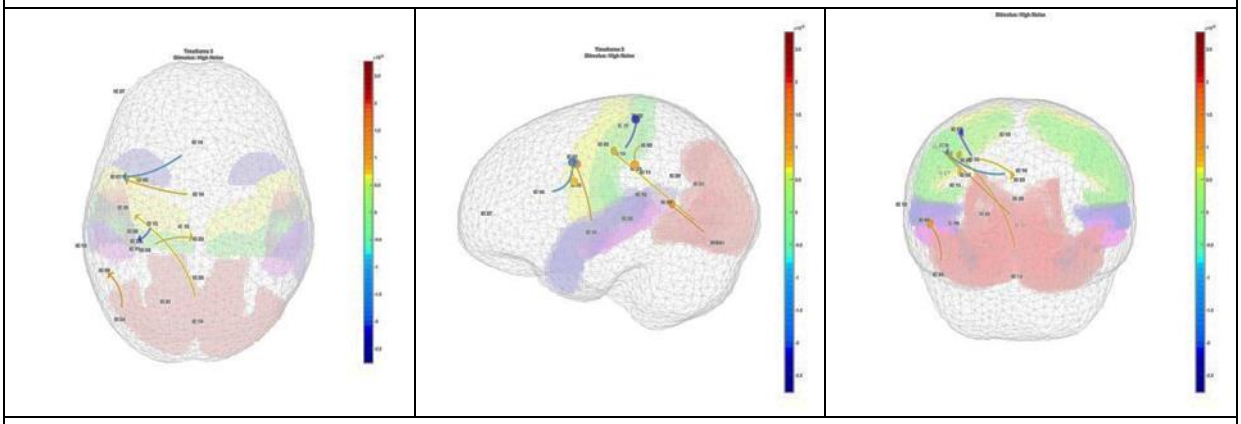
The results of the medium noise connectivity are as seen in Table 6. There is some frontal processing showing there is a requirement of processing for the more difficult task. There is an initial increased connection from the visual cortex to the STS initially. Then there is connection from the STS and visual cortex to auditory cortex.

*Table 7: High noise connectivity results (scalebar -2.5*10^{-3} to 2.5*10^{-3})*

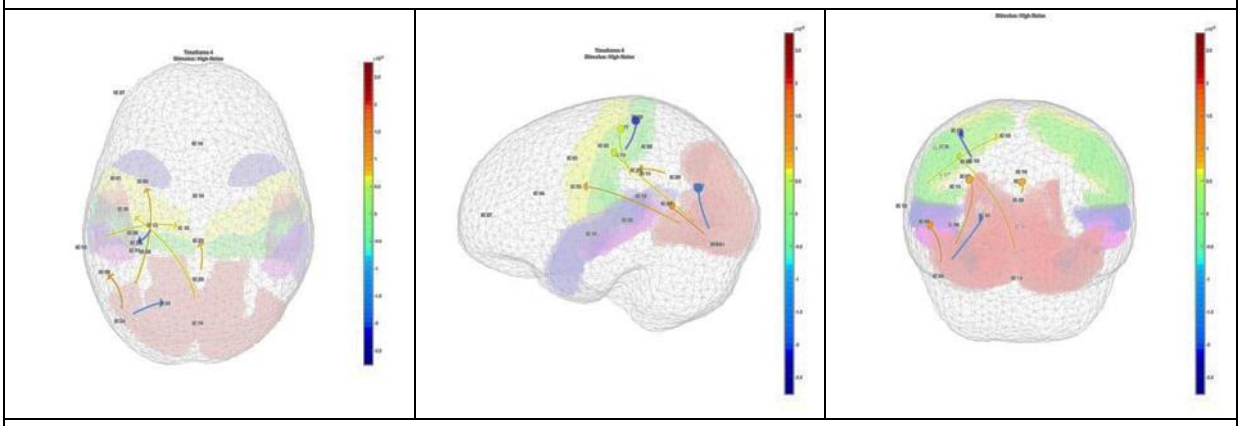
Superior transverse view	Left sagittal view	Posterior coronal view
0-200 ms		
		
100-300 ms		



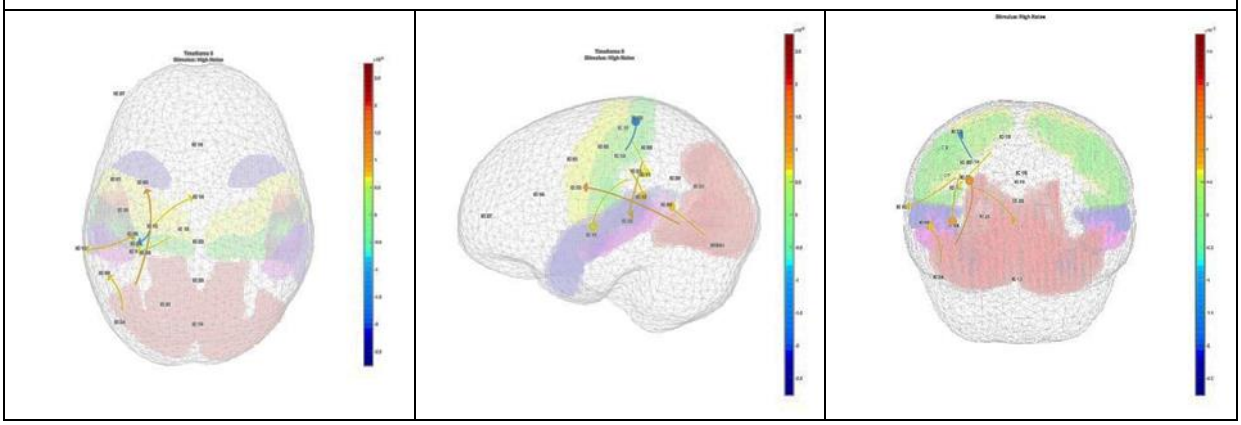
200-400 ms



300-500 ms



400-600 ms

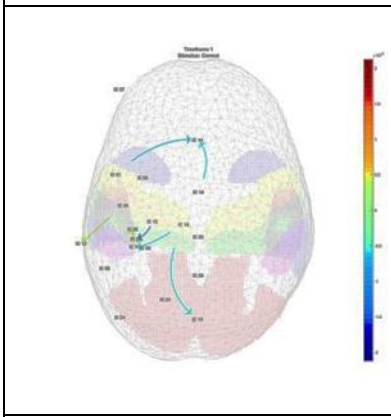
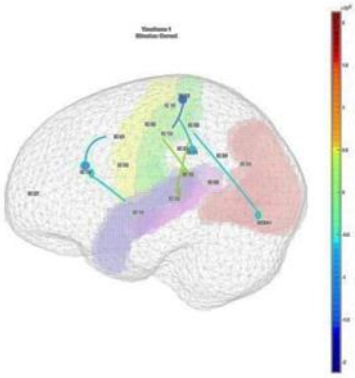
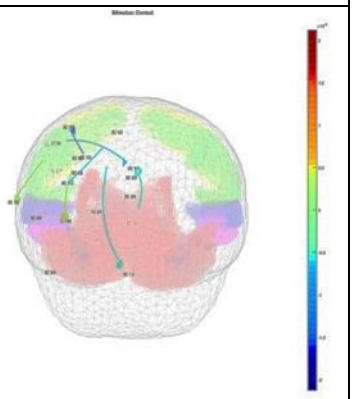
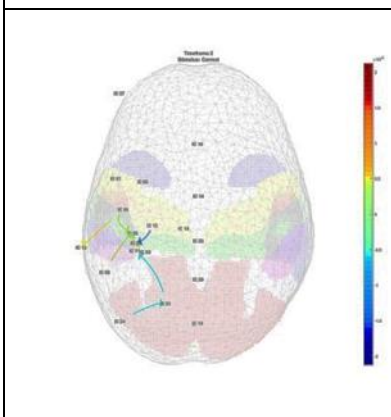
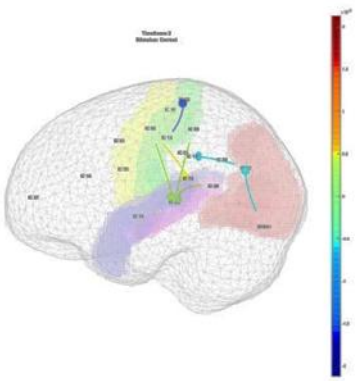
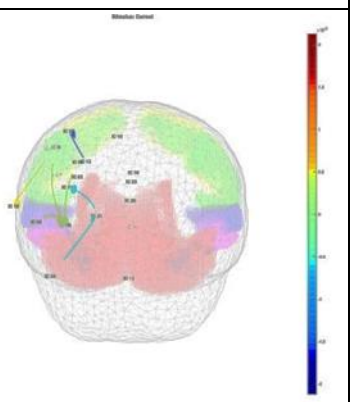


The results of the high noise connectivity can be seen in Table 7. There is some frontal processing, although not as much as expected. There are immediately signals from the STS to the visual cortex. Then there are signals back to the STS from visual and auditory cortex. The signals then go out from the visual cortex.

7.4.3.3 Correct compared to incorrect

The correct responses were compared to the incorrect for comparison.

*Table 8: Correct connectivity responses (scalebar -2*10^{-3} to 2*10^{-3})*

Superior transverse view	Left sagittal view	Posterior coronal view
0-200 ms		
		
100-300 ms		
		
200-400 ms		

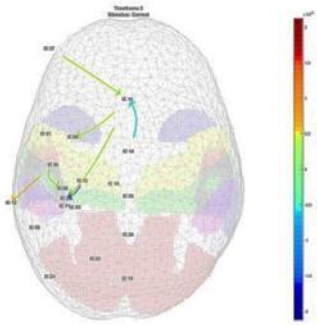
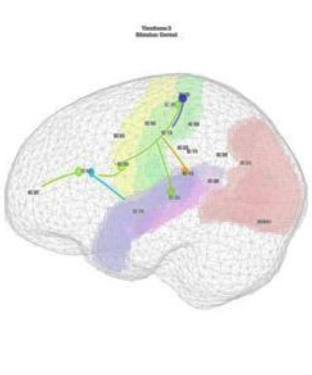
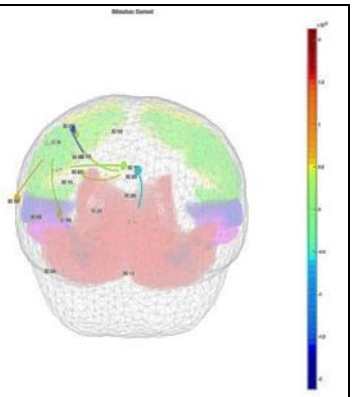
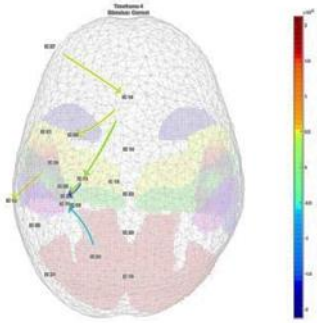
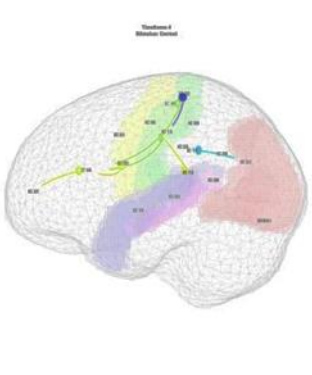
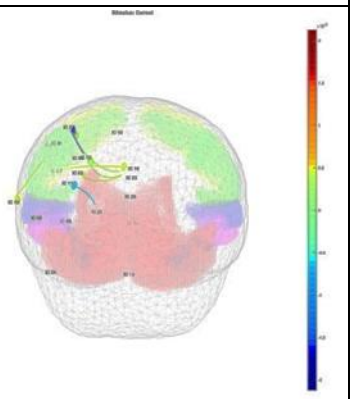
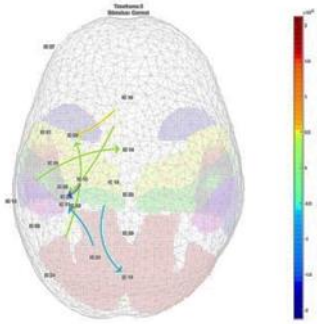
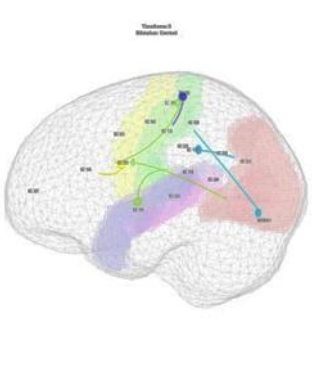
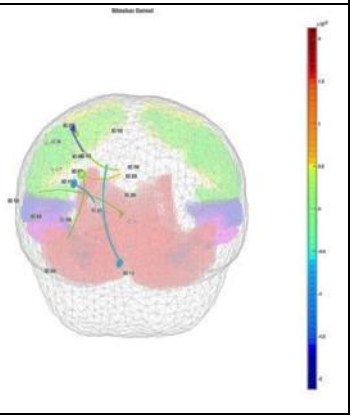
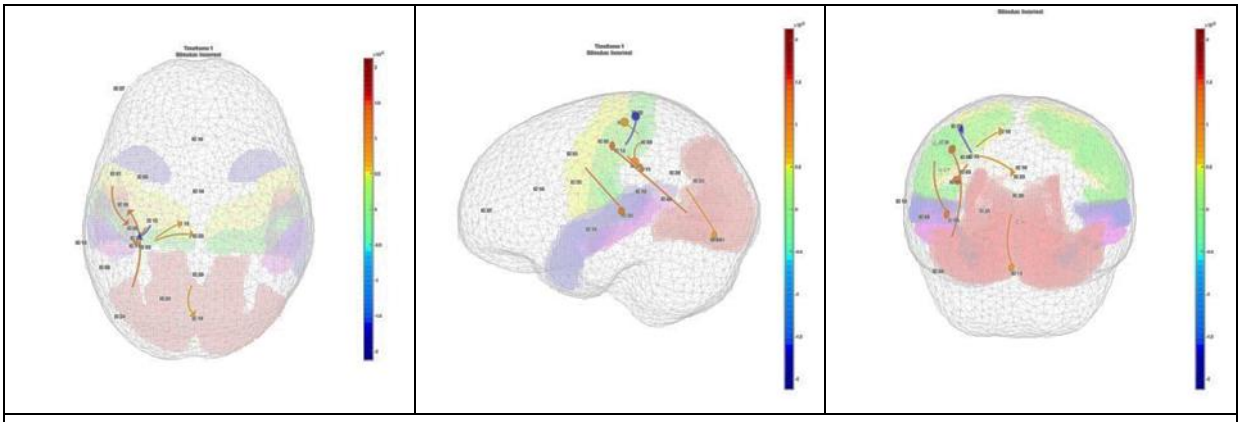
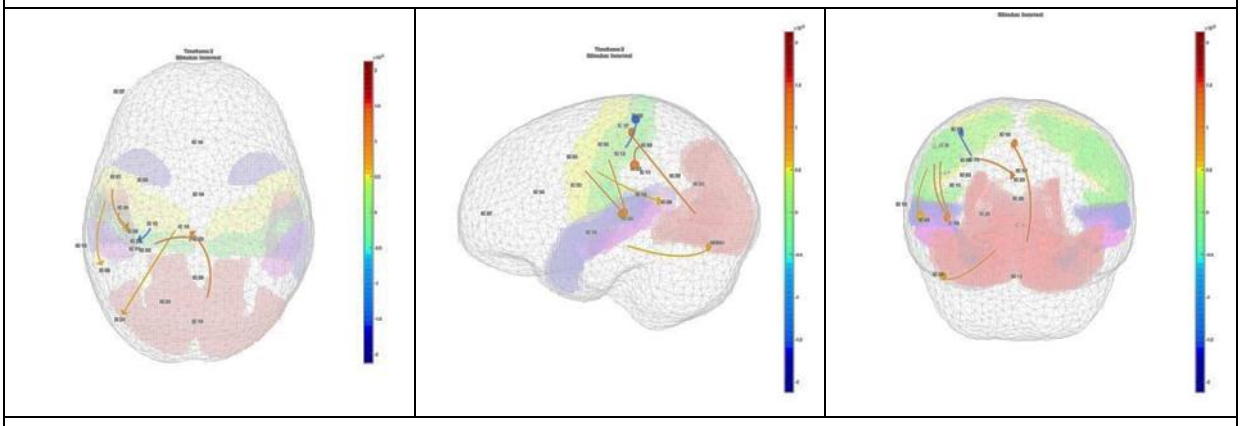
		
300-500 ms		
		
400-600 ms		
		

*Table 9: Incorrect connectivity responses (scalebar -2*10^{-3} to 2*10^{-3})*

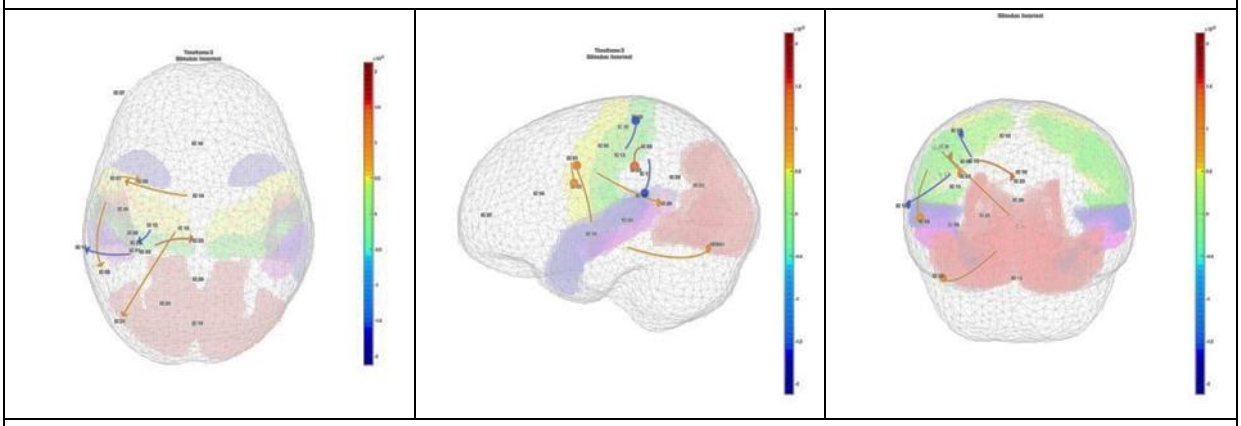
Superior transverse view	Left sagittal view	Posterior coronal view
0-200 ms		



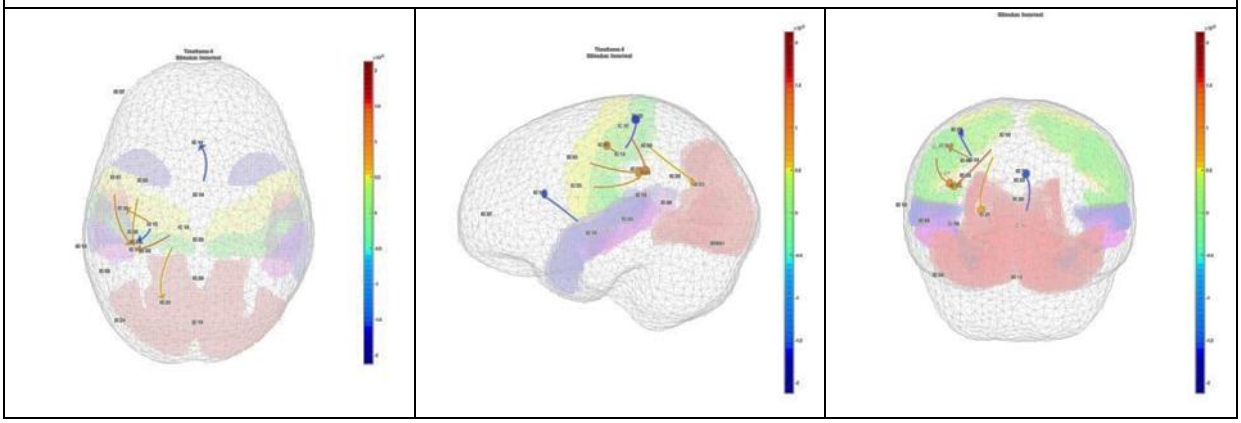
100-300 ms

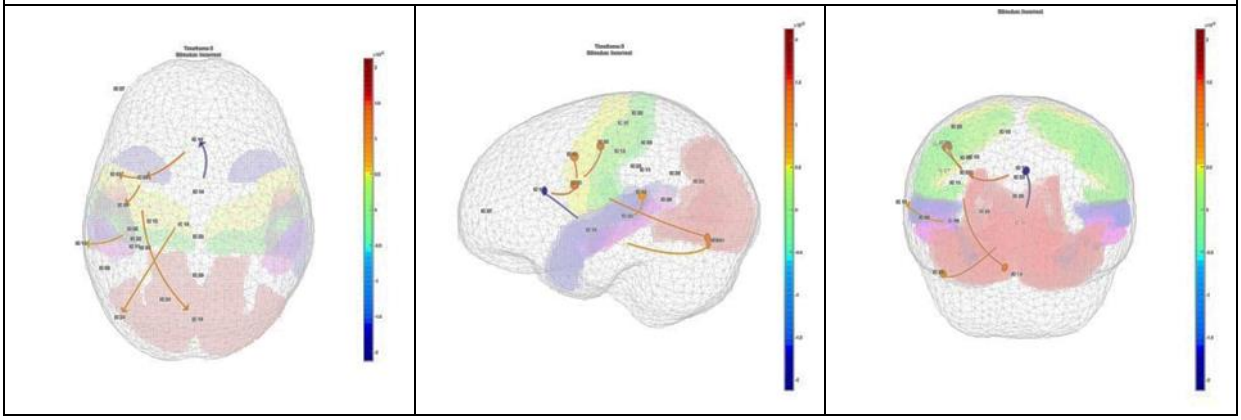


200-400 ms



300-500 ms



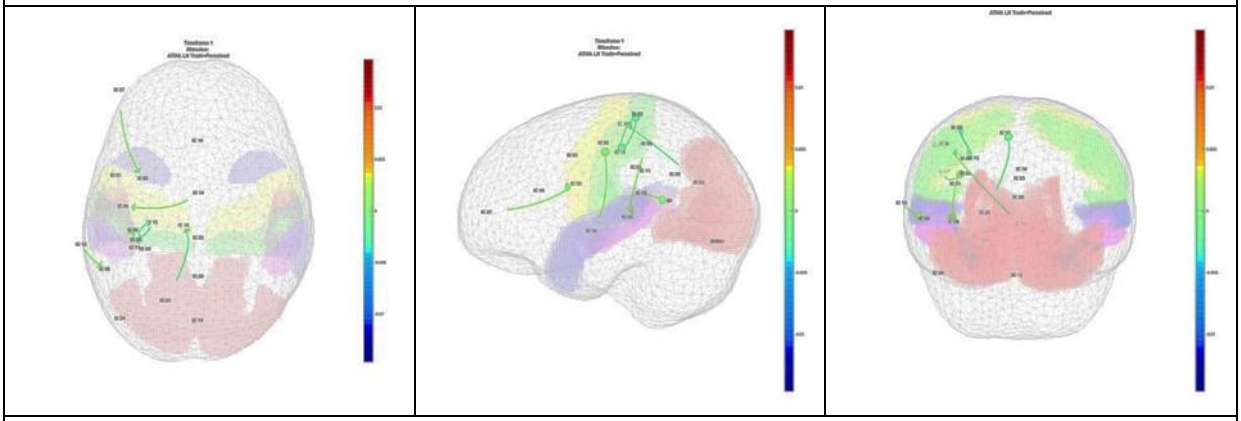
400-600 ms		
		

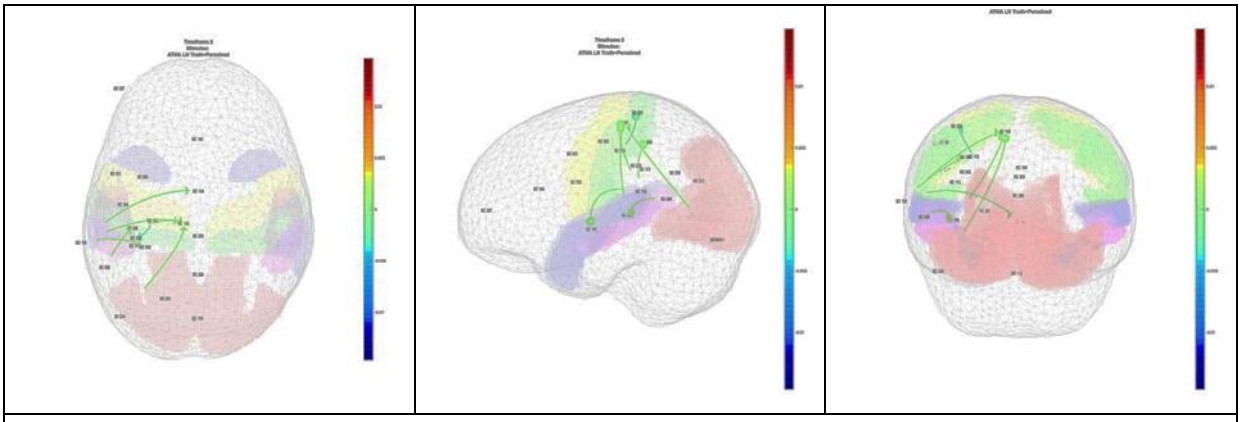
In the correct there is initially a connection to the STS. In the incorrect there was connection within the STS but not externally to. There were also more decreases in frontal activity implying less thinking and processing.

7.4.3.4 McGurk Effect

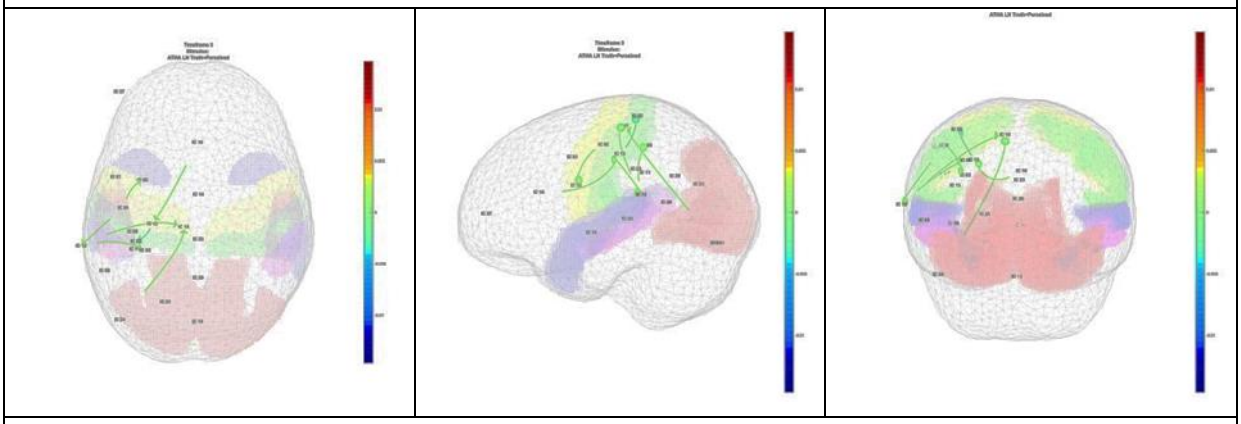
To investigate the McGurk effect on the data there was an investigation done comparing the ATHA response to an AGGA response at the same noise level – low noise. The comparison of ATHA and AGGA for the McGurk effect was chosen since there was the confusion in the choosing of ABBA for ATHA so this would not be the best choice. This is just the ATHA chosen correctly and the AGGA chosen correctly.

*Table 10: ATHA chosen correctly connectivity response (scalebar -15*10^{-3} to 15*10^{-3})*

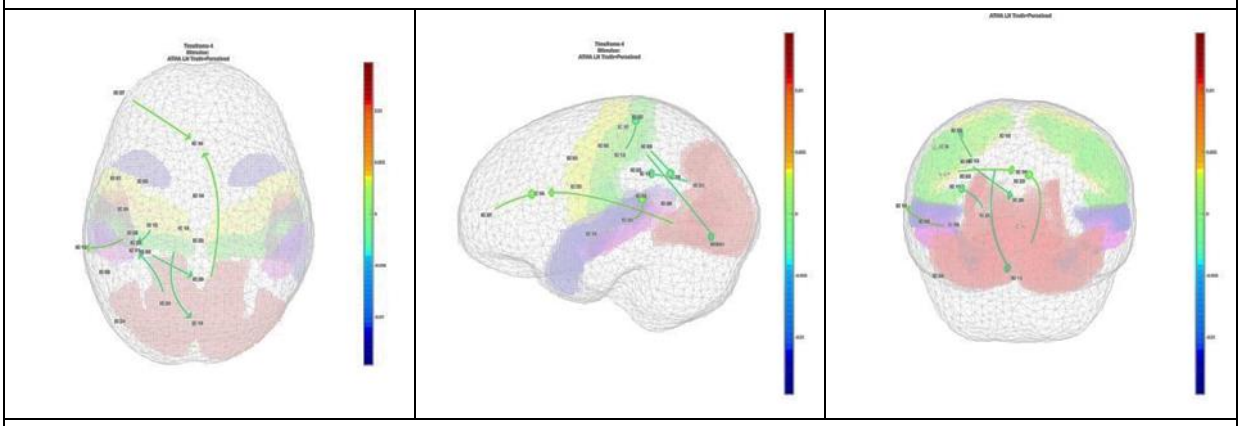
Superior transverse view	Left sagittal view	Posterior coronal view
0-200 ms		
		
100-300 ms		



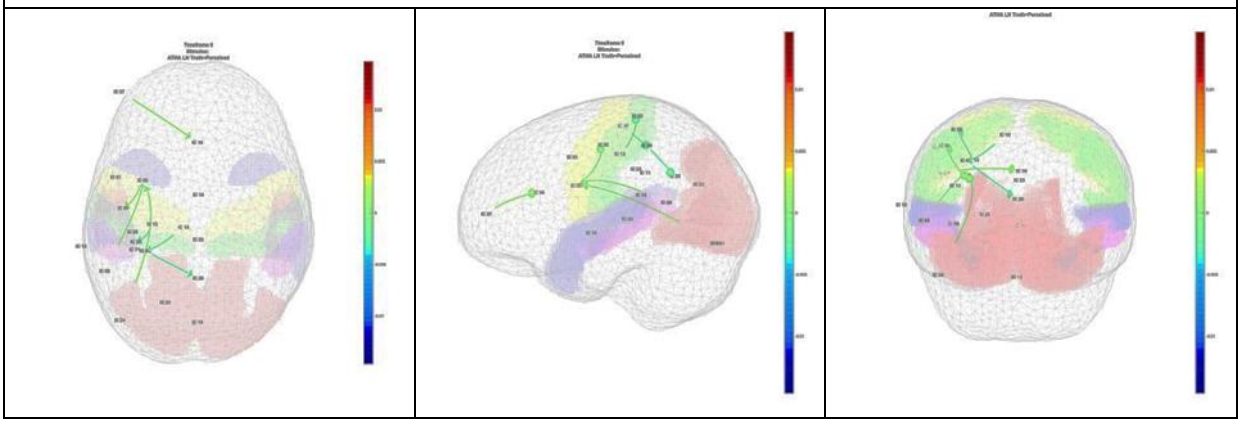
200-400 ms



300-500 ms

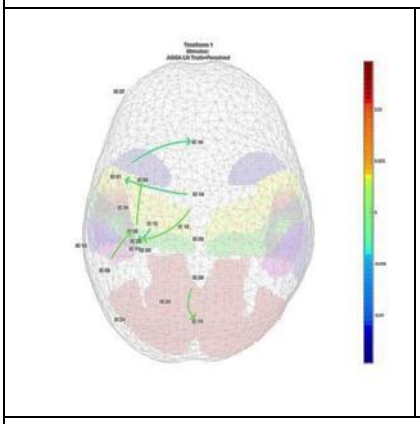
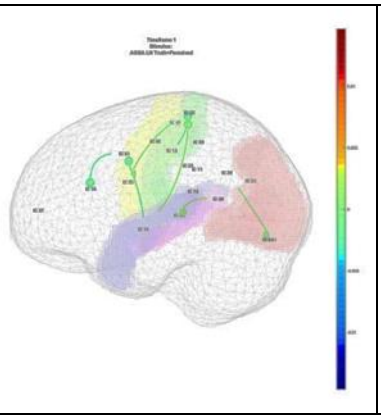
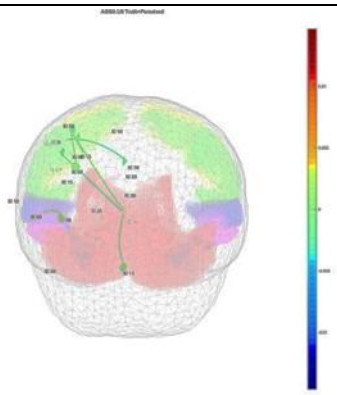
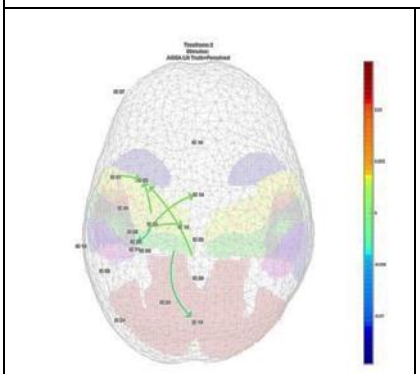
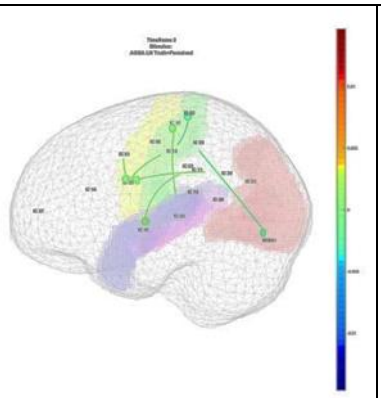
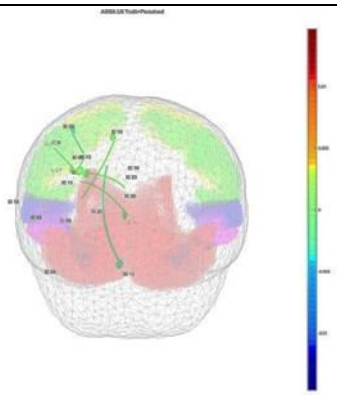
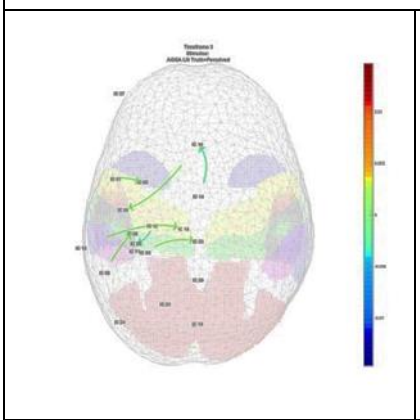
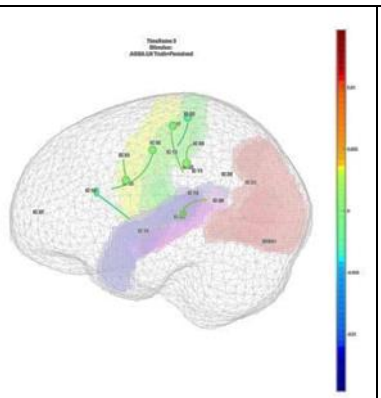
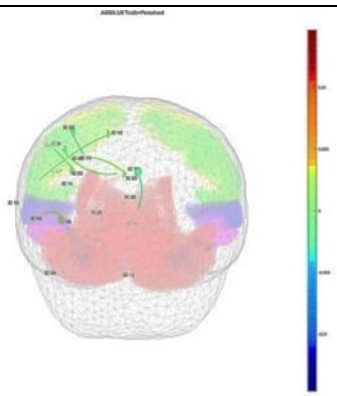


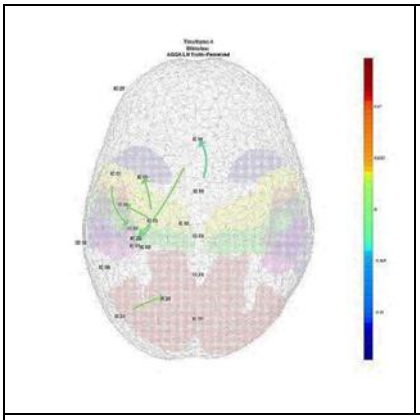
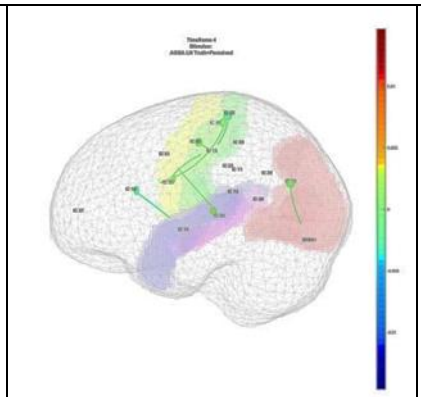
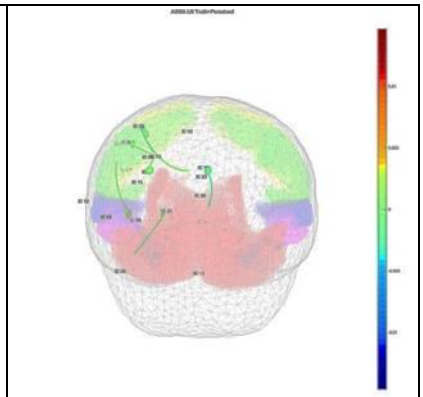
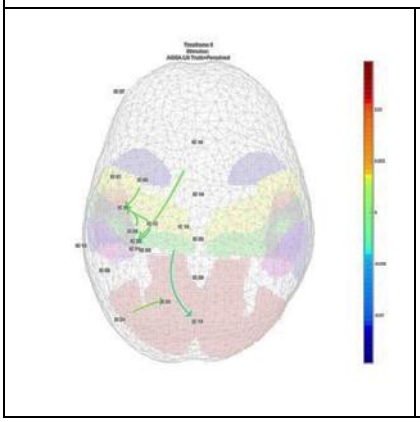
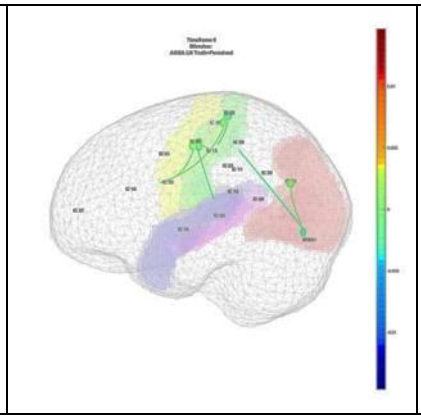
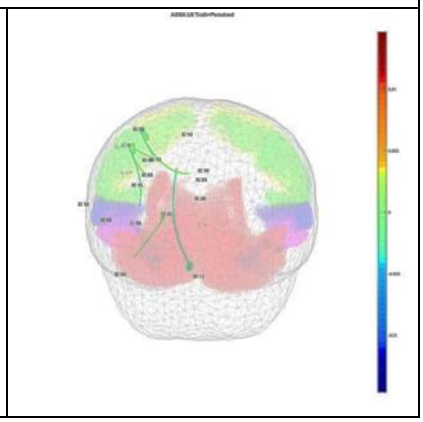
400-600 ms



In the ATHA chosen correctly, as seen in Table 10, it can be seen there are connections to the STS from auditory cortex and sensory components. Then there are connections from the STS to the auditory, then signals sent out from there.

*Table 11: AGGA chosen correctly connectivity response (scalebar -15*10^{-3} to 15*10^{-3})*

Superior transverse view	Left sagittal view	Posterior coronal view
0-200 ms		
		
100-300 ms		
		
200-400 ms		
		
300-500 ms		

		
400-600 ms		
		

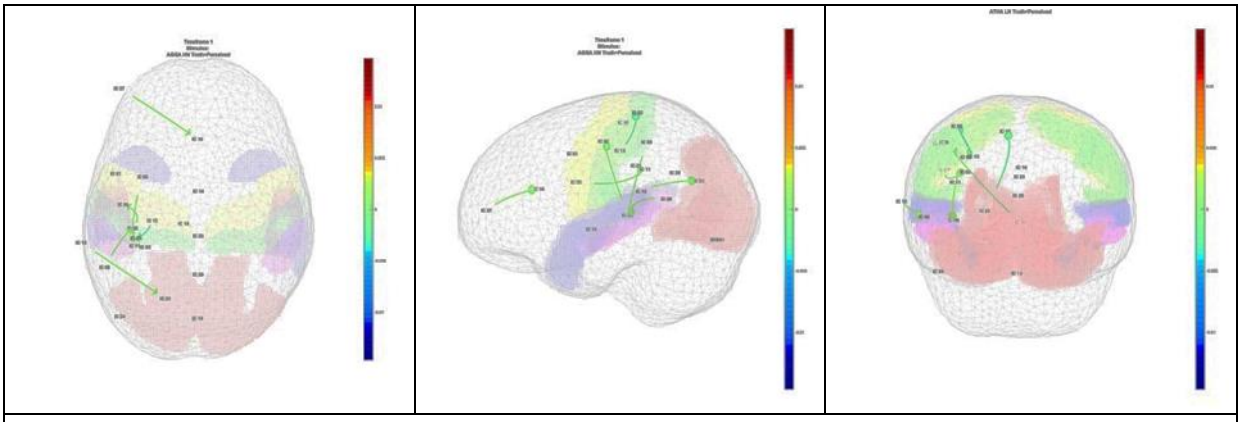
In AGGA connectivity, seen in Table 11, there is an initial signal from somatosensory to the STS. There are also connections of the auditory cortex and the STS. There are connections from the visual cortex as well. This is similar to the ATHA would imply that there is the integration of the STS and visual always, not just when it is needed.

7.4.3.5 Audio only compared to audio-visual

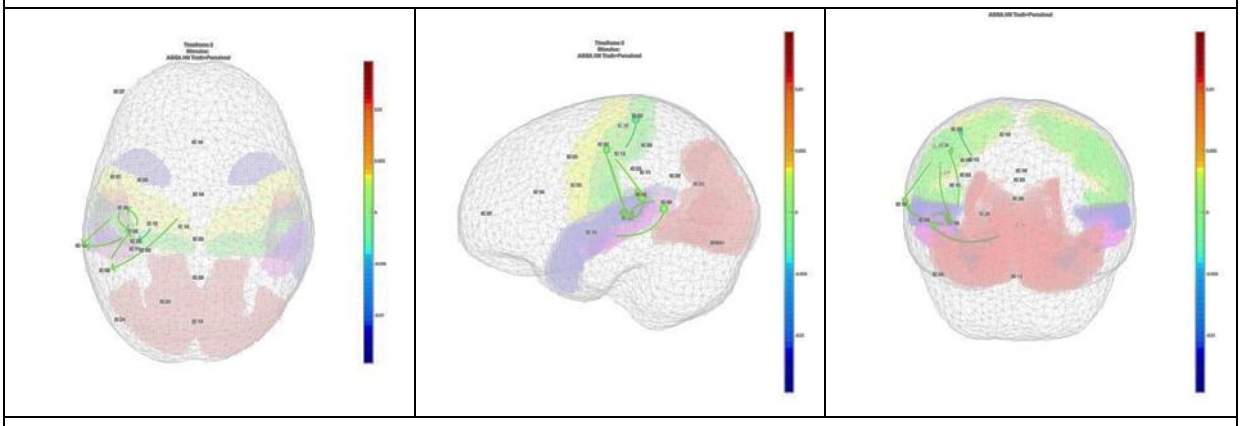
A comparison was done of the connections when a visual stimulation was presented with the audio as in comparison to when just an auditory stimulation was presented. There was confusion in ABBA guessing ATHA, so the comparison was made between AGGA and AGGA-AO.

*Table 12: AGGA at high noise chosen correctly (scalebar -15*10^{-3} to 15*10^{-3})*

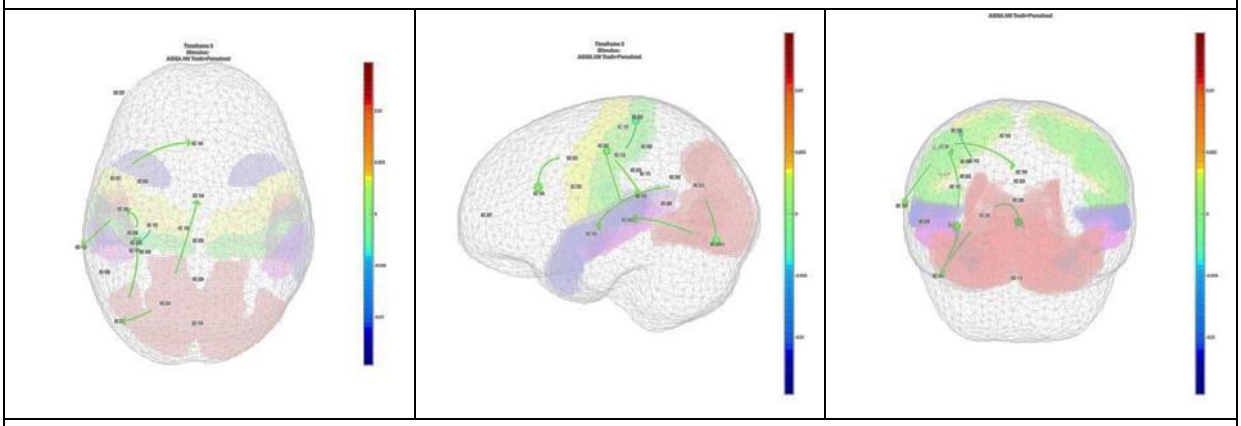
Superior transverse view	Left sagittal view	Posterior coronal view
0-200 ms		



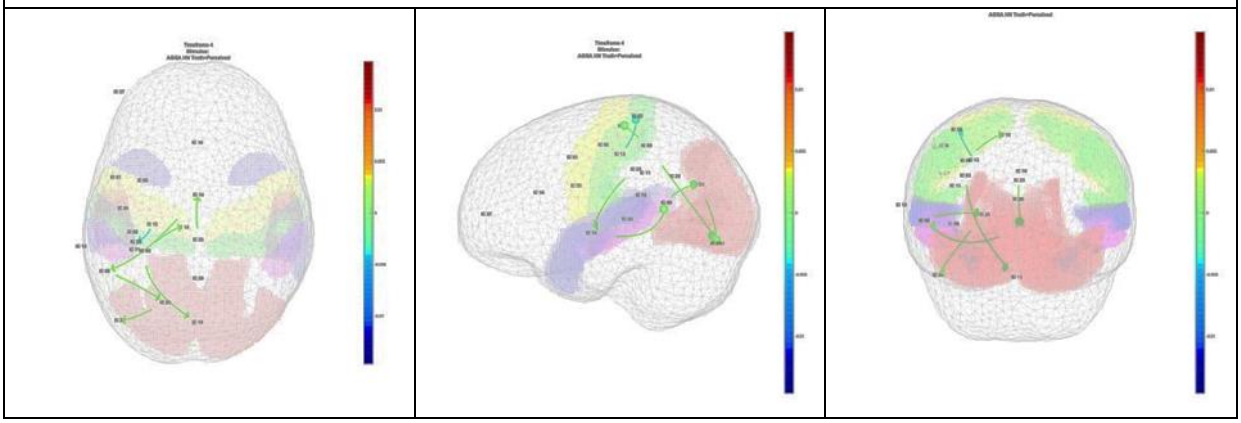
100-300 ms

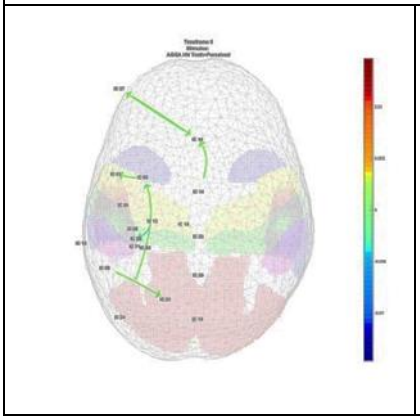
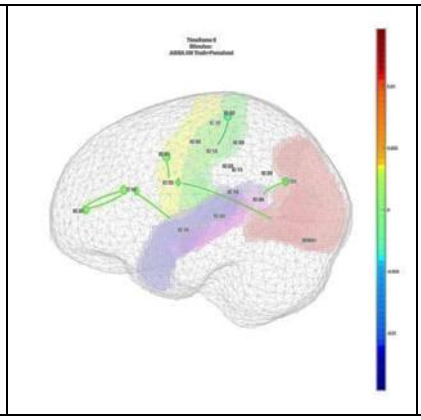
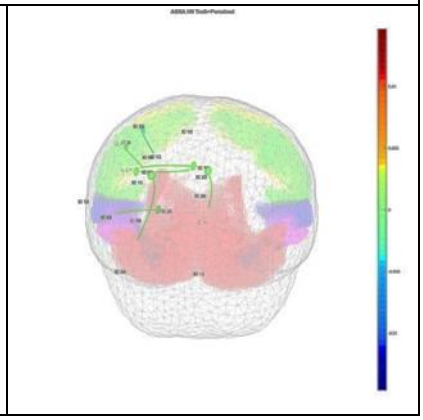


200-400 ms



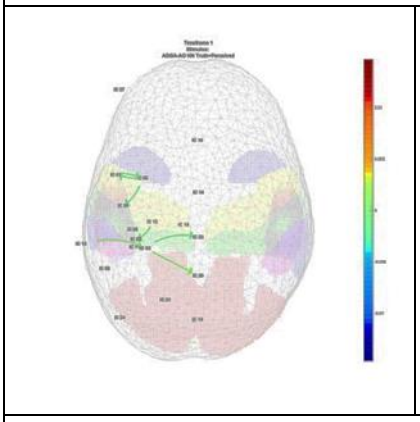
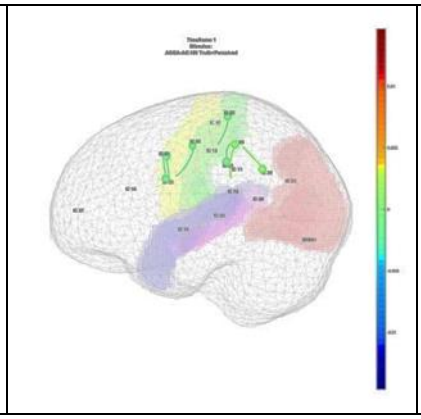
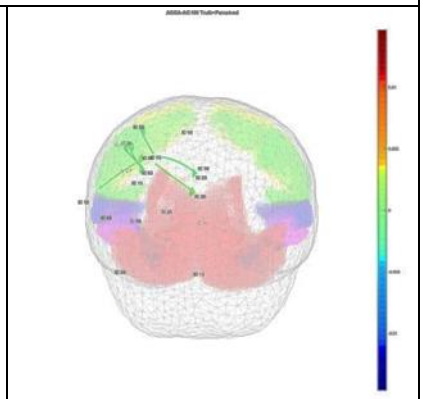
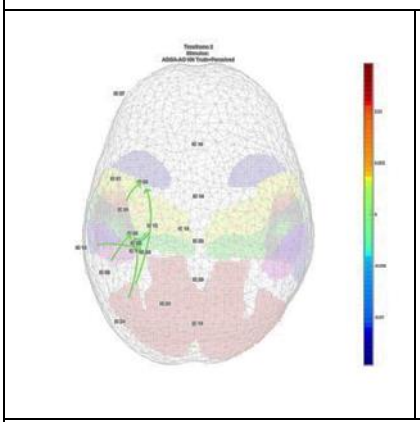
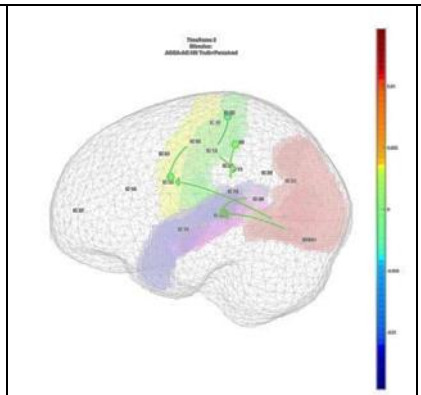
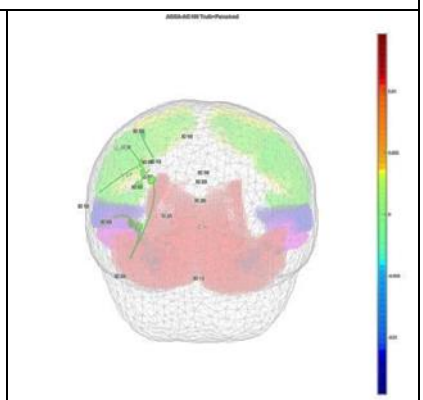
300-500 ms

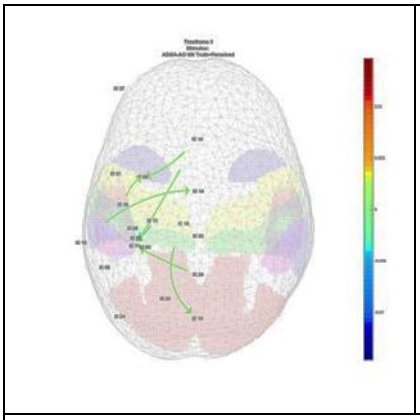
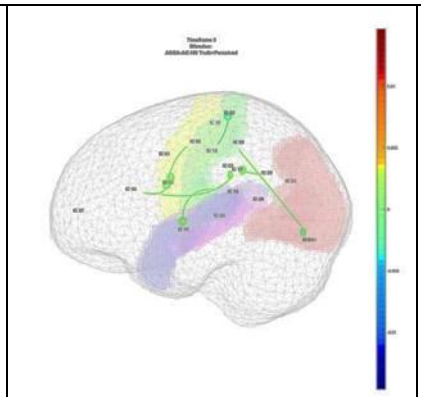
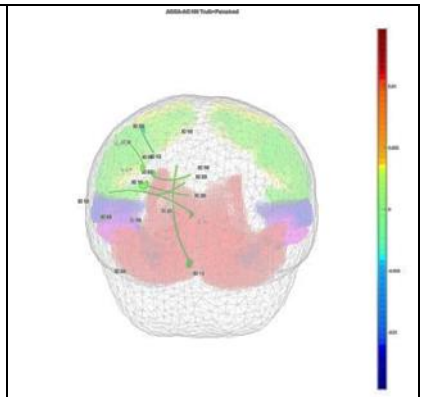
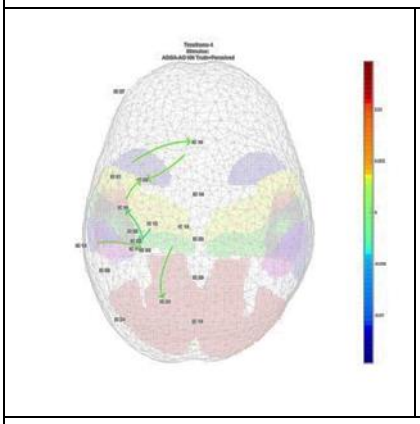
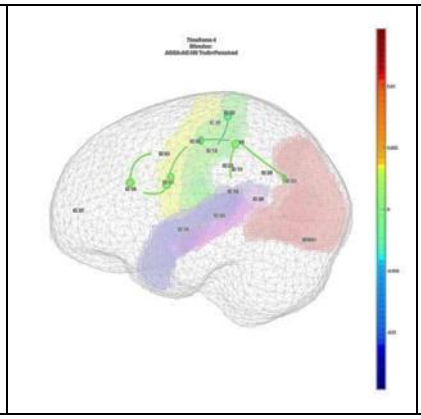
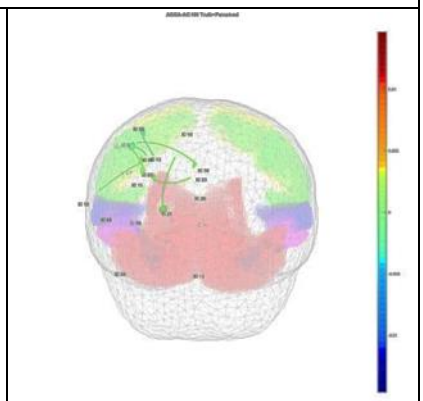
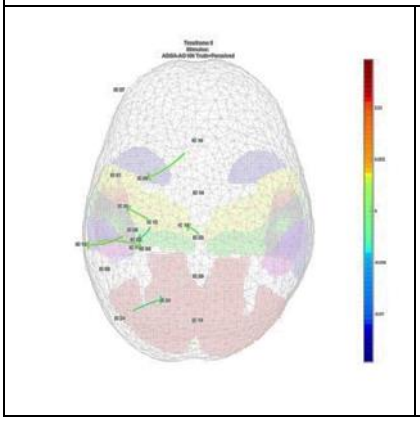
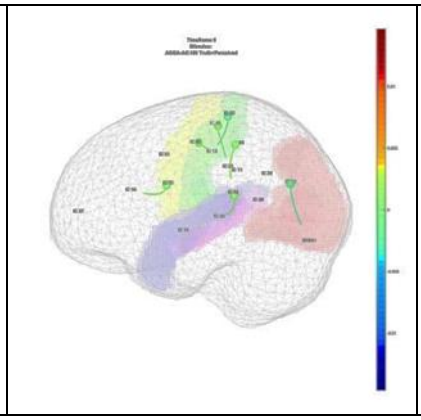
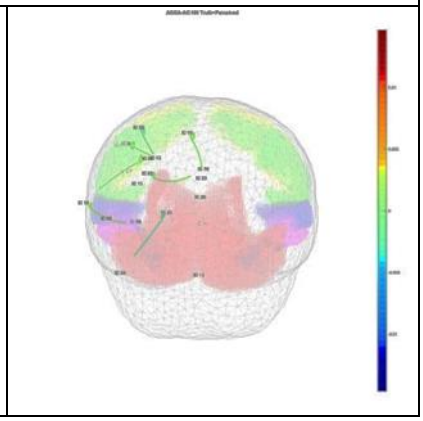


400-600 ms		
		

In the AGGA with video, Table 12, there is the obvious recruitment of the STS in those initial times of recruitment and then later connections of the auditory cortex and visual with the sensory and frontal regions.

*Table 13: AGGA audio-only at high noise chosen correctly (scalebar -15*10^{-3} to 15*10^{-3})*

Superior transverse view	Left sagittal view	Posterior coronal view
0-200 ms		
		
100-300 ms		
		
200-400 ms		

		
300-500 ms		
		
400-600 ms		
		

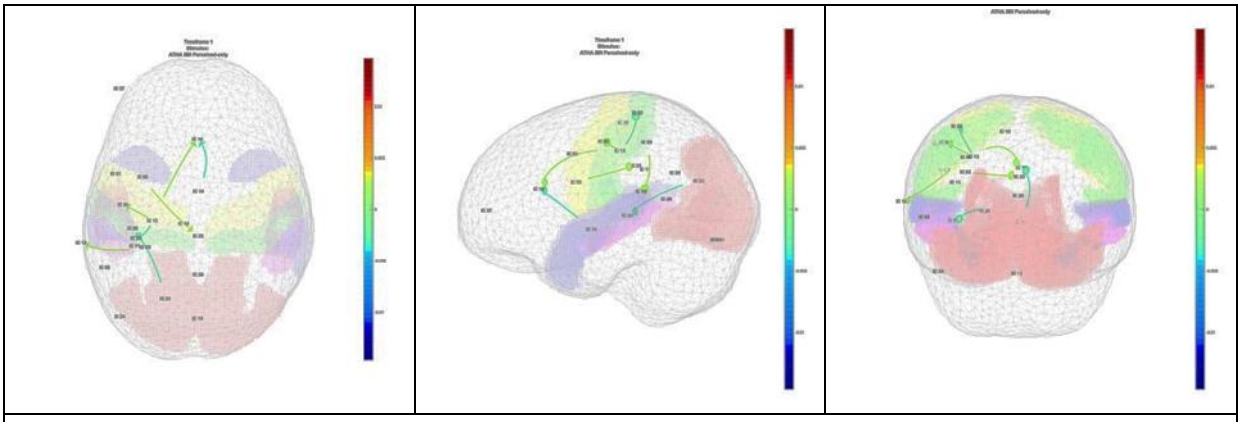
In the AGGA- audio only, Table 13, there is recruitment of the visual cortex and STS, but the STS has little output. There is still output of the visual cortex.

7.4.3.6 Preconditioning of Stimulus

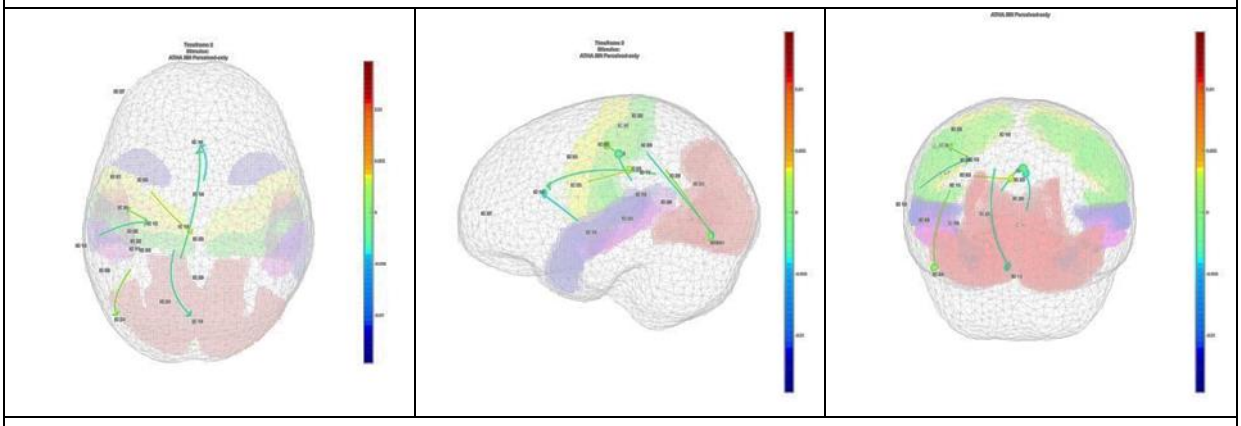
From the behavioural results there was this mistake of choosing ABBA audio only as ATHA at high noise levels. The connectivity under these conditions was investigated.

*Table 14: ATHA chosen but not correct (scalebar -15*10^{-3} to 15*10^{-3})*

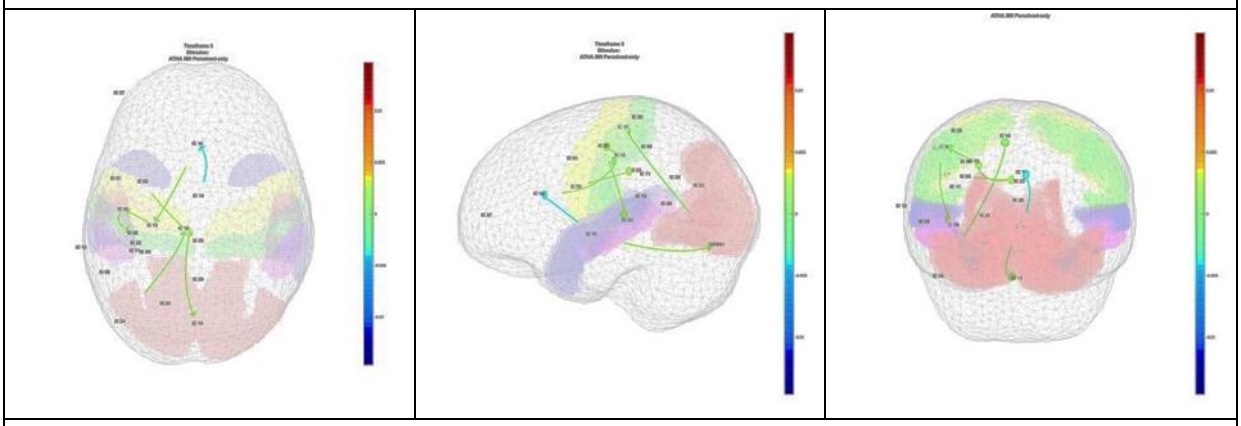
Superior transverse view	Left sagittal view	Posterior coronal view
0-200 ms		



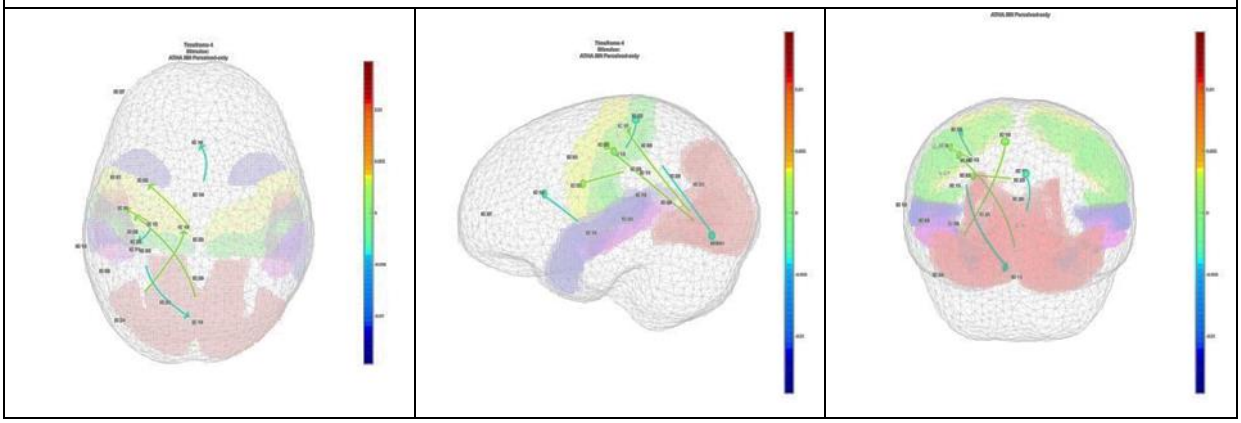
100-300 ms

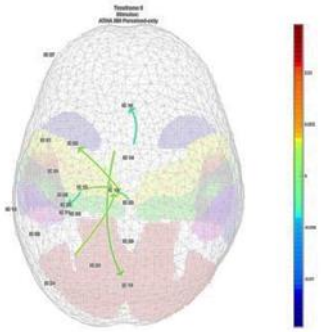
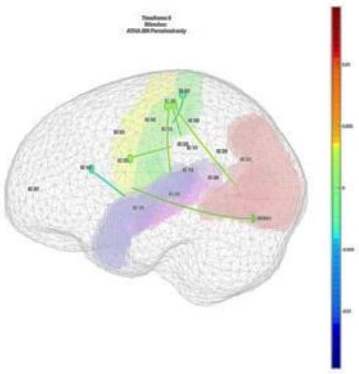
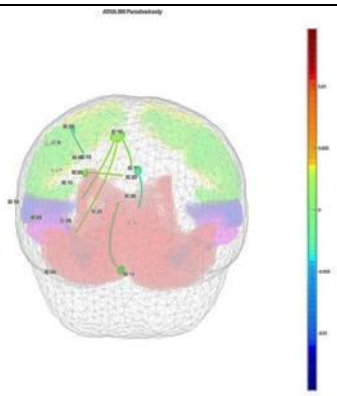


200-400 ms



300-500 ms



400-600 ms		
		

As can be seen from Table 14, there is a more frontal activity and then later recruitment of the STS and the visual cortex. This implies it is from memory of a similar sound that then recruits the STS to decipher and the result is confusing ABBA with ATHA.

8 DISCUSSION

In comparison of the conditional granger causality and transfer entropy, conditional granger causality appeared to have much better results. Transfer entropy is useful in the study as it is a very fast computing measure so that it can be applied in an initial study. However, it is not the best measure to use for EEG applications. It does improve in accuracy as the EEG noise increases but still gives a worse measure than conditional granger causality. Conditional granger causality appeared to be a good measure of connectivity. There were statistically significant results for many of the relevant stimulus of the conditional granger causality measure.

When the different stimuli were averaged across noise level there were many different results. At low noise the stimulus is not a difficult task so there is little frontal activity occurring. Initially, there is an increase in connection between the sensory components to the STS. There are signals sent to the visual cortex, and some return from it. There are no signals from the STS. At medium noise there is some frontal processing showing there is a requirement of processing for the more difficult task. There is an initial increased connection from the visual cortex to the STS initially. Then there is connection from the STS and visual cortex to auditory cortex. At high noise there is some frontal processing, although not as much as expected. There are immediately signals from the STS to the visual cortex. Then there are signals back to the STS from visual and auditory cortex. The signals then go out from the visual cortex. This shows that the noise level does have an effect on the connections. As the noise level increases, so does the connection to the STS, in order to process these signals. It shows how the connection to the STS was always within these first 0-300ms and then slightly later from the STS. The frontal processing in the initial stages of 0-400ms is generally with an increase in difficulty of the task. The signal conduction appears to be a signal sent to the STS and the visual cortex, and then if needed there is a signal returned from then to the auditory cortex and sensory components.

In the comparison of the correct and incorrect guessed stimuli, there were a few major differences. In the correct, there is initially a connection to the STS. In the incorrect, there was connection within the STS but not externally to. There were also more decreases in frontal activity implying less thinking and processing. This shows that the STS were recruited in the correctly guessed as compared to incorrect. A very interesting result showing that the recruitment of the STS helps in the processing of the audio-visual information, along with frontal processing.

The ATHA response and an AGGA response at the same noise level, low noise, were compared in order to investigate the McGurk effect on the connections in the brain. The comparison of ATHA and AGGA for the McGurk effect was chosen, since there was the confusion in the choosing of ABBA for ATHA, so this would not be the best choice. This was just the ATHA chosen correctly and the AGGA chosen correctly. In the ATHA chosen correctly, it can be seen there are connections to the STS from auditory cortex and sensory components. Then there are connections from the STS to the auditory, and then signals sent

out from there. In AGGA connectivity, there is an initial signal from somatosensory to the STS. There are also connections of the auditory cortex and the STS. There are connections from the visual cortex as well. This is similar to the ATHA would imply that there is the integration of the STS and visual always, not just when it is needed. So, the brain functions by always recruiting the STS in functioning when the stimulus is difficult to interpret, and then when there is a need for it due to a difference between the visual and auditory, from signals sent to the STS, then it sends signals to the auditory cortex to process this.

In the comparison of a visual stimulation being presented with the audio to when just an auditory stimulation was presented, there were some significant differences. There was confusion in ABBA guessing ATHA, so the comparison was made between AGGA and AGGA-AO. In the AGGA with video, there is the recruitment of the STS in those initial times of recruitment and then later connections of the auditory cortex and visual with the sensory and frontal regions. In the AGGA- audio only, there is recruitment of the visual cortex and STS, but the STS has little output. There is still output of the visual cortex. This builds on the findings in the McGurk effect examination to see that there are signals sent to the STS when the stimulus needs to be interpreted and is difficult. However, when there is no visual input or no conflicting visual input, such as in the McGurk effect, there is no signals sent out of the STS. However, there is always signal still sent out to the auditory cortex or somatosensory cortex, even when there is no conflicting visual or even no visual at all.

There was an interesting behavioural result of participants responding with ATHA for the ABBA audio only stimulus at high noise levels. The connectivity under these conditions was investigated. There is a more frontal activity and then later recruitment of the STS and the visual complex. This implies it is from memory of a similar sound that then recruits the STS to decipher and the result is confusing ABBA with ATHA.

There are several limitations of this study. The sample size of 20 subjects was sufficient in order to do these connectivity measures, but for a better study with better statistical results across more stimuli, there could be more subjects included in this. The other interesting results that were unable to be completed in time, due to the high computational power needed were the frequency banded results. These would be able to show the different types of functioning and see the connections. It results in a better outcome generally, since the other level processing is filtered out. A final limitation of this study would be fact that a general MRI was used to fit the dipoles and other spatial awareness in the subjects. While the use of the Polhemus to 3D locate the electrodes helped in accurate location data, the data would be better fit and located into brain regions if subjects personal MRI were used that could then take in the different brain regions for subjects.

9 CONCLUSIONS

There were a number of results from this study in mapping the temporal dynamics during audio-visual speech processing using connectivity analysis. Conditional granger causality was found to be a much better measure of connectivity than transfer entropy resulting in statistically significant connections. The ATHA stimulus caused there to be recognition of ABBA as ATHA in high noise conditions due to this pre-disposure. This was then found to be a frontal activity that then recruited the STS and visual streams in recognising this from memory. The level of noise caused there to be an increase in connection to the STS and increase in recruitment of the visual cortex as the noise increased, to be able to better understand the speech in noisy conditions. There was shown recruitment of the STS in the correct vs incorrect in the first 300ms, showing it helps in correct processing; although this was not statistically significant. In comparing a McGurk stimulus to a non-McGurk stimulus, it is shown that in high noise conditions there is still this recruitment of the STS in understanding noisy speech, even when this does not change what is perceived. When there is no visual stimulation there are still signals sent to the STS but just no output from it, however interestingly there are still signals sent from the visual cortex, which may be signals that are saying there is no visual. The STS recruitment was almost always a very initial process in the first 200-300ms except when being accessed from memory – such as the case of guessing ATHA in a high noise ABBA, although this was not statistically significant. The recruitment of the STS and visual cortex was generally through the auditory cortex, except when it was low noise then the visual cortex sent signals to the somatosensory cortex or frontal areas.

10 REFERENCES

Aydore, S., Pantazis, D. & Leahya, R., 2013. A Note on the Phase Locking Value and its Properties. *NeuroImage,* Volume 74, pp. 231-244.

Barraclough, N. et al., 2005. Integration of visual and auditory information by superior temporal sulcus neurons responsive to the sight of actions. *Journal of Cognitive neuroscience,* Volume 17, pp. 377-391.

Beauchamp, M., Lee, K., Argall, B. & Martin, A., 2004. Integration of auditory and visual information about objects in superior temporal sulcus. *Neuron,* Volume 41, pp. 809-823.

Bell, A. & Sejnowski, T., 1995. An information-maximization approach to blind separation and blind deconvolution. *Neural Comput.,* 7(6), pp. 1129-59.

Bell, M. & Cuevas, K., 2012. Using EEG to Study Cognitive Development: Issues and Practices. *J Cogn Dev.,* 13(3), p. 281–294.

Blau, V. et al., 2009. Reduced neural integration of letters and speech sounds links phonological and reading deficits in adult dyslexia. *Current biology,* 19(6), pp. 503-508.

Blinowska, K., 2011. Review of the methods of determination of directed connectivity from multichannel data. *Med Biol Eng Comput,* Volume 49, pp. 521-529.

Boliek, C., Keintz, C., Norrix, L. & Obrzut, J., 2010. Auditory-visual Perception of Speech in Children with Learning Disabilities: The McGurk Effect. *Canadian Journal of Speech-Language Pathology and Audiology,* 34(2), pp. 124-131.

Cardoso, J. & Souloumiac, A., 1993. Blind beamforming for non-Gaussian signals. *Radar and Signal Processing,* Volume 140, pp. 362-370.

Cardoso & Souloumiac, 1993. Joint Approximate Diagonalization of Eigenmatrices (JADE). *IEE proceedings-F,* 140(6), pp. 362-370.

Casson, A. et al., 2017. Electroencephalogram. In: *Seamless Healthcare Monitoring.* s.l.:Springer, Cham, pp. 45-81.

Cuevas, K., Raj, V. & Bell, M., 2012. Functional connectivity and infant spatial working memory:A frequency band analysis. *Psychophysiology,* Volume 49, pp. 271-280.

de Gelder, B. et al., 2003. Audio-visual integration in schizophrenia. *Schizophrenia research,* 59(2-3), pp. 211-8.

Delorme, A. & Makeig, S., 2004. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics. *Journal of Neuroscience Methods,* Volume 134, pp. 9-21.

Duffy, F., Iyer, V. & Surwillo, W., 1989. *Clinical electroencephalography and topographic brain mapping.* New York: Springer-Verlag .

Fan, L. et al., 2016. The Human Brainnetome Atlas: A New Brain Atlas Based on Connectional Architecture. *Cerebral Cortex,* August, Volume 26, p. 3508–3526.

Franciotti, R. & Falasca, N., 2018. The reliability of conditional Granger causality analysis in the time domain. *PeerJ Preprints,* Volume March, pp. 1-36.

Friston, K., 2011. Functional and effective connectivity: a review. *Brain Connectivity,* pp. 13-36.

Gentilucci, M. & Cattaneo, L., 2005. Automatic audiovisual integration in speech perception. *Exp. Brain Res.,* Volume 167, pp. 66-75.

Hamilton, R., Shenton, J. & Coslett, H., 2006. An acquired deficit of audiovisual speech processing. *Brain and language,* 98(1), pp. 66-73.

Haufe, S., Nikulin, V., Muller, K. & Nolte, G., 2013. A critical assessment of connectivity measures for EEG data: A simulation study. *NeuroImage,* Volume 64, pp. 120-133.

Huster, R., Plis, S. & Calhoun, V., 2015. Group-level component analyses of EEG: validation and evaluation. *Front Neurosci.,* Volume 9, pp. 1-14.

Hyvarinen, A. & Oja, E., 1997. A Fast Fixed-Point Algorithm for Independent Component Analysis. *Neural Computation,* Volume 7, pp. 1487-1492.

Ide, A. et al., 2007. Cross-correlation based methods for estimating the functional connectivity in cortical networks. *BMC Neurosci.,* 8(2), p. 63.

Irwin, J., Tornatore, L., Brancazio, L. & Whalen, D., 2011. Can children with autism spectrum disorders “hear” a speaking face?. *Child Development,* 82(5), pp. 1397-1403.

Kayser, J. & Tenke, C., 2014. Issues and considerations for using the scalp surface Laplacian in EEG/ERP. *International Journal of Psychophysiology,* Volume 97, p. 189–209.

Kirschner, A., Kam, J., Handy, T. & Ward, L., 2012. Differential synchronization in default and task-specific networks of the human brain. *Front. Hum. Neurosci.,* Volume 6, p. 139.

Kothe, C., 2013. *Simulation and Neuroscience Application Platform.* US: s.n.

Kumar, G. et al., 2016. Large Scale Functional Brain Networks Underlying Temporal Integration of Audio-Visual Speech Perception: An EEG Study. *Front. Psychol.,* Volume 7, pp. 1-13.

Lang, E. et al., 2012. Brain Connectivity Analysis: A Short Survey. *Computational Intelligence and Neuroscience,* pp. 1-21.

Leutheuser, H. et al., 2013. Comparison of the AMICA and the InfoMax algorithm for the reduction of electromyogenic artifacts in EEG data. *Conf Proc IEEE Eng Med Biol Soc. ,* Volume 2013, pp. 6804-6807.

Liu, Y. & Aviyente, S., 2012. Quantification of Effective Connectivity in the Brain Using a Measure of Directed Information. *Computational and Mathematical Methods in Medicine,* Volume 2012, pp. 1-16.

Macaluso, E. et al., 2004. Spatial and temporal factors during processing of audiovisual speech: a PET study. *NeuroImage,* Volume 21, pp. 725-732.

Maris, E. & Oostenveld, R., 2007. Nonparametric statistical testing of EEG- and MEG-data. *Journal of Neuroscience Methods,* Volume 164, pp. 177-190.

McGurk, H. & MacDonald, J., 1976. Hearing lips and seeing voices. *Nature,* Volume 264, pp. 746-748.

Nahorna, O., Berthommier, B. & Schwartz, J., 2012. Binding and unbinding in the McGurk effect. *Journal of the Acoustical Society of America,* 132(2), pp. 1061-1077.

Nath, R. & Beauchamp, M., 2012. A neural basis for interindividual differences in the McGurk effect, a multisensory speech illusion. *NeuroImage,* 59(1), pp. 781-787.

Noesselt, T. et al., 2007. Audiovisual temporal correspondence modulates human multisensory superior temporal sulcus plus primary sensory cortices. *Journal of Neuroscience,* Volume 27, pp. 11431-11441.

Olejarczyk, E., Marzetti, L., Pizzella, V. & Zappasodi, F., 2017. Comparison of connectivity analyses for resting state EEG data. *Journal of Neural Engineering,* Volume 14, pp. 1-13.

Onton, J., Westerfield, M., Townsend, J. & Makeig, S., 2006. Imaging human EEG dynamics using independent component analysis. *Neuroscience and Biobehavioral Reviews ,* Volume 30, pp. 808-822.

Oostenveld, R., Fries, P., Maris, E. & Schoffelen, J., 2011. FieldTrip: Open Source Software for Advanced Analysis of MEG, EEG, and Invasive Electrophysiological Data. *Computational Intelligence and Neuroscience,* Volume 2011, pp. 1-9.

Palmer, J. A., Kreutz-Delgado, K. & Makeig, S., 2011. AMICA: An Adaptive Mixture of Independent Component Analyzers with Shared Components. *Technical report, Swartz Center for Computational Neuroscience.*

Pearl, D. et al., 2009. Differences in audiovisual integration, as measured by McGurk phenomenon, among adult and adolescent patients with schizophrenia and age-matched healthy control groups. *Comprehensive psychiatry,* 50(2), pp. 186-192.

Pittau, F. et al., 2011. Changes preceding interictal epileptic EEG abnormalities:Comparison between EEG/fMRI and intracerebral EEG. *Epilepsia,* 52(6), pp. 1120-1129.

Poeppel, D., Idsardi, W. & van Wassenhove, V., 2008. Speech perception at the interface of neurobiology and linguistics. *Philos. Trans. R. Soc. B,* Volume 363, pp. 1071-1086.

Sharon, D. et al., 2007. The advantage of combining MEG and EEG: Comparison to fMRI in focally stimulated visual cortex. *NeuroImage,* 36(4), pp. 1225-1235.

Simon, D. & Wallace, M., 2017. Integration of Temporal Processing of Asynchronous Audiovisual Speech. *Journal of Cognitive Neuroscience,* 30(3), pp. 319-337.

Stenner, T. et al., 2018. *LabRecorder.* s.l.:s.n.

Stevenson, R. & James, T., 2009. Audiovisual integration in human superior temporal sulcus: inverse effectiveness and the neural processing of speech and object recognition. *NeuroImage,* Volume 44, pp. 1210-1223.

Stevenson, R., Zemtsov, R. & Wallace, M. T., 2012. Individual Differences in the Multisensory Temporal Binding Window Predict Susceptibility to Audiovisual Illusions. *Journal of Experimental Psychology,* 38(6), pp. 1517-1529.

Stone, J., 2002. Independent component analysis: an introduction. *Trends in Congnitive Sciences,* 6(2), pp. 59-64.

Tiippana, K., 2014. What is the McGurk effect?. *Frontiers in Psycology,* 5(1), pp. 1-3.

Tong, L., 1991. Indeterminacy and identifiability of blind indentification. *IEEE Transactions on Circuits and Systems,* Volume 38, pp. 499-509.

Ujiie, Y. et al., 2014. Autistic traits predict weaker visual influence in the McGurk effect. *Personality and Individual Differences,* Volume 60, pp. 51-52.

Van Engen, K., Xie, Z. & Chandrasekaran, B., 2017. Audiovisual sentence recognition not predicted by susceptibility to the McGurk effect. *Atten Percept Psychophys,* Volume 79, pp. 396-403.

Vincente, R., Wibral, M., Lindner, M. & Pipa, G., 2011. Transfer entropy—a model-free measure of effective connectivity for the neurosciences. *J Comput Neurosci,* Volume 30, pp. 45-67.

Wang, H. et al., 2014. A systematic framework for functional connectivity measures. *Front. Neurosci.,* Volume 8, pp. 1-22.

Woynaroski, T. et al., 2013. Multisensory Speech Perception in Children with Autism Spectrum Disorders. *Journal of Autism and Developmental Disorders,* Volume 1, pp. 1-12.

Wyczesany, M., Ligeza, T. & Grzybowski, S., 2015. Effective connectivity during visual processing is affected by emotional state. *Brain Imaging and Behavior,* Volume 9, p. 717–728.

Youssofzadeh, V. et al., 2012. Comparison of Independent Component Analysis algorithms for EEG-fMRI data fusion. *International Conference on Intelligent and Advanced Systems,* Volume 4th.

Zalesky, A., Fornito, A. & Bullmore, E., 2010. Network-based statistic: identifying differences in brain networks. *NeuroImage,* 53(4), pp. 197-207.

11 APPENDIX A – EXTRA FIGURES

11.1 NUMBER OF STIMULUS

11.1.1 ABBA

Subjects	Low-noise Correct	Medium-noise Correct	High-noise Correct	Low-noise Incorrect	Medium-noise Incorrect	High-noise Incorrect
Subject01	66	70	58	15	13	23
Subject02	74	79	75	6	1	7
Subject03	50	47	39	30	35	45
Subject04	118	119	120	1	1	3
Subject05	68	61	59	12	19	22
Subject06	57	46	52	14	26	22
Subject07	63	59	51	22	24	32
Subject08	58	52	43	24	33	38
Subject09	53	54	44	20	18	28
Subject10	67	63	49	20	17	28
Subject11	77	66	58	6	15	23
Subject12	70	63	61	10	17	19
Subject13	72	71	57	8	9	23
Subject14	60	65	62	21	18	22
Subject15	78	71	60	5	9	20
Subject16	62	61	56	18	19	22
Subject17	59	53	58	20	25	21
Subject18	111	112	96	2	0	4
Subject19	93	77	49	5	11	34
Subject20	75	63	64	5	15	14
Subject21	97	89	73	8	10	24

Figure 34: ABBA Stimulus Presented

11.1.2 ABBA – Audio only

Subjects	Low-noise Correct	Medium-noise Correct	High-noise Correct	Low-noise Incorrect	Medium-noise Incorrect	High-noise Incorrect
Subject01	26	28	19	14	12	21
Subject02	34	39	39	6	1	1
Subject03	11	9	6	29	31	34
Subject04	39	39	38	1	1	2
Subject05	28	21	23	12	19	17
Subject06	22	11	16	14	25	20
Subject07	19	19	14	21	21	26
Subject08	22	13	12	18	27	28
Subject09	17	18	10	19	18	26
Subject10	18	23	16	20	15	23

Subject11	34	26	17	6	14	23
Subject12	30	24	22	10	16	18
Subject13	32	31	21	8	9	19
Subject14	21	22	21	19	18	19
Subject15	35	32	25	5	8	15
Subject16	23	20	17	16	18	19
Subject17	20	15	20	20	24	20
Subject18	36	38	38	1	0	0
Subject19	36	30	15	4	10	25
Subject20	33	24	26	5	13	13
Subject21	35	32	21	5	8	19

Figure 35: ABBA Audio only Stimulus Presented

11.1.3 AGGA

Subjects	Low-noise Correct	Medium-noise Correct	High-noise Correct	Low-noise Incorrect	Medium-noise Incorrect	High-noise Incorrect
Subject01	79	78	79	2	2	2
Subject02	80	80	81	0	0	1
Subject03	79	77	74	1	3	7
Subject04	83	81	81	0	0	1
Subject05	81	81	78	0	0	2
Subject06	68	71	68	3	2	5
Subject07	80	80	78	1	0	3
Subject08	81	79	83	2	2	3
Subject09	72	72	72	0	1	0
Subject10	75	71	77	7	9	6
Subject11	80	79	79	0	1	1
Subject12	81	81	80	0	0	1
Subject13	80	80	79	0	0	2
Subject14	82	83	80	0	0	1
Subject15	80	81	81	0	0	1
Subject16	77	80	79	1	0	0
Subject17	79	79	80	0	0	0
Subject18	80	79	90	1	0	0
Subject19	82	79	77	0	1	4
Subject20	94	103	113	1	0	0
Subject21	79	75	77	2	6	7

Figure 36: AGGA Stimulus Presented

11.1.4 AGGA – Audio only

Subjects	Low-noise Correct	Medium-noise Correct	High-noise Correct	Low-noise Incorrect	Medium-noise Incorrect	High-noise Incorrect
Subject01	40	39	40	0	1	0
Subject02	40	40	40	0	0	0

Subject03	40	38	36	0	2	4
Subject04	40	40	40	0	0	0
Subject05	40	40	38	0	0	2
Subject06	35	36	34	1	0	2
Subject07	40	40	39	0	0	1
Subject08	38	38	39	2	2	1
Subject09	36	35	36	0	1	0
Subject10	35	32	34	4	5	4
Subject11	40	40	40	0	0	0
Subject12	40	40	40	0	0	0
Subject13	40	40	39	0	0	1
Subject14	40	40	39	0	0	1
Subject15	40	40	40	0	0	0
Subject16	38	39	39	1	0	0
Subject17	39	39	40	0	0	0
Subject18	37	38	38	1	0	0
Subject19	40	39	38	0	1	2
Subject20	37	39	37	1	0	0
Subject21	40	38	33	0	2	7

Figure 37: AGGA Audio only Stimulus Presented

11.1.5 ATHA

Subjects	Low-noise Correct	Medium-noise Correct	High-noise Correct	Low-noise Incorrect	Medium-noise Incorrect	High-noise Incorrect
Subject01	53	52	60	0	0	1
Subject02	45	41	37	1	0	4
Subject03	71	73	79	0	0	0
Subject04	0	0	0	40	40	40
Subject05	51	59	54	1	0	0
Subject06	47	59	49	1	0	3
Subject07	61	60	65	0	1	3
Subject08	59	63	73	2	6	2
Subject09	56	52	59	0	0	0
Subject10	51	50	61	5	7	6
Subject11	45	53	60	1	1	0
Subject12	49	55	57	1	0	1
Subject13	48	50	58	0	0	1
Subject14	57	55	59	2	2	0
Subject15	42	47	45	3	1	2
Subject16	50	56	54	6	2	2
Subject17	60	63	57	0	0	1
Subject18	0	0	0	37	37	37
Subject19	28	47	67	16	5	1
Subject20	23	27	17	18	19	29
Subject21	21	36	47	23	15	12

Figure 38: ATHA Stimulus Presented

11.1.6 APPA

Subjects	Low-noise Correct	Medium-noise Correct	High-noise Correct	Low-noise Incorrect	Medium-noise Incorrect	High-noise Incorrect
Subject01	41	40	42	1	1	1
Subject02	41	40	46	0	0	0
Subject03	40	43	48	0	2	4
Subject04	39	40	37	1	1	6
Subject05	40	39	49	0	1	1
Subject06	42	40	47	0	1	1
Subject07	36	41	45	4	2	1
Subject08	41	42	40	1	1	3
Subject09	35	36	41	1	0	0
Subject10	35	36	39	6	5	2
Subject11	38	42	43	3	0	1
Subject12	40	41	42	0	1	0
Subject13	40	39	46	0	1	0
Subject14	41	37	38	1	3	5
Subject15	40	41	53	0	0	0
Subject16	42	37	43	1	3	2
Subject17	39	40	40	1	0	0
Subject18	38	38	40	3	0	1
Subject19	37	37	46	5	4	5
Subject20	36	40	39	2	1	2
Subject21	41	38	42	4	6	9

Figure 39: APPA Stimulus Presented

11.2 Response Times

11.2.1 ABBA

Subjects	Low-noise Correct	Medium-noise Correct	High-noise Correct	Low-noise Incorrect	Medium-noise Incorrect	High-noise Incorrect
Subject01	1.90	1.91	1.94	1.41	1.65	1.80
Subject02	1.84	1.88	1.81	1.75	2.09	2.00
Subject03	2.00	2.13	2.04	2.05	2.01	2.10
Subject04	1.44	1.47	1.49	1.36	1.46	1.41
Subject05	1.89	1.93	1.97	1.98	1.93	2.13
Subject06	1.80	1.71	1.80	1.67	1.81	1.76
Subject07	1.54	1.51	1.57	1.47	1.40	1.52
Subject08	2.00	2.23	2.06	2.50	2.16	2.45
Subject09	2.11	2.04	2.04	2.12	1.92	2.14
Subject10	1.68	1.68	1.79	1.89	1.75	1.87
Subject11	1.84	1.80	1.87	1.84	1.79	1.91

Subject12	1.64	1.70	1.80	2.02	2.06	2.01
Subject13	1.88	1.90	2.01	1.91	1.97	2.18
Subject14	1.89	1.83	1.81	1.67	1.73	1.67
Subject15	1.68	1.71	1.70	1.68	1.82	1.85
Subject16	1.96	1.97	2.03	1.95	1.95	2.20
Subject17	1.69	1.76	1.85	1.75	1.79	1.81
Subject18	1.99	2.02	2.05	2.08	-	1.89
Subject19	1.70	1.81	1.78	1.71	1.82	1.82
Subject20	1.70	1.70	1.86	1.97	2.25	2.19
Subject21	1.90	1.81	1.94	2.56	2.23	2.18

Figure 40: ABBA Stimulus Presented

11.2.2 ABBA – Audio only

Subjects	Low-noise Correct	Medium-noise Correct	High-noise Correct	Low-noise Incorrect	Medium-noise Incorrect	High-noise Incorrect
Subject01	2.00	1.76	2.07	1.41	1.69	1.79
Subject02	1.85	1.94	1.77	1.75	2.09	1.90
Subject03	2.17	2.04	2.05	2.04	1.97	2.00
Subject04	1.38	1.46	1.45	1.36	1.46	1.51
Subject05	1.89	1.87	1.96	1.98	1.93	2.11
Subject06	1.83	1.72	1.96	1.67	1.77	1.76
Subject07	1.65	1.46	1.59	1.45	1.43	1.51
Subject08	1.91	2.20	2.14	2.16	2.11	2.31
Subject09	2.24	2.02	2.17	2.14	1.92	2.12
Subject10	1.80	1.67	1.88	1.89	1.75	1.88
Subject11	1.88	2.00	1.88	1.84	1.79	1.91
Subject12	1.66	1.77	1.89	2.02	2.05	2.01
Subject13	1.94	1.85	2.07	1.91	1.97	2.14
Subject14	1.96	1.92	1.70	1.71	1.73	1.62
Subject15	1.67	1.75	1.73	1.68	1.78	1.89
Subject16	2.00	2.09	1.98	1.95	1.92	2.11
Subject17	1.88	1.76	1.94	1.75	1.78	1.79
Subject18	1.92	1.94	1.93	1.85	-	-
Subject19	1.68	1.78	1.82	1.74	1.81	1.74
Subject20	1.69	1.73	1.79	1.97	2.29	2.23
Subject21	1.87	1.75	1.90	2.66	2.14	2.17

Figure 41: ABBA Audio only Stimulus Presented

11.2.3 AGGA

Subjects	Low-noise Correct	Medium-noise Correct	High-noise Correct	Low-noise Incorrect	Medium-noise Incorrect	High-noise Incorrect
Subject01	1.79	1.73	1.84	3.56	3.15	0.53
Subject02	1.68	1.77	1.80	-	-	0.47
Subject03	1.82	1.80	1.88	1.97	2.35	2.44

Subject04	1.46	1.49	1.51	-	-	1.78
Subject05	1.97	1.88	1.89	-	-	3.40
Subject06	1.66	1.72	1.78	1.95	2.21	2.12
Subject07	1.44	1.47	1.45	0.73		1.45
Subject08	1.88	1.86	1.90	1.89	2.41	2.51
Subject09	2.13	2.11	2.13	-	-	-
Subject10	1.93	1.88	2.06	1.83	1.63	2.17
Subject11	1.52	1.59	1.59	-	2.56	0.14
Subject12	1.72	1.77	1.85	-	-	3.28
Subject13	1.75	1.75	1.83	-	-	2.39
Subject14	1.74	1.71	1.73	-	-	2.73
Subject15	1.66	1.68	1.68	-	-	1.63
Subject16	2.04	2.00	2.03	-	-	-
Subject17	1.59	1.62	1.65	-	-	-
Subject18	1.77	1.96	2.04	4.04		-
Subject19	1.56	1.55	1.63		2.25	2.10
Subject20	1.68	1.72	1.77	0.51	-	-
Subject21	1.82	2.01	1.89	2.82	2.43	2.89

Figure 42: AGGA Stimulus Presented

11.2.4 AGGA – Audio only

Subjects	Low-noise Correct	Medium-noise Correct	High-noise Correct	Low-noise Incorrect	Medium-noise Incorrect	High-noise Incorrect
Subject01	1.89	1.71	1.86	-	4.22	-
Subject02	1.65	1.82	1.79	-	-	-
Subject03	1.76	1.79	1.90	-	1.94	2.19
Subject04	1.45	1.48	1.50	-	-	-
Subject05	2.04	1.91	1.84	-	-	3.40
Subject06	1.67	1.68	1.88	1.96	-	2.34
Subject07	1.49	1.48	1.48	-	-	1.52
Subject08	1.82	1.81	1.95	1.89	2.41	0.89
Subject09	2.15	2.03	2.13	-	-	-
Subject10	1.85	1.99	2.08	2.07	1.19	2.24
Subject11	1.52	1.60	1.68	-	-	-
Subject12	1.68	1.79	1.87	-	-	-
Subject13	1.77	1.73	1.82	-	-	2.68
Subject14	1.78	1.73	1.75	-	-	2.73
Subject15	1.68	1.67	1.68	-	-	-
Subject16	2.07	2.04	2.06	-	-	-
Subject17	1.58	1.66	1.75	-	-	-
Subject18	1.76	2.02	2.06	4.04	-	-
Subject19	1.57	1.59	1.69	-	2.25	2.12
Subject20	1.55	1.58	1.63	0.51	-	-
Subject21	1.91	2.15	1.81	-	0.94	2.89

Figure 43: AGGA Audio only Stimulus Presented

11.2.5 ATHA

Subjects	Low-noise Correct	Medium-noise Correct	High-noise Correct	Low-noise Incorrect	Medium-noise Incorrect	High-noise Incorrect
Subject01	1.63	1.59	1.77	-	-	-0.15
Subject02	1.86	1.85	1.90	2.17	-	2.80
Subject03	1.91	1.92	1.95	-	-	
Subject04	-	-	-	1.53	1.54	1.62
Subject05	1.84	2.06	2.02	2.13	-	
Subject06	1.70	1.72	1.76	3.51	-	2.01
Subject07	1.49	1.44	1.50	-	1.67	1.52
Subject08	2.27	2.31	2.36	1.16	2.76	1.32
Subject09	2.02	1.90	1.96	-	-	-
Subject10	1.86	1.85	1.84	1.95	1.46	2.32
Subject11	1.79	1.79	1.82	0.08	0.29	-
Subject12	1.96	2.05	2.01	1.99	-	2.42
Subject13	1.83	1.93	2.03	-	-	1.65
Subject14	1.71	1.67	1.72	1.32	2.26	-
Subject15	1.75	1.75	1.70	2.17	2.54	1.62
Subject16	1.87	2.05	2.01	2.64	1.98	2.53
Subject17	1.76	1.80	1.88	-	-	1.96
Subject18	-	-	-	2.20	2.24	2.37
Subject19	1.64	1.62	1.72	1.73	2.13	1.43
Subject20	2.21	2.07	2.09	2.04	2.14	2.18
Subject21	2.19	2.20	2.10	2.17	2.16	1.94

Figure 44: ATHA Stimulus Presented

11.2.6 APPA

Subjects	Low-noise Correct	Medium-noise Correct	High-noise Correct	Low-noise Incorrect	Medium-noise Incorrect	High-noise Incorrect
Subject01	1.90	1.76	2.03	2.22	2.26	3.77
Subject02	1.82	1.87	1.86	-	-	-
Subject03	2.17	2.17	2.10	-	1.92	2.29
Subject04	1.45	1.49	1.42	1.62	1.87	1.51
Subject05	2.20	2.01	2.30	-	0.70	1.50
Subject06	1.72	1.68	1.73	-	1.87	2.24
Subject07	1.67	1.56	1.60	1.77	1.65	2.00
Subject08	2.09	2.11	1.91	2.71	3.74	1.35
Subject09	2.03	2.17	2.07	2.06	-	-
Subject10	1.86	1.88	1.82	2.02	1.99	1.63
Subject11	1.77	2.06	1.91	1.76	-	1.40
Subject12	1.92	1.92	1.86	-	2.34	-
Subject13	1.91	1.91	1.90	-	2.88	-
Subject14	2.03	1.94	2.00	2.03	2.26	1.86

Subject15	1.76	1.68	1.76	-	-	-
Subject16	2.09	2.41	2.28	0.74	2.01	2.70
Subject17	1.81	1.83	1.83	1.92	-	-
Subject18	2.03	1.94	1.89	2.17	-	2.27
Subject19	1.64	1.63	1.74	1.63	1.72	1.88
Subject20	1.93	1.89	2.00	0.87	1.60	2.18
Subject21	1.99	1.94	1.92	2.65	1.60	2.00

Figure 45: APPA Stimulus Presented

11.3 Complete Set of Conditional Granger Causality Pictures

For complete set of pictures of all connectivity see attached zip under 'ConnectivityPictures'.

12 Appendix B – Code

For a full set of code see attached zip file 'code'.

12.1 Connectivity Script in Colossus

The connectivity script that was run in colossus is as seen below.

```
function Connectivity_run( task, CONN_method, recalculate_connectivity,
freqrange),


% first set up the paths
homepath = '/home/wout0004';
matlabrootpath = fullfile( homepath, 'matlab');
addpath( fullfile( homepath, 'Scripts-McGurk'));
addpath( genpath( fullfile( homepath, 'matlab')));

% load subject IDs
subjects = { 'Subject01', 'Subject02', 'Subject03', 'Subject05',
'Subject06', ...
    'Subject07', 'Subject08', 'Subject09', 'Subject10', ...
    'Subject11', 'Subject12', 'Subject13', 'Subject14', 'Subject15', ...
    'Subject16', 'Subject17', 'Subject18', 'Subject19', 'Subject20',
'Subject21'};
% - later
Ns = numel( subjects);

% work out iteration matrix
sounds = { 'ABBA', 'AGGA', 'ATHA', 'APPA', 'ABBA-AO', 'AGGA-AO'};
noises = { 'LN', 'MN', 'HN'};
conditions = {'Truth-only' 'Perceived-only' 'Truth+Perceived'};
tasks = { 'McGurk-Audio-degradation'};
Nso = numel(sounds);
Nno = numel( noises);
Npp = numel(conditions);

% what time did we start?
fprintf( 'Time started: %s\n', datestr( now));

% default values
```

```
if nargin < 2, recalculate_connectivity = false; end
if isequal( recalculate_connectivity, 'false'), recalculate_connectivity =
false; end
if isequal( recalculate_connectivity, 'true'), recalculate_connectivity =
true; end
if nargin < 3
    fre = [4 45];
else
    switch (freqrange)
        case 'delta'
            fre = [0.5 4];
        case 'theta'
            fre = [4 8];
        case 'alpha'
            fre = [9 13];
        case 'beta'
            fre = [14 30];
        case 'gamma'
            fre = [30 45];
        case 'NA'
            fre = [];
        otherwise
            fprintf( 'Invalid frequency range');
            exit
    end
end
%if nargin < 1,
%    tasks = '';
%elseif ~iscell( tasks),
%    tasks = { tasks};
%end

%% trying to reduce the writing
%if strcmpi( tasks, 'test'),
%    tasks = { 'McGurk-Audio-degredation', 'McGurk-EO-start', ...
%        'McGurk-EC-start', 'McGurk-EC-end'};
%end
%Nt = numel( tasks);

ME = 1;
fail_count = 0;
fprintf( '|| ');
while ~isempty( ME),
    try
        setpref( 'eeg3', 'verbose', 0);
        ME = [];
    catch ME;
        fail_count = fail_count + 1;
    end
    fprintf( '%d ', fail_count);
    if fail_count==1000, break; end
    if fail_count>0 && rem( fail_count, 100)==0 && ~isempty( ME),
        pause_time = randi( 100, 1);
        fprintf( '\npausing for %d seconds\n', pause_time);
        pause( pause_time);
    end
end
fprintf( '||');
fprintf( '\nFail count = %d\n', fail_count);
```

```
if strcmp( task, 'McGurk-Audio-degradation'),
    IM = zeros( Ns*Nso*Nno*Npp, 2);
    count = 1;
    for ss = 1:Ns
        for ii = 1:Nso*Nno*Npp
            IM( count, :) = [ ss, ii];
            count = count + 1;
        end
    end
else
    IM = zeros( Ns, 2);
    for ss = 1:Ns
        IM( ss, :) = [ ss, 1];
    end
end

% get iteration ID
[ ~, task_id] = system( 'printenv SGE_TASK_ID');
iter_ind = sscanf( task_id, '%d');

% determine subject ID
subject = subjects{ IM( iter_ind, 1)};

%%% work out which subject this process is working on
%[ ~, task_id] = system( 'printenv SGE_TASK_ID');
%subject_ind = sscanf( task_id, '%d');
%if isempty( subject_ind),
%    subject = 'Subject01';
%else
%    subject = subjects{ subject_ind}; %#ok<USENS>
%end

% identify folder where the data will reside
input_folderpath = fullfile( homepath, 'Data-McGurk', subject, task);
if strcmp(CONN_method, 'CGC_freq')
    output_folderpath = fullfile( homepath, 'Data-McGurk', 'Connectivity',
[CONN_method ' Connectivity'], freqrange, subject, task);
else
    output_folderpath = fullfile( homepath, 'Data-McGurk', 'Connectivity',
[CONN_method ' Connectivity'], subject, task);
end

% make output folder if it doesnt exist
if ~exist( output_folderpath, 'dir')
    mkdir( output_folderpath);
end

% calculate connectivity
conn_files = filesearch( sprintf( '%s_AMICA_forCONN', task),
input_folderpath);
Ncf = numel( conn_files);

% determine input and output files
if IM( iter_ind, 2) > Ncf, disp( 'iteration index is beyond number of
connectivity files'); return; end
input_fn = conn_files( IM( iter_ind, 2)).name;
```

```
splits = strsplit( input_fn, filesep);
input_fn = splits{ end};
output_fn = sprintf( '%s_%s_%s.mat', input_fn( 1:strfind( input_fn,
'_forCONN')-1), ...
    CONN_method, input_fn( strfind( input_fn, '_forCONN')+9:strfind(
input_fn, '.eeg3-eeg')-1));
input_fp = fullfile( input_folderpath, input_fn);
output_fp = fullfile( output_folderpath, output_fn);

% determine time conditions
clear times
if strcmp( task, 'McGurk-Audio-degradation'),
    times( :, 1) = -1:0.1:1;
    times( :, 2) = 0:0.1:2;
else
    times = [ 0, 5.9];
end
Nti = size( times, 1);

wait1 = tic;
%for cf = 1:Ncf
%   input_fn = conn_files( cf).name;
%   output_fn = sprintf( '%s_%s_%s.mat', input_fn( 1:strfind( input_fn,
'_forCONN')-1), ...
%           CONN_method, input_fn( strfind( input_fn,
'_forCONN')+9:strfind( input_fn, '.eeg3-eeg')-1));

disp( '------------------------------------------------------');
fprintf( 'Connectivity method: %s\n', CONN_method);
fprintf( 'Subject: %s\n', subject);
fprintf( 'Task: %s\n', task);
fprintf( '%s, %d of %d\n', input_fp, IM( iter_ind, 2), Ncf);
disp( '------------------------------------------------------');

% check for pre-existing input data
if ~exist( input_fp, 'file')
    disp( 'file does not exist, skipping ..');
    return
end

% check for pre-existing output data
if exist( output_fp, 'file') && ~recalculate_connectivity
    disp( 'file pre-exists, skipping ..');
    return
end

% load data
try;
    ep = eeg3.eeg.load( input_fp);
catch;
    error( 'data cannot be loaded, likely corrupt: %s', input_fp);
end

% channel info
chans = ep( 1).chan.getlabels;
Nchan = numel( chans);

% allocate room for AM
```

```
AM = zeros( Nchan, Nchan, Nti, numel( ep));

% load temporary file
temp_file = sprintf( '%s_temp.mat', output_fp( 1:strfind( output_fp,
'.mat')-1));
if exist( temp_file, 'file'),
    try
        load( temp_file);
    catch ME;
        disp( ME);
        delete( temp_file);
        clear ME;
    end
end

% calculate connectivity
switch CONN_method
    case 'dPLV'

        for ti = 1:Nti

            % skip time period if completed
            if ~all( all( AM( :, :, ti, 1)==0)),
                fprintf( 'time period %d of %d pre-completed\n', ti, Nti);
                continue
            end

            disp( '-------------------------------------------------------');
            fprintf( 'Connectivity method: dPLV\n');
            fprintf( 'Subject: %s\n', subject);
            fprintf( 'Task: %s\n', task);
            fprintf( 'Epoch element: %d of %d\n', IM( iter_ind, 2), Ncf);
            fprintf( 'Time period: %d of %d\n', ti, Nti);
            disp( '-------------------------------------------------------');

            wait2 = tic;
            AM( :, :, ti, :) = directionalphaselockingvalue( ...
                ep.selecttime( times( ti, 1), times( ti, 2)));
            how_much_longer( toc( wait2), [ti Nti]);

            % save temporary file
            save( temp_file, 'AM');
        end

    case 'TE'
        min_lag = 1;
        step = 1;
        max_lag = 20;

        for ti = 1:Nti

            % skip time period if completed
            if ~all( all( AM( :, :, ti, 1)==0)),
                fprintf( 'time period %d of %d pre-completed\n', ti, Nti);
                continue
            end

            disp( '-------------------------------------------------------');
```

```
            fprintf( 'Connectivity method: TE\n');
            fprintf( 'Subject: %s\n', subject);
            fprintf( 'Task: %s\n', task);
            fprintf( 'Epoch element: %d of %d\n', IM( iter_ind, 2), Ncf);
            fprintf( 'Time period: %d of %d\n', ti, Nti);
            disp( '-------------------------------------------------------');

            wait2 = tic;
            AM( :, :, ti, :) = transferentropy( ...
                ep.selecttime( times( ti, 1), times( ti, 2)), ...
                'min_lag', round( ep( 1).time.samplerate*(min_lag/1000)),
...
                'lag_step', round( ep( 1).time.samplerate*(step/1000)), ...
                'max_lag', round( ep( 1).time.samplerate*(max_lag/1000)));
            how_much_longer( toc( wait2), [ti Nti]);

            % save temporary file
            save( temp_file, 'AM');
        end

    case 'CGC'

        for ti = 1:Nti

            % skip time period if completed
            if ~all( all( AM( :, :, ti, 1)==0)),
                fprintf( 'time period %d of %d pre-completed\n', ti, Nti);
                continue
            end

            disp( '-------------------------------------------------------');
            fprintf( 'Connectivity method: CGC\n');
            fprintf( 'Subject: %s\n', subject);
            fprintf( 'Task: %s\n', task);
            fprintf( 'Epoch element: %d of %d\n', IM( iter_ind, 2), Ncf);
            fprintf( 'Time period: %d of %d\n', ti, Nti);
            disp( '-------------------------------------------------------');

            wait2 = tic;
            AM( :, :, ti, :) = C_GRanger_t( ...
                ep.selecttime( times( ti, 1), times( ti, 2)));
            how_much_longer( toc( wait2), [ti Nti]);

            % save temporary file
            save( temp_file, 'AM');
        end

    case 'CGC_freq'

        for ti = 1:Nti

            % skip time period if completed
            if ~all( all( AM( :, :, ti, 1)==0)),
                fprintf( 'time period %d of %d pre-completed\n', ti, Nti);
                continue
            end

            disp( '-------------------------------------------------------');
```

```
            fprintf( 'Connectivity method: CGC_freq\n');
            fprintf( 'Subject: %s\n', subject);
            fprintf( 'Task: %s\n', task);
            fprintf( 'Epoch element: %d of %d\n', IM( iter_ind, 2), Ncf);
            fprintf( 'Time period: %d of %d\n', ti, Nti);
            disp( '------------------------------------------------------');

            wait2 = tic;
            AM( :, :, ti, :) = CGC_s_test( ...
                ep.selecttime( times( ti, 1), times( ti, 2)),...
                'mean', 'Y', 'fre', fre/ep( 1).time.samplerate, 'fres', ep(
1).time.samplerate/2);
            how_much_longer( toc( wait2), [ti Nti]);

            % save temporary file
            save( temp_file, 'AM');
        end
end

delete( temp_file);
save( output_fp, 'AM', 'chans', '-v7.3');
%end

fprintf( 'Connectivity calculation took %0.2f minutes to complete\n', toc(
wait1)/60);
```

12.2 Plot Results

This script was used to plot different results.

```
%% switches
warning( 'off', 'all');
clear all
new_file = false;

%% preamble
sharedrootpath = getpref( 'tsg', 'sharedrootpath');
homepath = fullfile( sharedrootpath, 'Projects\AVSP\McGurk');
matlabrootpath = getpref( 'tsg', 'matlabrootpath');
addpath( genpath( 'V:\EEG\People\Caitlin'));
eeglabpath = fullfile( matlabrootpath, 'Matlab', 'eeglab');
hdmfile = fullfile(eeglabpath,'plugins','dipfit2.3', ...
    'standard_BEM','standard_vol.mat');

%% define variables
subjects = { 'Subject01', 'Subject02', 'Subject03', 'Subject05',
'Subject06', ...
     'Subject07', 'Subject08', 'Subject09', 'Subject10', ...
     'Subject11', 'Subject12', 'Subject13', 'Subject14', 'Subject15', ...
    'Subject16', 'Subject17', 'Subject18', 'Subject19', 'Subject20',
'Subject21'};
tasks = { 'McGurk-Audio-degradation'};
ICA = 'AMICA';
CONN = 'CGC';
sounds = { 'ABBA', 'ABBA-AO', 'APPA', 'ATHA', 'AGGA', 'AGGA-AO'};
noises = { 'LN', 'MN', 'HN'};
```

```
perception = { 'Truth+Perceived', 'Perceived-only', 'Truth-only'};
bands = {'theta', 'alpha', 'beta', 'gamma'};

%% Number of ...
Ns = numel( subjects);
Nt = numel( tasks);
Nso = numel( sounds);
Nno = numel( noises);
Np = numel( perception);
Nba = numel(bands);

%% load data
wait1 = tic;
for t = 1:Nt
    task = tasks{ t};

    switch task
        case 'McGurk-Audio-degradation'
            AMs = cell( Nso, Nno, Np, Ns);


            % time definition
            times( :, 1) = -1:0.1:0.8;
            times( :, 2) = -0.8:0.1:1;


            for s = 1:Ns
                subject = subjects{ s};

                % define folderpath
                folderpath = fullfile( homepath, 'data', subject, task);

                % load epoch file
                d = eeg3.eeg.load( fullfile( folderpath, sprintf(
'%s_%s_epoched_IC.eeg3-eeg', ...
                    task, ICA)));
                chaninfo{ s} = d{ 1}( 1).chan;
%                 clear d

                % load connectivity file
                temp = load( fullfile( folderpath, sprintf( '%s_%s_%s.mat',
...
                    task, ICA, CONN)));
                AMs( :, :, :, s) = temp.AMs;
                chans{ s} = temp.chans;
                clear temp

                disp( '------------------------------------');
                fprintf( 'Subject: %d of %d\n', s, Ns);
                fprintf( 'Task: %s, %d of %d\n', task, t, Nt);
                disp( 'Data loading completed');
                disp( '------------------------------------');

            end

            % intersect channels
            channels = chans{ 1};
```

```
            chanlabels = channels;
            for s = 2:numel( chans)
                chanlabels = intersect(chans{ s}, chanlabels);
            end

            % index into AMs
            for s = 1:size( AMs, 4)
                ind = find( ismember( chans{ s}, chanlabels));
                AM = AMs( :, :, :, s);
                for ep = 1:numel( AM),
                    if isempty( AM{ ep}), continue; end
                    AM{ ep} = AM{ ep}( ind, ind, :, :);
                end
                AMs( :, :, :, s) = AM;
                clear AM
            end

            % number of channels
            Nchan = numel( chanlabels);


    end
end

%% calculate baseline
poststimtimeframes = 5;
baselines = cell(Nso,Nno,Np);
poststim = cell(Nso,Nno,Np,poststimtimeframes);

for k = 1:Nso
    for kk = 1:Nno
        for kkk = 1:Np
            if ~isempty(AMs(k,kk,kkk,~cellfun( @isempty, AMs(k,kk,kkk,:))))

                clear temp
                clear temp2

                temp = AMs(k,kk,kkk,~cellfun( @isempty, AMs(k,kk,kkk,:)));
                baseline = cellfun( @(x) mean(mean( x( :, :, 2:5, :),
3),4), temp, ...
                    'uniformoutput', false);
                baselineaverage =
zeros(length(baseline{:,1,1,1}),length(baseline{1,:,1,1}),length(baseline))
;

                for i = 1:length(baseline)
                    baselineaverage(:,:,i) = baseline{:,:,1,i};
                end

                baselines{k,kk,kkk} = mean(baselineaverage,3);

                for frame = 1:poststimtimeframes
                    clear tempposts timframe;
                    clear tempposts timframeaverage;
                    tempposts timframe = cellfun( @(x) mean( x( :, :,
10+frame, :), 4), temp, ...
                        'uniformoutput', false);
```

```
                    temppoststimframeaverage =
zeros(length(temppoststimframe{:,1,1,1}),length(temppoststimframe{1,:,1,1})
,length(temppoststimframe));

                    for i = 1:length(temppoststimframe)
                        temppoststimframeaverage(:,:,i) =
temppoststimframe{:,:,1,i};
                    end

                    poststim{k,kk,kkk,frame} =
mean(temppoststimframeaverage,3);

                end
            end
        end
    end
end

%% visual plot of the connectivity matricies with colours
%

imagesc(baselineaverage);
title('Baseline')
colorbar
caxis([0 0.3])
figure
imagesc(mean(poststimframe1average,3));
colorbar
caxis([0 0.3])
figure
imagesc(mean(poststimframe2average,3));
colorbar
caxis([0 0.3])
figure
imagesc(mean(poststimframe3average,3));
colorbar
caxis([0 0.3])

imagesc(mean(poststimframe1average,3)-mean(baselineaverage,3));
colorbar
figure
imagesc(mean(poststimframe2average,3)-mean(baselineaverage,3));
colorbar
figure
imagesc(mean(poststimframe3average,3)-mean(baselineaverage,3));
colorbar



%%
% comparison file
comp_fn = 'V:\EEG\Projects\AVSP\McGurk\Data\Subject01\McGurk-Audio-
degradation\McGurk-Audio-degradation_AMICA_epoched_IC.eeg3-eeg';
ep = eeg3.eeg.load( comp_fn);


%% get difference from baseline
difference = cell(Nso,Nno,Np,poststimtimeframes);
```

```
for k = 1:Nso
    for kk = 1:Nno
        for kkk = 1:Np
            if ~isempty(baselines{k,kk,kkk})
                for frame = 1:length(poststim(1,1,1,:))
                    difference{k,kk,kkk,frame} = poststim{k,kk,kkk,frame}-
baselines{k,kk,kkk};
                end
            end
        end
    end
end


%% plots

maxvals = cellfun( @(x) max( x(:)), difference, ...
    'uniformoutput', false);
limitmax = maxvals;
absolutemax = max([limitmax{:}]);
minvals = cellfun( @(x) min( x(:)), difference, ...
    'uniformoutput', false);
limitmin = minvals;
absolutemin = min([limitmin{:}]);
if (abs(absolutemin)>absolutemax)
    absolutemax = abs(absolutemin);
else
    absolutemin = -absolutemax;
end

ind = find( ismember(chans{1}, chanlabels));

percentshow = 0.35;

chanlocarray(:,:,1) = [[ep{1}(1).chan(ind).x]',[ep{1}(1).chan(ind).x]'];
chanlocarray(:,:,2) = [[ep{1}(1).chan(ind).y]',[ep{1}(1).chan(ind).y]'];
chanlocarray(:,:,3) = [[ep{1}(1).chan(ind).z]',[ep{1}(1).chan(ind).z]'];

poststimtimeframes = 5;
load('backview.mat');
load('sideview.mat');



for k = 1:Nso
    for kk = 1:Nno
        for kkk = 1:Np
            if ~isempty(ep{k,kk,kkk})
                stimulus = ep{k,kk,kkk}(1).label;
                stimtitle = stimulus;
                stimulus = strrep(stimulus,' ','_');
                im = [];
                %             writerObj = VideoWriter(['stimulus_' stimulus
'.avi']);
                %             writerObj.FrameRate = 0.2;
                for frame = 1:poststimtimeframes %change back
                    if ~isempty(difference{k,kk,kkk,frame})
```

```
                        percentshow = 0;
                        mat2plot =
((difference{k,kk,kkk,frame}>(limitmax{k,kk,kkk,frame}*percentshow)) +
(difference{k,kk,kkk,frame}<(limitmin{k,kk,kkk,frame}*percentshow))).*diffe
rence{k,kk,kkk,frame};
                        mat2plot(isnan(mat2plot)) = 0;
                        limitloop = 0;
                        while (sum(~(mat2plot(:) == 0))>=8 && limitloop ==
0)
                            percentshow = percentshow + 0.000001;
                            mat2plot =
((difference{k,kk,kkk,frame}>(limitmax{k,kk,kkk,frame}*percentshow)) +
(difference{k,kk,kkk,frame}<(limitmin{k,kk,kkk,frame}*percentshow))).*diffe
rence{k,kk,kkk,frame};
                            mat2plot(isnan(mat2plot)) = 0;
                            if percentshow >= 1
                                limitloop = 1;
                            end
                        end
                        plot_connections3d_regions(mat2plot, chanlabels,
'', 'xyz', chanlocarray, 'method', 'arrow', 'conlims', [absolutemin
absolutemax]);
                        title({['Timeframe ' num2str(frame)],'Stimulus:
',stimtitle})
                        saveas(gcf,['stimulus_' stimulus '_timeframe_'
num2str(frame) '_top.jpg']);
                        if isempty(im)
                            f = getframe(gcf);
                            [im,map] = rgb2ind(f.cdata,256);
                            %                           open(writerObj);
                            %                           writeVideo(writerObj,
f);
                            imwrite(im,map,['stimulus_' stimulus
'_top.gif'],'gif','LoopCount',inf,'DelayTime',1);
                            im = [];
                        else
                            f = getframe(gcf);
                            ifr = frame2im(f);
                            %                           writeVideo(writerObj,
f);
                            [im,map] = rgb2ind(f.cdata,map);
                            imwrite(im,map,['stimulus_' stimulus
'_top.gif'],'gif','DisposalMethod', 'leaveInPlace',
'WriteMode','append','DelayTime',1);
                        end

                        view(backviewaz, backviewel)
                        title({['Timeframe ' num2str(frame)],'Stimulus:
',stimtitle})
                        saveas(gcf,['stimulus_' stimulus '_timeframe_'
num2str(frame) '_back.jpg']);
                        if isempty(im)
                            f = getframe(gcf);
                            [im,map] = rgb2ind(f.cdata,256);
                            %                           open(writerObj);
                            %                           writeVideo(writerObj,
f);
                            imwrite(im,map,['stimulus_' stimulus
'_back.gif'],'gif','LoopCount',inf,'DelayTime',1);
                            im = [];
```

```
                        else
                            f = getframe(gcf);
                            ifr = frame2im(f);
                            %                           writeVideo(writerObj,
f);
                            [im,map] = rgb2ind(f.cdata,map);
                            imwrite(im,map,['stimulus_' stimulus
'_back.gif'],'gif','DisposalMethod', 'leaveInPlace',
'WriteMode','append','DelayTime',1);
                        end

                        view(sideviewaz, sideviewel)
                        saveas(gcf,['stimulus_' stimulus '_timeframe_'
num2str(frame) '_side.jpg']);
                        if isempty(im)
                            f = getframe(gcf);
                            [im,map] = rgb2ind(f.cdata,256);
                            %                           open(writerObj);
                            %                           writeVideo(writerObj,
f);
                            imwrite(im,map,['stimulus_' stimulus
'_side.gif'],'gif','LoopCount',inf,'DelayTime',1);
                        else
                            f = getframe(gcf);
                            ifr = frame2im(f);
                            %                           writeVideo(writerObj,
f);
                            [im,map] = rgb2ind(f.cdata,map);
                            imwrite(im,map,['stimulus_' stimulus
'_side.gif'],'gif','DisposalMethod', 'leaveInPlace',
'WriteMode','append','DelayTime',1);
                        end
                        close all

                    end
                    %                   imwrite(im,map,['stimulus_' stimulus
'.gif'], 'gif', 'DelayTime', 5, 'LoopCount', inf);

                end
                %             close(writerObj);
                %             clear im
                close all
            end
        end
    end
end

%% plot using con_mats from statistics

maxvals = cellfun( @(x) max( x(:)), difference, ...
    'uniformoutput', false);
limitmax = maxvals*1.05;
absolutemax = max([limitmax{:}]);
minvals = cellfun( @(x) min( x(:)), difference, ...
    'uniformoutput', false);
limitmin = minvals*1.5;
absolutemin = min([limitmin{:}]);
if (abs(absolutemin)>absolutemax)
    absolutemax = abs(absolutemin);
```

```
else
    absolutemin = -absolutemax;
end

ind = find( ismember( chans{1}, chanlabels));

percentshow = 0.35;

chanlocarray(:,:,1) = [[ep{1}(1).chan(ind).x]',[ep{1}(1).chan(ind).x]'];
chanlocarray(:,:,2) = [[ep{1}(1).chan(ind).y]',[ep{1}(1).chan(ind).y]'];
chanlocarray(:,:,3) = [[ep{1}(1).chan(ind).z]',[ep{1}(1).chan(ind).z]'];

poststimtimeframes = 5;
load('backview.mat');
load('sideview.mat');


for k = 1:Nso
    for kk = 1:Nno
        for kkk = 1:Np
            if ~isempty(ep{k,kk,kkk})
                stimulus = ep{k,kk,kkk}(1).label;
                stimtitle = stimulus;
                stimulus = strrep(stimulus,' ','_');
                im = [];
                %             writerObj = VideoWriter(['stimulus_' stimulus
'.avi']);
                %             writerObj.FrameRate = 0.2;
                for frame = 1:poststimtimeframes %change back
                    if ~isempty(difference{k,kk,kkk,frame})
                        percentshow = 0;
                        mat2plot = (dec_con_mats{k,kk,kkk,frame}{1} +
inc_con_mats{k,kk,kkk,frame}{1}).*difference{k,kk,kkk,frame};
                        plot_connections3d(mat2plot, chanlabels, '', 'xyz',
chanlocarray, 'method', 'arrow', 'conlims', [absolutemin absolutemax]);
                        title({['Timeframe ' num2str(frame)],'Stimulus:
',stimtitle})
                        saveas(gcf,['stimulus_' stimulus '_timeframe_'
num2str(frame) '_top.jpg']);
                        if isempty(im)
                            f = getframe(gcf);
                            [im,map] = rgb2ind(f.cdata,256);
                            %                         open(writerObj);
                            %                         writeVideo(writerObj,
f);
                            imwrite(im,map,['stimulus_' stimulus
'_top.gif'],'gif','LoopCount',inf,'DelayTime',1);
                            im = [];
                        else
                            f = getframe(gcf);
                            ifr = frame2im(f);
                            %                         writeVideo(writerObj,
f);
                            [im,map] = rgb2ind(f.cdata,map);
                            imwrite(im,map,['stimulus_' stimulus
'_top.gif'],'gif','DisposalMethod', 'leaveInPlace',
'WriteMode','append','DelayTime',1);
                        end
```

```
                        view(backviewaz, backviewel)
                        title({['Timeframe ' num2str(frame)],'Stimulus:
',stimtitle})
                        saveas(gcf,['stimulus_' stimulus '_timeframe_'
num2str(frame) '_back.jpg']);
                        if isempty(im)
                            f = getframe(gcf);
                            [im,map] = rgb2ind(f.cdata,256);
                            %                           open(writerObj);
                            %                           writeVideo(writerObj,
f);
                            imwrite(im,map,['stimulus_' stimulus
'_back.gif'],'gif','LoopCount',inf,'DelayTime',1);
                            im = [];
                        else
                            f = getframe(gcf);
                            ifr = frame2im(f);
                            %                           writeVideo(writerObj,
f);
                            [im,map] = rgb2ind(f.cdata,map);
                            imwrite(im,map,['stimulus_' stimulus
'_back.gif'],'gif','DisposalMethod', 'leaveInPlace',
'WriteMode','append','DelayTime',1);
                        end

                        view(sideviewaz, sideviewel)
                        saveas(gcf,['stimulus_' stimulus '_timeframe_'
num2str(frame) '_side.jpg']);
                        if isempty(im)
                            f = getframe(gcf);
                            [im,map] = rgb2ind(f.cdata,256);
                            %                           open(writerObj);
                            %                           writeVideo(writerObj,
f);
                            imwrite(im,map,['stimulus_' stimulus
'_side.gif'],'gif','LoopCount',inf,'DelayTime',1);
                        else
                            f = getframe(gcf);
                            ifr = frame2im(f);
                            %                           writeVideo(writerObj,
f);
                            [im,map] = rgb2ind(f.cdata,map);
                            imwrite(im,map,['stimulus_' stimulus
'_side.gif'],'gif','DisposalMethod', 'leaveInPlace',
'WriteMode','append','DelayTime',1);
                        end


                    end
                    %                   imwrite(im,map,['stimulus_' stimulus
'.gif'], 'gif', 'DelayTime', 5, 'LoopCount', inf);

                end
                %             close(writerObj);
                %             clear im
                close all
            end
        end
    end
```

```
end



%% noise results

lownoiseresults = zeros([size(difference{1,1,1,1}),
length(difference(1,1,1,:))]);
mednoiseresults = zeros([size(difference{1,1,1,1}),
length(difference(1,1,1,:))]);
highnoiseresults = zeros([size(difference{1,1,1,1}),
length(difference(1,1,1,:))]);


for frame = 1:poststimtimeframes %change back
    clear temp
    clear temp2
    empty = ~cellfun( @isempty, difference(:,1,:,frame));
    temp = difference(:,1,:,frame);
    temp = temp(empty);
    for i = 1:length(temp)
        temp2(:,:,i) = temp{i};
    end
    lownoiseresults(:,:,frame) = mean(temp2,3);
    clear temp
    clear temp2
    empty = ~cellfun( @isempty, difference(:,2,:,frame));
    temp = difference(:,2,:,frame);
    temp = temp(empty);
    for i = 1:length(temp)
        temp2(:,:,i) = temp{i};
    end
    mednoiseresults(:,:,frame) = mean(temp2,3);
    clear temp
    clear temp2
    empty = ~cellfun( @isempty, difference(:,3,:,frame));
    temp = difference(:,3,:,frame);
    temp = temp(empty);
    for i = 1:length(temp)
        temp2(:,:,i) = temp{i};
    end
    highnoiseresults(:,:,frame) = mean(temp2,3);
end

maxlow = max(lownoiseresults(:));
maxmed = max(mednoiseresults(:));
maxhigh = max(highnoiseresults(:));
minlow = min(lownoiseresults(:));
minmed = min(mednoiseresults(:));
minhigh = min(highnoiseresults(:));

maxoverall = max([maxlow maxmed maxhigh]);
minoverall = min([minlow minmed minhigh]);
if maxoverall>abs(minoverall)
    minoverall = -maxoverall;
else
    maxoverall = abs(minoverall);
end
```

```
shownum = 8;
im = [];
for frame = 1:poststimtimeframes %change back
    percentshow = 0;
    limitloop = 0;
    mat2plotlow = (((lownoiseresults(:,:,frame)>(maxlow*percentshow)) +
(lownoiseresults(:,:,frame)<(minlow*percentshow))).*lownoiseresults(:,:,fra
me));
    mat2plotlow(isnan(mat2plotlow)) = 0;
    while ((sum([sum(~(mat2plotlow == 0))])>=shownum) && (limitloop == 0))
        percentshow = percentshow + 0.00001;
        mat2plotlow = (((lownoiseresults(:,:,frame)>(maxlow*percentshow)) +
(lownoiseresults(:,:,frame)<(minlow*percentshow))).*lownoiseresults(:,:,fra
me));
        mat2plotlow(isnan(mat2plotlow)) = 0;
        if percentshow >= 1
            limitloop = 1;
        end
    end
    plot_connections3d_regions(mat2plotlow, chanlabels, '', 'xyz',
chanlocarray, 'method', 'arrow', 'conlims', [minoverall maxoverall]);
    title({['Timeframe ' num2str(frame)],'Stimulus: Low Noise'})
    saveas(gcf,['stimulus_lownoise_timeframe_' num2str(frame) '_top.jpg']);
    if isempty(im)
        f = getframe(gcf);
        [im,map] = rgb2ind(f.cdata,256);

imwrite(im,map,'stimulus_lownoise_top.gif','gif','LoopCount',inf,'DelayTime
',1);
        im = [];
    else
        f = getframe(gcf);
        ifr = frame2im(f);
        [im,map] = rgb2ind(f.cdata,map);
        imwrite(im,map,'stimulus_lownoise_top.gif','gif','DisposalMethod',
'leaveInPlace', 'WriteMode','append','DelayTime',1);
    end

    view(backviewaz, backviewel)
    title({['Timeframe ' num2str(frame)],'Stimulus: Low Noise'})
    saveas(gcf,['stimulus_lownoise_timeframe_' num2str(frame)
'_back.jpg']);
    if isempty(im)
        f = getframe(gcf);
        [im,map] = rgb2ind(f.cdata,256);

imwrite(im,map,'stimulus_lownoise_back.gif','gif','LoopCount',inf,'DelayTim
e',1);
        im = [];
    else
        f = getframe(gcf);
        ifr = frame2im(f);
        [im,map] = rgb2ind(f.cdata,map);
        imwrite(im,map,'stimulus_lownoise_back.gif','gif','DisposalMethod',
'leaveInPlace', 'WriteMode','append','DelayTime',1);
    end
```

```
    view(sideviewaz, sideviewel)
    title({['Timeframe ' num2str(frame)],'Stimulus: Low Noise'})
    saveas(gcf,['stimulus_lownoise_timeframe_' num2str(frame)
'_side.jpg']);
    if isempty(im)
        f = getframe(gcf);
        [im,map] = rgb2ind(f.cdata,256);

imwrite(im,map,'stimulus_lownoise_side.gif','gif','LoopCount',inf,'DelayTim
e',1);
    else
        f = getframe(gcf);
        ifr = frame2im(f);
        [im,map] = rgb2ind(f.cdata,map);
        imwrite(im,map,'stimulus_lownoise_side.gif','gif','DisposalMethod',
'leaveInPlace', 'WriteMode','append','DelayTime',1);
    end
    if (frame == 1)
        im = [];
    end
    close all

    percentshow = 0;
    limitloop = 0;
    mat2plotmed = (((mednoiseresults(:,:,frame)>(maxmed*percentshow)) +
(mednoiseresults(:,:,frame)<(minmed*percentshow))).*mednoiseresults(:,:,fra
me));
    mat2plotmed(isnan(mat2plotmed)) = 0;
    while ((sum([sum(~(mat2plotmed == 0))])>=shownum) && (limitloop == 0))
        percentshow = percentshow + 0.00001;
        mat2plotmed = (((mednoiseresults(:,:,frame)>(maxmed*percentshow)) +
(mednoiseresults(:,:,frame)<(minmed*percentshow))).*mednoiseresults(:,:,fra
me));
        mat2plotmed(isnan(mat2plotmed)) = 0;
        if percentshow >= 1
            limitloop = 1;
        end
    end
    plot_connections3d_regions(mat2plotmed, chanlabels, '', 'xyz',
chanlocarray, 'method', 'arrow', 'conlims', [minoverall maxoverall]);
    title({['Timeframe ' num2str(frame)],'Stimulus: Medium Noise'})
    saveas(gcf,['stimulus_mednoise_timeframe_' num2str(frame) '_top.jpg']);
    if isempty(im)
        f = getframe(gcf);
        [im,map] = rgb2ind(f.cdata,256);

imwrite(im,map,'stimulus_mednoise_top.gif','gif','LoopCount',inf,'DelayTime
',1);
        im = [];
    else
        f = getframe(gcf);
        ifr = frame2im(f);
        [im,map] = rgb2ind(f.cdata,map);
        imwrite(im,map,'stimulus_mednoise_top.gif','gif','DisposalMethod',
'leaveInPlace', 'WriteMode','append','DelayTime',1);
    end

    view(backviewaz, backviewel)
    title({['Timeframe ' num2str(frame)],'Stimulus: Medium Noise'})
```

```
    saveas(gcf,['stimulus_mednoise_timeframe_' num2str(frame)
'_back.jpg']);
    if isempty(im)
        f = getframe(gcf);
        [im,map] = rgb2ind(f.cdata,256);

imwrite(im,map,'stimulus_mednoise_back.gif','gif','LoopCount',inf,'DelayTim
e',1);
        im = [];
    else
        f = getframe(gcf);
        ifr = frame2im(f);
        [im,map] = rgb2ind(f.cdata,map);
        imwrite(im,map,'stimulus_mednoise_back.gif','gif','DisposalMethod',
'leaveInPlace', 'WriteMode','append','DelayTime',1);
    end

    view(sideviewaz, sideviewel)
    title({['Timeframe ' num2str(frame)],'Stimulus: Medium Noise'})
    saveas(gcf,['stimulus_mednoise_timeframe_' num2str(frame)
'_side.jpg']);
    if isempty(im)
        f = getframe(gcf);
        [im,map] = rgb2ind(f.cdata,256);

imwrite(im,map,'stimulus_mednoise_side.gif','gif','LoopCount',inf,'DelayTim
e',1);
    else
        f = getframe(gcf);
        ifr = frame2im(f);
        [im,map] = rgb2ind(f.cdata,map);
        imwrite(im,map,'stimulus_mednoise_side.gif','gif','DisposalMethod',
'leaveInPlace', 'WriteMode','append','DelayTime',1);
    end
    if (frame == 1)
        im = [];
    end
    close all


    percentshow = 0;
    limitloop = 0;
    mat2plothigh = (((highnoiseresults(:,:,frame)>(maxhigh*percentshow)) +
(highnoiseresults(:,:,frame)<(minhigh*percentshow))).*highnoiseresults(:,:,
frame));
    mat2plothigh(isnan(mat2plothigh)) = 0;
    while ((sum([sum(~(mat2plothigh == 0))])>=shownum) && (limitloop == 0))
        percentshow = percentshow + 0.00001;
        mat2plothigh =
(((highnoiseresults(:,:,frame)>(maxhigh*percentshow)) +
(highnoiseresults(:,:,frame)<(minhigh*percentshow))).*highnoiseresults(:,:,
frame));
        mat2plothigh(isnan(mat2plothigh)) = 0;
        if percentshow >= 1
            limitloop = 1;
        end
    end
    plot_connections3d_regions(mat2plothigh, chanlabels, '', 'xyz',
chanlocarray, 'method', 'arrow', 'conlims', [minoverall maxoverall]);
```

```
    title({['Timeframe ' num2str(frame)],'Stimulus: High Noise'})
    saveas(gcf,['stimulus_highnoise_timeframe_' num2str(frame)
'_top.jpg']);
    if isempty(im)
        f = getframe(gcf);
        [im,map] = rgb2ind(f.cdata,256);

imwrite(im,map,'stimulus_highnoise_top.gif','gif','LoopCount',inf,'DelayTim
e',1);
        im = [];
    else
        f = getframe(gcf);
        ifr = frame2im(f);
        [im,map] = rgb2ind(f.cdata,map);
        imwrite(im,map,'stimulus_highnoise_top.gif','gif','DisposalMethod',
'leaveInPlace', 'WriteMode','append','DelayTime',1);
    end

    view(backviewaz, backviewel)
    title({['Timeframe ' num2str(frame)],'Stimulus: High Noise'})
    saveas(gcf,['stimulus_highnoise_timeframe_' num2str(frame)
'_back.jpg']);
    if isempty(im)
        f = getframe(gcf);
        [im,map] = rgb2ind(f.cdata,256);

imwrite(im,map,'stimulus_highnoise_back.gif','gif','LoopCount',inf,'DelayTi
me',1);
        im = [];
    else
        f = getframe(gcf);
        ifr = frame2im(f);
        [im,map] = rgb2ind(f.cdata,map);

imwrite(im,map,'stimulus_highnoise_back.gif','gif','DisposalMethod',
'leaveInPlace', 'WriteMode','append','DelayTime',1);
    end

    view(sideviewaz, sideviewel)
    title({['Timeframe ' num2str(frame)],'Stimulus: High Noise'})
    saveas(gcf,['stimulus_highnoise_timeframe_' num2str(frame)
'_side.jpg']);
    if isempty(im)
        f = getframe(gcf);
        [im,map] = rgb2ind(f.cdata,256);

imwrite(im,map,'stimulus_highnoise_side.gif','gif','LoopCount',inf,'DelayTi
me',1);
    else
        f = getframe(gcf);
        ifr = frame2im(f);
        [im,map] = rgb2ind(f.cdata,map);

imwrite(im,map,'stimulus_highnoise_side.gif','gif','DisposalMethod',
'leaveInPlace', 'WriteMode','append','DelayTime',1);
    end
    close all

end
```

```
%% correct/incorrect results

correctresults = zeros([size(difference{1,1,1,1}),
length(difference(1,1,1,:))]);
incorrectresults = zeros([size(difference{1,1,1,1}),
length(difference(1,1,1,:))]);


for frame = 1:poststimtimeframes %change back
    clear temp
    clear temp2
    empty = ~cellfun( @isempty, difference(:,:,1,frame));
    temp = difference(:,:,1,frame);
    temp = temp(empty);
    for i = 1:length(temp)
        temp2(:,:,i) = temp{i};
    end
    correctresults(:,:,frame) = mean(temp2,3);

    clear temp
    clear temp2
    empty = ~cellfun( @isempty, difference(:,:,2:3,frame));
    temp = difference(:,:,2:3,frame);
    temp = temp(empty);
    for i = 1:length(temp)
        temp2(:,:,i) = temp{i};
    end
    incorrectresults(:,:,frame) = mean(temp2,3);

end


maxcorrect = max(correctresults(:));
maxincorrect = max(incorrectresults(:));
mincorrect = min(correctresults(:));
minincorrect = min(incorrectresults(:));

maxoverall = max([maxcorrect maxincorrect]);
minoverall = min([mincorrect minincorrect]);
if maxoverall>abs(minoverall)
    minoverall = -maxoverall;
else
    maxoverall = abs(minoverall);
end

shownum = 8;
im = [];
for frame = 1:poststimtimeframes %change back
    percentshow = 0;
    limitloop = 0;
    mat2plotcorrect =
(((correctresults(:,:,frame)>(maxcorrect*percentshow)) +
(correctresults(:,:,frame)<(mincorrect*percentshow))).*correctresults(:,:,f
rame));
    mat2plotcorrect(isnan(mat2plotcorrect)) = 0;
    while ((sum([sum(~(mat2plotcorrect == 0))])>=shownum) && (limitloop ==
0))
```

```
        percentshow = percentshow + 0.00001;
        mat2plotcorrect =
(((correctresults(:,:,frame)>(maxcorrect*percentshow)) +
(correctresults(:,:,frame)<(mincorrect*percentshow))).*correctresults(:,:,f
rame));
        mat2plotcorrect(isnan(mat2plotcorrect)) = 0;
        if percentshow >= 1
            limitloop = 1;
        end
    end
    plot_connections3d_regions(mat2plotcorrect, chanlabels, '', 'xyz',
chanlocarray, 'method', 'arrow', 'conlims', [minoverall maxoverall]);
    title({['Timeframe ' num2str(frame)],'Stimulus: Correct'})
    saveas(gcf,['stimulus_correct_timeframe_' num2str(frame) '_top.jpg']);
    if isempty(im)
        f = getframe(gcf);
        [im,map] = rgb2ind(f.cdata,256);

imwrite(im,map,'stimulus_correct_top.gif','gif','LoopCount',inf,'DelayTime'
,1);
        im = [];
    else
        f = getframe(gcf);
        ifr = frame2im(f);
        [im,map] = rgb2ind(f.cdata,map);
        imwrite(im,map,'stimulus_correct_top.gif','gif','DisposalMethod',
'leaveInPlace', 'WriteMode','append','DelayTime',1);
    end

    view(backviewaz, backviewel)
    title({['Timeframe ' num2str(frame)],'Stimulus: Correct'})
    saveas(gcf,['stimulus_correct_timeframe_' num2str(frame) '_back.jpg']);
    if isempty(im)
        f = getframe(gcf);
        [im,map] = rgb2ind(f.cdata,256);

imwrite(im,map,'stimulus_correct_back.gif','gif','LoopCount',inf,'DelayTime
',1);
        im = [];
    else
        f = getframe(gcf);
        ifr = frame2im(f);
        [im,map] = rgb2ind(f.cdata,map);
        imwrite(im,map,'stimulus_correct_back.gif','gif','DisposalMethod',
'leaveInPlace', 'WriteMode','append','DelayTime',1);
    end

    view(sideviewaz, sideviewel)
    title({['Timeframe ' num2str(frame)],'Stimulus: Correct'})
    saveas(gcf,['stimulus_correct_timeframe_' num2str(frame) '_side.jpg']);
    if isempty(im)
        f = getframe(gcf);
        [im,map] = rgb2ind(f.cdata,256);

imwrite(im,map,'stimulus_correct_side.gif','gif','LoopCount',inf,'DelayTime
',1);
    else
        f = getframe(gcf);
        ifr = frame2im(f);
        [im,map] = rgb2ind(f.cdata,map);
```

```
        imwrite(im,map,'stimulus_correct_side.gif','gif','DisposalMethod',
'leaveInPlace', 'WriteMode','append','DelayTime',1);
    end
    if (frame == 1)
        im = [];
    end
    close all


    percentshow = 0;
    limitloop = 0;
    mat2plotincorrect =
(((incorrectresults(:,:,frame)>(maxincorrect*percentshow)) +
(incorrectresults(:,:,frame)<(minincorrect*percentshow))).*incorrectresults
(:,:,frame));
    mat2plotincorrect(isnan(mat2plotincorrect)) = 0;

    while ((sum([sum(~(mat2plotincorrect == 0))])>=shownum) && (limitloop
== 0))
        percentshow = percentshow + 0.00001;
        mat2plotincorrect =
(((incorrectresults(:,:,frame)>(maxincorrect*percentshow)) +
(incorrectresults(:,:,frame)<(minincorrect*percentshow))).*incorrectresults
(:,:,frame));
        mat2plotincorrect(isnan(mat2plotincorrect)) = 0;
        if percentshow >= 1
            limitloop = 1;
        end
    end
    plot_connections3d_regions(mat2plotincorrect, chanlabels, '', 'xyz',
chanlocarray, 'method', 'arrow', 'conlims', [minoverall maxoverall]);
    title({['Timeframe ' num2str(frame)],'Stimulus: Incorrect'})
    saveas(gcf,['stimulus_incorrect_timeframe_' num2str(frame)
'_top.jpg']);
    if isempty(im)
        f = getframe(gcf);
        [im,map] = rgb2ind(f.cdata,256);

imwrite(im,map,'stimulus_incorrect_top.gif','gif','LoopCount',inf,'DelayTim
e',1);
        im = [];
    else
        f = getframe(gcf);
        ifr = frame2im(f);
        [im,map] = rgb2ind(f.cdata,map);
        imwrite(im,map,'stimulus_incorrect_top.gif','gif','DisposalMethod',
'leaveInPlace', 'WriteMode','append','DelayTime',1);
    end

    view(backviewaz, backviewel)
    title({['Timeframe ' num2str(frame)],'Stimulus: Incorrect'})
    saveas(gcf,['stimulus_incorrect_timeframe_' num2str(frame)
'_back.jpg']);
    if isempty(im)
        f = getframe(gcf);
        [im,map] = rgb2ind(f.cdata,256);

imwrite(im,map,'stimulus_incorrect_back.gif','gif','LoopCount',inf,'DelayTi
me',1);
```

```
        im = [];
    else
        f = getframe(gcf);
        ifr = frame2im(f);
        [im,map] = rgb2ind(f.cdata,map);

imwrite(im,map,'stimulus_incorrect_back.gif','gif','DisposalMethod',
'leaveInPlace', 'WriteMode','append','DelayTime',1);
    end

    view(sideviewaz, sideviewel)
    title({['Timeframe ' num2str(frame)],'Stimulus: Incorrect'})
    saveas(gcf,['stimulus_incorrect_timeframe_' num2str(frame)
'_side.jpg']);
    if isempty(im)
        f = getframe(gcf);
        [im,map] = rgb2ind(f.cdata,256);

imwrite(im,map,'stimulus_incorrect_side.gif','gif','LoopCount',inf,'DelayTi
me',1);
    else
        f = getframe(gcf);
        ifr = frame2im(f);
        [im,map] = rgb2ind(f.cdata,map);

imwrite(im,map,'stimulus_incorrect_side.gif','gif','DisposalMethod',
'leaveInPlace', 'WriteMode','append','DelayTime',1);
    end
    close all

end
```

12.3 Connectivity Statistics

This is the script to calculate the statistics of the across stimulus p values – statistics of the correct vs incorrect only or the noise level comparison is attached in a zip file of all the code.

```
%% switches
warning( 'off', 'all');
clear all
new_file = false;

%% preamble
sharedrootpath = getpref( 'tsg', 'sharedrootpath');
homepath = fullfile( sharedrootpath, 'Projects\AVSP\McGurk');
matlabrootpath = getpref( 'tsg', 'matlabrootpath');
addpath( genpath( 'V:\EEG\People\Tyler'));
addpath( genpath( 'V:\EEG\People\Caitlin'));
addpath( genpath( 'V:\EEG\People\Kenneth\Matlab\bin\connectivity
measures\'));
addpath( genpath( 'V:\EEG\People\Kenneth\Matlab\contrib'));
eeglabpath = fullfile( matlabrootpath, 'Matlab', 'eeglab');
hdmfile = fullfile(eeglabpath,'plugins','dipfit2.3', ...
    'standard_BEM','standard_vol.mat');

%% define variables
```

```
subjects = { 'Subject01', 'Subject02', 'Subject03', 'Subject05',
'Subject06', ...
     'Subject07', 'Subject08', 'Subject09', 'Subject10', ...
     'Subject11', 'Subject12', 'Subject13', 'Subject14', 'Subject15', ...
    'Subject16', 'Subject17', 'Subject18', 'Subject19', 'Subject20',
'Subject21'};
tasks = { 'McGurk-Audio-degradation'};
ICA = 'AMICA';
CONN = 'CGC';
sounds = { 'ABBA', 'ABBA-AO', 'APPA', 'ATHA', 'AGGA', 'AGGA-AO'};
noises = { 'LN', 'MN', 'HN'};
perception = { 'Truth+Perceived', 'Perceived-only', 'Truth-only'};
bands = {'theta', 'alpha', 'beta', 'gamma'};

%% Number of ...
Ns = numel( subjects);
Nt = numel( tasks);
Nso = numel( sounds);
Nno = numel( noises);
Np = numel( perception);
Nba = numel(bands);

% STILL NEED TO ADJUST FOR BANDS

%% calculate coherence
wait1 = tic;
for t = 1:Nt
    task = tasks{ t};

    switch task
        case 'McGurk-Audio-degradation'
            AMs = cell( Nso, Nno, Np, Ns);


            % time definition
            times( :, 1) = -1:0.1:0.8;
            times( :, 2) = -0.8:0.1:1;

            %load subject channels and connectivity
            for s = 1:Ns
                subject = subjects{ s};

                % define folderpath
                folderpath = fullfile( homepath, 'data', subject, task);

                % load epoch file
                d = eeg3.eeg.load( fullfile( folderpath, sprintf(
'%s_%s_epoched_IC.eeg3-eeg', ...
                    task, ICA)));
                chaninfo{ s} = d{ 1}( 1).chan;
                clear d

                % load connectivity file
                temp = load( fullfile( folderpath, sprintf( '%s_%s_%s.mat',
...
                    task, ICA, CONN)));
                AMs( :, :, :, s) = temp.AMs;
                chans{ s} = temp.chans;
```

```
        clear temp


        disp( '-----------------------------------');
        fprintf( 'Subject: %d of %d\n', s, Ns);
        fprintf( 'Task: %s, %d of %d\n', task, t, Nt);
        disp( 'Data loading completed');
        disp( '-----------------------------------');

    end

    % intersect channels
    channels = chans{ 1};
    chanlabels = channels;
    for s = 2:numel( chans)
        chanlabels = intersect( chans{ s}, chanlabels);
    end

    % index into AMs
    for s = 1:size( AMs, 4)
        ind = find( ismember( chans{ s}, chanlabels));
        AM = AMs( :, :, :, s);
        for p = 1:numel( AM),
            if isempty( AM{ p}), continue; end
            AM{ p} = AM{ p}( ind, ind, :, :);
        end
        AMs( :, :, :, s) = AM;
        clear AM
    end

    % number of channels
    Nchan = numel( chanlabels);

    %% stats processing


    dec = cell(Nso, Nno, Np);
    inc = cell(Nso, Nno, Np);

    for i = 1:Nso %cycle through sounds
        for ii = 1:Nno %cylce through noises
            for iii = 1:Np %cycle thorugh perceptions
                AM = [];
                clear tempall
                tempall = AMs(i,ii,iii,:);
                tempall = tempall(:);
                k = 1;
                for l = 1:length(tempall)
                    clear temp3
                    if (~isempty(tempall{l}))
                        temp3 = tempall{l};
                        for frame = 11:15
                            if (length(size(temp3))>3)
                                for m = 1:length(temp3(1,1,1,:))
                                    AM(:,:,k) =
mean(temp3(:,:,4:5,m),3) - mean(temp3(:,:,2,m),3);
                                    AM(:,:,k+1) =
temp3(:,:,frame,m) - mean(temp3(:,:,2,m),3);
```

```
                                        k = k + 2;
                                    end
                                else
                                    AM(:,:,k) = mean(temp3(:,:,4:5),3)
- mean(temp3(:,:,2),3);
                                    AM(:,:,k+1) = temp3(:,:,frame) -
mean(temp3(:,:,2),3);
                                    k = k + 2;
                                end
                            end
                        end
                    end

                    num = 1:2:size( AM, 3);
                    design = zeros( size( AM, 3), 2);
                    design( num, 1) = 1;
                    design( num+1, 2) = 1;

                    %define stats parameters
                    stats_method = 'NBS';
                    GLM.perms = fastif( strcmp( stats_method, 'FDR'),
50000, 5000);
                    GLM.X = design;
                    GLM.test = 'ttest'; % 'ttest' or 'ftest'
                    STATS.size = 'Intensity'; %'Intensity' or 'Extent'
                    STATS.thresh = 0.3;
                    STATS.alpha = 0.05;

                    if (~isempty(AM))

                        % increase
                        GLM.contrast = [-1 1];
                        inc{i,ii,iii} = NBS_directional_hack( AM, GLM,
STATS, chanlabels);

                        % decrease
                        GLM.contrast = [1 -1];
                        dec{i,ii,iii} = NBS_directional_hack( AM, GLM,
STATS, chanlabels);

                    end

                    clear AM

                end
            end
        end

        save(['V:\EEG\People\Caitlin\' CONN '_inc.mat'], 'inc');
        save(['V:\EEG\People\Caitlin\' CONN '_dec.mat'], 'dec');

        %% pull out for easy viewing

        decsize = size(dec);
        incsize = size(inc);
        dec_p_val = cell(decsize(1),decsize(2),decsize(3));
        dec_con_mats = cell(decsize(1),decsize(2),decsize(3));
        inc_p_val = cell(incsize(1),incsize(2),incsize(3));
```

```
            inc_con_mats = cell(incsize(1),incsize(2),incsize(3));
            for k = 1:decsize(1)
                for kk = 1:decsize(2)
                    for kkk = 1:decsize(3)
                            dec_struct = dec{k,kk,kkk};
                            inc_struct = inc{k,kk,kkk};
                            if (~isempty(dec_struct) &&
~isempty(dec_struct.pval))
                                dec_p_val{k,kk,kkk} = dec_struct.pval;
                                dec_con_mats{k,kk,kkk} =
dec_struct.con_mat;
                            end
                            if (~isempty(inc_struct) &&
~isempty(inc_struct.pval))
                                inc_p_val{k,kk,kkk} = inc_struct.pval;
                                inc_con_mats{k,kk,kkk} =
inc_struct.con_mat;
                            end
                    end
                end
            end

            save(['V:\EEG\People\Caitlin\' CONN '_dec_p_val.mat'],
'dec_p_val');
            save(['V:\EEG\People\Caitlin\' CONN '_dec_con_mats.mat'],
'dec_con_mats');
            save(['V:\EEG\People\Caitlin\' CONN '_inc_p_val.mat'],
'inc_p_val');
            save(['V:\EEG\People\Caitlin\' CONN '_inc_con_mats.mat'],
'inc_con_mats');


    end
end
fprintf( 'Time elapsed from connectivity stats: %0.2f minutes\n', toc(
wait1)/60);
```