Author:
Blyth, Mark D

Title:
Recognise my emotions

on the automatic recognition of emotions from human speech

University of Bristol

Masters Thesis

Recognise My Emotions - On The Automatic Recognition Of Emotions From Human Speech

Author:
M. Blyth

Supervised by:
Dr K. Shalonova
Dr R. Szalai

Project Report submitted in support of the degree Master of Engineering

Department of Engineering Mathematics

Compiled April 10, 2019

Abstract

The ability to automatically recognise human emotions will open up a range of new possibilities for human-computer interaction. Such possibilities will become increasingly valuable, as autonomous computational agents reach ubiquity. Here, the recognition of emotions in human speech is considered. While methods exist for this, they demonstrate a low classification accuracy. It is argued here that the poor performance of these existing solutions arises from the choice of classification features used. Results from dynamical systems theory are used to develop a new classification approach. To achieve this, speech production dynamics are reconstructed from audio recordings. The equilibria, Lyapunov exponents, and correlation dimension of the dynamics are extracted. A feature space is defined on these data. A classification accuracy of 74% is achieved, when distinguishing between calm and angry emotional speech. The performance of the classifier reduces to 54% for fearful and sad emotions. A set of novel algorithms are proposed, for determining the embedding dimension and nonlinear equilibria of a system from time series data, and to detect voiced speech in audio recordings.

Contents

Chapter 1

Introduction

This project considers the recognition of emotions in human speech. Current human-computer interaction is dominated by explicit user inputs, typically through a mouse and keyboard. Zeng et al. make the argument that for computational agents to fully integrate into everyday life, a switch from this explicit, computer-oriented interface, to a more implicit, human-oriented interface, will become necessary [52]. This requires systems to exhibit artificial emotional intelligence. A fully human-orientated interface must be able to identify the emotional state of the user, and respond appropriately. This emotional awareness allows the construction of artificial emotional intelligence in computers and autonomous agents, which in turn provides opportunities for more natural and more meaningful interactions between humans and computers.

Emotion-aware interfaces are not a new idea. Recently, Hernandez et al. used an emotion recognition system to quantify driver stress in an intelligent car; the car then adapts its interactions with the driver in such a way as to help manage stress [13]. More generally, if the emotional state of an individual can be accurately assessed, a computational agent can seek to interact with the user in such a way as to maximise user utility. This could be used to improve worker safety, or to increase user satisfaction with a product or service. It is hoped that improvements in automatic emotion recognition, as investigated in this project, will open up more possibilities for the implementation of emotionally intelligent systems.

Several existing emotion recognition systems for audio data are examined in section 1.2. Current research focuses primarily on using prosodic features to classify emotions. These seek to quantify not what is spoken, but how it is spoken. Emotion recognition systems trained on prosodic features generally exhibit a low classification accuracy - typically around 70%. Section 1.2 makes the argument that prosodic features are heavily speaker-dependent, and are therefore of limited use for emotion recognition. This project seeks to overcome these limitations, by developing an alternative method for recognising emotions from speech recordings. Here, arguments based on human physiology and dynamical systems theory are used to motivate a novel feature extraction method. The dynamics of the vocal chords are reconstructed from audio recordings. Features of the reconstructed dynamics are used for emotion classification. It is hoped that this approach will overcome the limitations of conventional prosodic approaches, and therefore lead to an improvement in classification accuracy.

1.1 Report plan

Solutions already exist for recognising emotions in audio data. Nevertheless, these show too low a classification accuracy to be useful in real-world systems. Section 1.2 considers existing methods, and concludes that their accuracy is limited by the feature sets that are used. An

alternative classification approach is proposed, which considers the effects of emotions on the dynamics of the vocal chords.

Speech contains a mixture of sounds, created by both the vocal chords (voiced speech), and in the oral cavity (unvoiced speech) [11]. Only the dynamics of the vocal chords are considered in this project. Hence, voiced speech must be identified, and separated from silence, noise, and unvoiced speech. Section 2.1 proposes a method for classifying short windows of speech as being voiced or unvoiced. The voiced sections are then retained for further analysis.

It is conjectured here that the dynamics of the vocal chords will contain sufficient information to classify speaker emotions. The dynamics are studied by means of an attractor reconstruction. Section 2.2 discusses how to construct a representation of audio data in such a way that the vocal fold dynamics are extracted. A state space is generated from audio data, such that the dynamics of the resulting state vectors are topologically equivalent to the dynamics of the speech production system.

To reason analytically about the speech production dynamics, a model is fitted to the reconstructed state space. Section 2.3.1 considers the fitting of a linear model to the data. It is found that the linear model contains insufficient information for categorising emotions. These results are used to motivate the use of a nonlinear state space model.

Section 2.3.2 considers the application of nonlinear methods to the problem. A method for fitting nonlinear models is proposed. Issues with fitting the model are discussed.

The nonlinear model is analysed through consideration of the model equilibria. Due to the nature of the model, finding meaningful equilibria is not trivial. Section 2.3.3 considers the issues in identifying meaningful equilibria from the fitted nonlinear model.

Nonlinear dynamics offers many methods for analysing systems. Section 2.4 considers how to apply some of these methods to emotion classification. Features such as Lyapunov exponents, equilibrium stability, and fractal dimension are discussed.

Section 2.5 considers whether the set of extracted dynamical features are sufficient for classifying emotions. A classifier is constructed and tested. It is found that the feature set is capable of classifying some pairs of emotions, but not others.

1.2 Literature review

Several studies have been conducted on auditory emotion recognition. These use prosodic features to classify speech recordings into different emotions. Prosodic features seek to quantify not what a speaker is saying, but how it is being said. Ververidis et al. provide a comprehensive review of commonly used prosodic features [49]. The most popular include the pitch and rate of speech. Pitch is often measured by the fundamental frequency F0 of the speech signal; rate can be quantified by features such as syllables or words per minute. Prosodic features such as these aim to identify speaking patterns that a human listener would pick up on. Nevertheless, existing prosody-based emotion classifiers typically exhibit a low classification accuracy. Here, it is argued that prosodic features are too dependent on the speaking style of an individual, resulting in classifiers with a limited classification accuracy.

Petrushin considers two different emotion recognition scenarios [31]. First, prosodic features are extracted from recordings of actors portraying different emotions. A neural network is trained on these features. The average classification accuracy is quoted as 'about 70%', dropping to as low as 35-55% for utterances exhibiting fear. Low classification accuracies such as these are of little practical value. Furthermore, the accuracies are hugely inconsistent between the different emotions. To become useful in real-world applications, a classifier must perform well across all emotions.

The second scenario considered by Petrushin uses prosodic features to discriminate between agitated and calm emotional states. A recogniser is capable of distinguishing between the two states with a 77% accuracy. This is an improvement on the classification accuracy for the previous single-emotion problem, demonstrating that stress states can be determined more accurately than specific emotions. It is therefore possible that the classification accuracy of the stress-state problem could be improved further, by seeking features that explicitly represent the effects of stress on speech. Physiological arguments will be used later to further motivate this idea.

Lexical features are the opposite of prosody, in that they seek to extract information from the structure and meaning of words. It is reasonable to consider whether the low classification accuracy achieved by Petrushin could be improved with the addition of lexical information. Hirschberg et al. seek to classify recordings into deceptive and non-deceptive speech [16]. The study considers a similar set of prosodic features to those chosen by Petrushin. In addition, lexical features are also considered. The word patterns used by a speaker are analysed with the lexical categorisation program LIWC [30], and used as classification features. Using entirely prosodic features, a classification error of 38.5% is obtained; using entirely lexical features, a 39.0% classification error is obtained. Using both lexical and prosodic features reduces this error to 37.2%. The improvement in accuracy attained by considering both feature sets is minimal, suggesting that prosodic and lexical features contain a large amount of mutual information. It is therefore reasonable to only consider either prosodic or lexical features. Furthermore, the low classification accuracy achieved by Petrushin cannot be attributed to the omission of lexical information.

Hirschberg et al. also consider personalised classifiers, trained and tested on audio samples from each individual speaker. Using this approach, the classification error is reduced to 33.6%. A large improvement in classification error is obtained when training a different classifier for each speaker; this implies that prosodic and lexical emotional expression differs between speakers. Features derived from these properties are therefore indicative as much about emotional states, as they are representative of the speech patterns of the speaker. They are hence a poor choice for emotion recognition tasks. Furthermore, Batliner et al. note that for realistic, non-acted scenarios, prosody ceases to be a reliable indicator of emotion [3]. A more general, speaker-independent set of features must be produced, if classification accuracies are to be improved. This project seeks to identify such features.

Scherer proposes that emotions cause a physiological response that affects speech production [40]. This physiological response is characterised by such changes as an increase or decrease in vocal chord tension, which in turn causes changes in the produced speech [22]. This is backed up by Tolkmitt et al., who investigate the impact of stress on speech production [48]. Differences in vocal parameters are found between stressed and unstressed conditions. Tolkmitt et al. suggest that these differences arise as a result of physiological changes in muscle tension, reinforcing the work of Scherer. The feature extraction method developed here builds on these ideas, by using techniques from dynamical systems theory to attempt to isolate the differences in stress

Emotion	Calm	Happy	Sad	Angry	Fearful	Disgust	Surprise	Neutral
Strong	0.75	0.44	0.62	0.91	0.73	0.54	0.74	n/a
Normal	0.79	0.29	0.34	0.59	0.59	0.50	0.62	0.91

Figure 1.1: Validation results for the RAVDESS data set. Human volunteers are asked to label randomly selected audio recordings from the database; the proportion of correct labels is shown. Taken from Livingstone and Russo [24].

response, and using these differences as features for emotion classification. This method aims to produce a more speaker-independent result than can be obtained from the consideration of prosody alone.

Speech production is a complex nonlinear phenomenon [15] [4] [18]. Herzel considers bifurcations in these nonlinear dynamics; it is found that slightly varying parameters such as muscle tensions in the speech-production system can induce qualitative changes in the produced voice signal [14]. Steinecke et al. build on this, to show that changes in the stiffness of the left and right vocal folds cause bifurcations to occur in the speech production dynamics [44]. One possible cause of these stiffness changes are changes in muscle tension within the vocal tract. The work of Tolkmitt et al. suggests that inducing stress causes a change in vocal tract muscle tension [48]. These results together present the possibility that differences in the emotional state of a speaker will cause different speech production dynamics. The work presented here thus relies on the conjecture that speech production dynamics will exhibit a speaker-independent response to emotional stimuli, that can be detected, extracted, and classified from, using an appropriate dynamical analysis.

1.3 Data

The work presented here uses the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [24], chosen due to its large number of audio recordings, and rigorous validation procedure. The database consists of 24 professional actors, speaking two short sentences in a North American accent. Each sentence is repeated twice per actor, in calm, happy, sad, angry, fearful, surprise, disgust, and neutral emotions. For all but the neutral case, recordings are repeated in both normal and strong emotional intensities. The database contains 1400 recordings in total - four normal and four strong recordings per emotion, plus four neutral clips, for each of the 24 actors.

To ensure the data are realistic, the audio recordings are validated by 247 participants. The target emotion of each recording is guessed by a minimum of ten different individuals. The proportion of correct labels are shown in table 1.1.

The strong-intensity emotional portrayals are generally well-recognised. The exception to this is ‘happy’, which was only recognised correctly by 44% of participants. The normal-intensity portrayals are generally poorly recognised, with the exception of ‘neutral’. Strong-angry and neutral are both identified correctly 91% of the time. Thus, while the work presented here will consider all the emotions, extra consideration should be paid to the strong-angry and the neutral results.

One would expect to achieve 50% accuracy on a pairwise classification, through random guessing alone. If the entire data set is considered, as opposed to pairs of emotions, one would expect a recogniser that matched the performance of the human validators to achieve a 67%

accuracy on strong data, 53% on normal (excluding neutral), and 62% on all data (including neutral). At the time of writing, few auditory emotion recognition studies have been undertaken with RAVDESS. Jannat et al. use a deep learning approach to achieve a 66% accuracy on classifying happy and sad audio recordings from RAVDESS [17]. These results provide a baseline classification accuracy, against which the results presented here can be compared.

Chapter 2

Methodology

2.1 Voiced Speech Detection

Audio recordings can be loosely categorised as containing three types of data - silence, voiced speech, and unvoiced speech. Voiced speech is produced by periodic pulses of air from the glottis, which are then filtered by the vocal tract [11]. Voiced speech produces a periodic waveform, and is typically associated with vowel sounds. Unvoiced speech consists of random-like, non-periodic wave forms, produced by the passage of air through a narrow constriction in the oral cavity.

The dynamics of the vocal chords cannot be directly studied in signals containing silence, or significant amounts of unvoiced noise. These non-voiced signals must be removed from the data before any dynamical analysis can be conducted. Voiced speech detection remains an open problem in signal processing. Here, a simple and computationally efficient voiced-unvoiced classifier is proposed, to extract voiced sections of data from an audio signal. Due to a lack of labelled training data, an unsupervised learning approach is used.

To construct the classifier, the periodic structure of voiced data is exploited. Fourier theory demonstrates that any periodic signal can be expressed as a linear combination of oscillatory basis functions. Consider a periodic signal $f(t)$, with period T. Let $\omega_0 = \frac{2\pi}{T}$ be the fundamental frequency of the signal. For parameters a_n, b_n, the harmonic expansion of $f(t)$ can be expressed as

$$f(t) = \sum_{n=0}^{\infty} \left[a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t) \right] \ . \tag{2.1}$$

For a given section of audio data, pure silence can be easily detected as an extended sequence of magnitude-zero samples. Impure silence - silence contaminated with random background noise - can be considered, along with unvoiced data, as being a random signal. To discriminate between voiced speech and noise signals, the goodness-of-fit of the harmonic model in equation (2.1) is considered. The classifier here seeks only to extract the voiced sections of an audio signal. Consequently, noisy silence is detected and removed in exactly the same way as unvoiced speech.

First, the target audio recording is split into contiguous, non-overlapping windows. (Note that, while overlapping windows would provide a more fine-grained approach, it would also be

significantly more computationally expensive.) Conventional speech processing methods split data into windows of duration 10 to 30 milliseconds, under the assumption that the speech signal remains stationary within each window [28]. This windowing approach is used here.

Next, the harmonic model in equation (2.1) is fitted to each window of audio data. This requires one to first compute a maximum likelihood estimate of ω_0, and then to fit the harmonic model parameters a_n and b_n. For a given audio window, the residuals of the model are calculated. The mean of the squared residuals is treated as a random variable. This mean squared error is passed to a Bayesian classifier, which determines if the audio sequence contains voiced data. Each window is classified as either containing voiced data, or the contrary. Windows lacking voiced data are discarded. Finally, all contiguous windows that are classified as containing voiced data are joined together, to form a set of the longest possible voiced sub-sequences within the signal. These voiced sub-sequences are retained for further analysis.

To fit the harmonic model, the fundamental frequency ω_0 must first be found. Conventionally, fundamental frequency estimation is achieved through examination of the signal autocorrelation function [33]. While fast, autocorrelation methods typically exhibit a low prediction accuracy [27]. Alternatively, ω_0 can be fitted by minimising the square error of the harmonic model. For signals perturbed with Gaussian white noise, this method is identical to the statistically optimal method of maximum likelihood estimation for ω_0. Nevertheless, fitting ω_0 in this way requires a nonlinear least squares method, which is computationally inefficient. Nielsen et al. address this issue, and present a fast algorithm for computing the nonlinear least squares fit for ω_0 [27]. The fundamental frequency estimation algorithm of Nielsen et al. is adopted here, due to the statistical optimality and computational efficiency of the method.

For a given estimate of ω_0, fitting the harmonic model can be approached as an ordinary least squares problem. Consider the fitting of a harmonic model up to the kth harmonic. Assume a discrete-sampled signal $x_t = f(t)$, with samples taken at times $t \in \{1, 2, \ldots, N\}$. Assume the signal has been transformed to have a zero DC offset. From equation (2.1), each sample x_t can be expressed as

$$
\begin{aligned}
x_1 &= \sum_{n=1}^{k} [a_n \cos(n\omega_0) + b_n \sin(n\omega_0)] \ , \\
x_2 &= \sum_{n=1}^{k} [a_n \cos(2n\omega_0) + b_n \sin(2n\omega_0)] \ , \\
x_3 &= \sum_{n=1}^{k} [a_n \cos(3n\omega_0) + b_n \sin(3n\omega_0)] \ , \\
&\vdots \\
x_N &= \sum_{n=1}^{k} [a_n \cos(Nn\omega_0) + b_n \sin(Nn\omega_0)] \ .
\end{aligned}
$$

Consider the problem in matrix form. Let

$$
A = \begin{pmatrix}
\cos(\omega_0) & \cos(2\omega_0) & \cdots & \cos(k\omega_0) & \sin(\omega_0) & \sin(2\omega_0) & \cdots & \sin(k\omega_0) \\
\cos(2\omega_0) & \cos(4\omega_0) & \cdots & \cos(2k\omega_0) & \sin(2\omega_0) & \sin(4\omega_0) & \cdots & \sin(2k\omega_0) \\
\cos(3\omega_0) & \cos(6\omega_0) & \cdots & \cos(3k\omega_0) & \sin(3\omega_0) & \sin(6\omega_0) & \cdots & \sin(3k\omega_0) \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\cos(N\omega_0) & \cos(2N\omega_0) & \cdots & \cos(Nk\omega_0) & \sin(N\omega_0) & \sin(2N\omega_0) & \cdots & \sin(Nk\omega_0)
\end{pmatrix} ,
$$

$$\mathbf{x} = [x_1,\ x_2, \ldots, x_N]^T\ ,$$

$$\mathbf{k} = [a_1,\ a_2, \ldots,\ a_k,\ b_1,\ b_2, \ldots, b_k]^T\ .$$

The harmonic model therefore becomes $\mathbf{x} = A\mathbf{k}$. The fundamental frequency ω_0 is given by $\arg\min_{\omega_0} \left\|\mathbf{x} - A\hat{\mathbf{k}}\right\|$; for a given ω_0, the least-squares-optimal estimate $\hat{\mathbf{k}}$ of parameter vector $\mathbf{k}$ is given by $\hat{\mathbf{k}} = (A^T A)^{-1} A^T \mathbf{x}$. The total residual squared error is $\left\|\mathbf{x} - A\hat{\mathbf{k}}\right\|_2^2$, giving a model mean square prediction error (MSPE) of

$$\text{MSPE} = \min_{\omega_0} \left(\frac{1}{N} \left\|\mathbf{x} - A\hat{\mathbf{k}}\right\|_2^2 \right)\ . \tag{2.2}$$

By fitting the model on signals that have been normalised to have a total energy of one, the prediction errors become directly comparable across all signals.

For a given mean square prediction error, one must now determine whether the audio sample contains voiced speech. To achieve this, a classifier is constructed, as follows. Let X be a random variable representing the mean square prediction error of some audio sample. Let x be the actual mean square prediction error of the audio sample, as calculated with equation (2.2). First, the probability distribution $P(X = x)$ is estimated. Next, a model for $P(X = x|v)$ is fitted, where v indicates that the sample contains voiced data. Finally, Bayes' rule is used to condition on evidence $X = x$, to get $P(v|X = x)$. If the posterior voiced probability $P(v|X = x)$ is greater than 0.5, the audio window is classified as containing voiced speech; otherwise, the data is classified as unvoiced.

First, the probability distribution of X must be estimated. This is achieved by using a Monte Carlo method to produce realisations of X, then estimating the distribution $P(X = x)$ from this. To apply the Monte Carlo method, the set of all audio recordings in the data set is considered. Windows of fixed lengths, here 10ms, are randomly selected from these recordings, and the mean square prediction error of the harmonic model is calculated over these windows. Each calculated prediction error produces a realisation of X, with the realisation denoted x. Let n be the number of realisations of X. As n becomes large, the set of realisations will become statistically representative of the probability distribution $P(X = x)$. To estimate the distribution, a kernel density estimation method is used [43]. Consider some kernel K, such that

$$\int_{-\infty}^{\infty} K(x)\mathrm{d}x = 1\ \ .$$

For n realisations x_i, the density estimate $\hat{f}(x)$ of $P(X = x)$ is given by

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} K(x - x_i)\ \ .$$

That is, the estimate $\hat{f}(x)$ is given by the mean of a set of kernels, with each kernel centered on a realisation of X. For a suitably chosen kernel K, this density estimate $\hat{f}(x)$ will converge to the actual probability density function $P(X = x)$. A Gaussian kernel is typically used for density estimation; the work produced here assumes this convention.

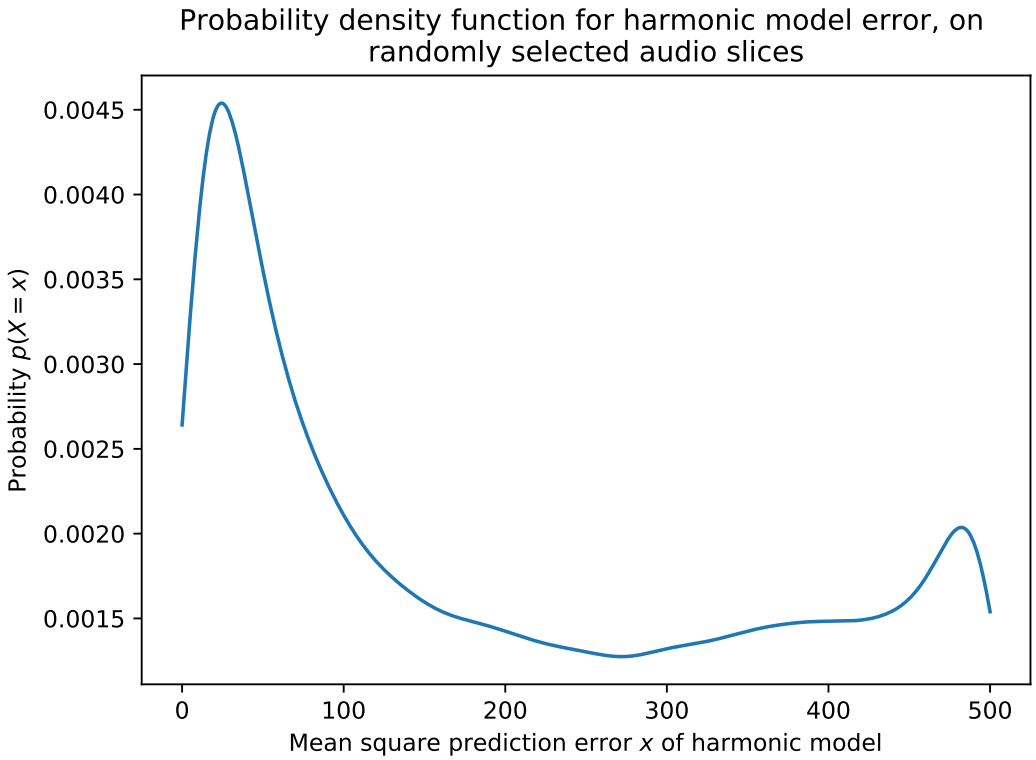


Figure 2.1: Kernel density estimate of the probability density function $P(X = x)$, where x is a realisation of the mean square prediction error X, as calculated by equation (2.2).

Figure 2.1 shows a kernel density estimate of the density function $P(X = x)$, as calculated from 10^5 Monte-Carlo-sampled realisations of X. The audio samples contain both voiced and unvoiced data. The distribution $P(X = x)$ is therefore a combination of voiced and unvoiced distributions. That is,

$$P(X = x) = P(X = x|v)P(v) + P(X = x|v^c)P(v^c) \quad ,$$

where v indicates that the data is voiced, and v^c indicates the complement. It is assumed that the distribution of X for voiced data is Gaussian, giving

$$P(X = x|v) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad ,$$

for some mean μ and variance σ^2. Unvoiced data are assumed to be noise, and will therefore have a greater prediction error than voiced data. Thus, for small enough x, say $x \in [0, x_1]$, $P(X = x|v^c) \approx 0$, so $P(X = x) \approx P(X = x|v)P(v)$. Using this result, the maximum likelihood estimate $\hat{\mu}$ of mean μ can be shown to be given by the first maximum of the density function $\hat{f}(x)$. Variance σ^2 and prior $p_v := P(v)$ can then be fitted numerically, by defining the quadratic error function $E(\sigma, p_v)$ as

$$E(\sigma, p_v) = \int_0^{x_1} \left(P(x) - p_v \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \right)^2 \mathrm{d}x \quad .$$

This quantifies the difference between the actual density function $P(X = x)$, and the small-x model for $P(X = x|v)P(v)$, and can be considered as a continuous-x equivalent of the traditional sum of square residuals objective function. Numerically minimising the error gives both the prior

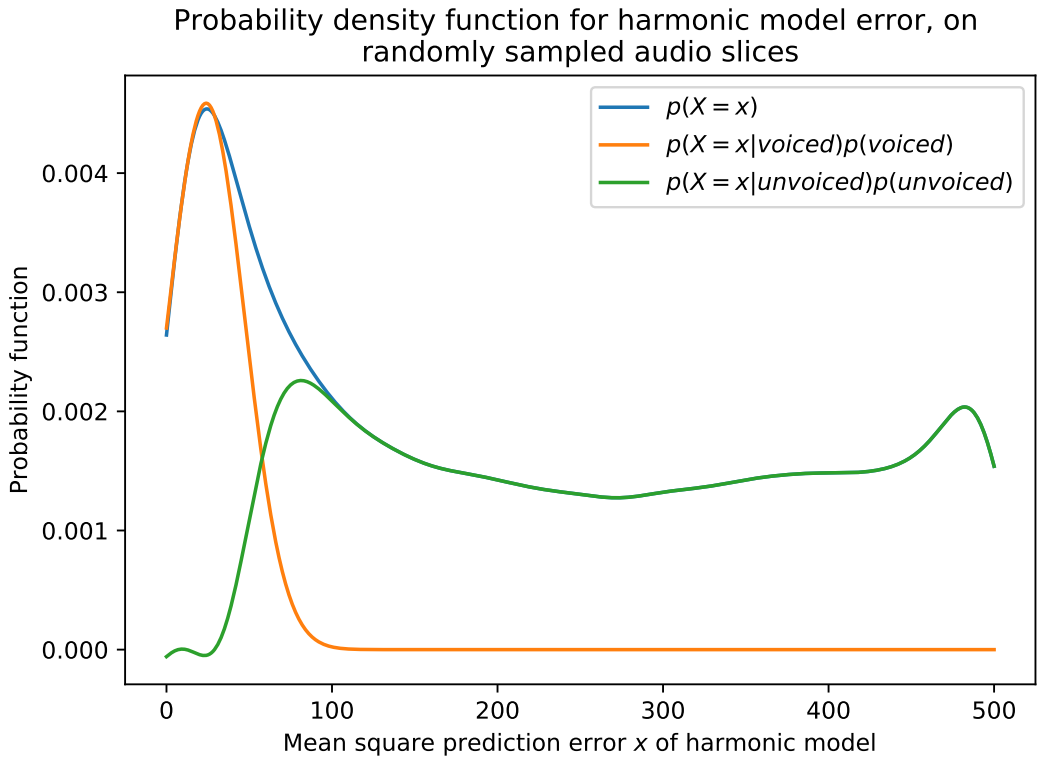


Figure 2.2: Decomposition of the prediction error probability distribution estimate, into (scaled) voiced and unvoiced probability distributions.

$P(v)$, and the fitted conditional probability model $P(X = x|v)$. Figure 2.2 shows a plot of the fitted distributions $P(X = x|v)$, $P(X = x|v^c)$, and $P(X = x)$.

Using these results, evidence can now be conditioned on. Taking the kernel density estimate $\hat{f}(x)$ as an estimate of the probability distribution $P(X = x)$, and taking the prior p_v and conditional model $P(X = x|v)$ as fitted above, the probability of a sample containing voiced data, given the prediction error x, is found as

$$P(v|X = x) = \frac{P(X = x|v)P(v)}{P(X = x)} = \frac{P(X = x|v)p_v}{\hat{f}(x)} \quad .$$

An audio sample is classified as voiced when $P(v|X = x) > 0.5$; otherwise, it is considered as either unvoiced speech, silence, or noisy silence, and is discarded.

The classification method presented here falls under the category of unsupervised learning. As no labelled training data are used, one cannot easily determine the classification accuracy of the method. To validate the classifier, the voiced-detection algorithm is applied to a randomly chosen selection of audio clips. Unvoiced data are removed by the algorithm. By listening to the resulting audio signals, it is found that the classifier always succeeds in removing all unvoiced sections of the input clip. While it is possible that the classifier erroneously removes sections of voiced data, this is not considered to be an issue, as a large amount of voiced data remains in the studied audio clips after processing. For the work presented here, a false negative voiced classification is significantly more desirable than a false positive.

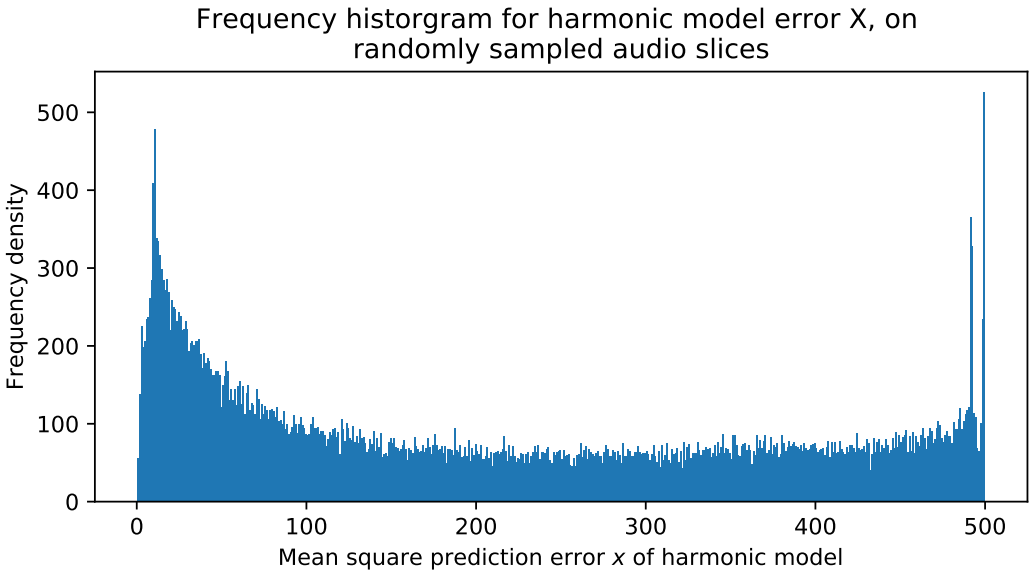


Figure 2.3: Histogram for realisations of the harmonic model prediction error X.

Two points must be discussed when considering this method. Firstly, speech is assumed here to be either voiced or unvoiced. In practice, this is not true. The International Phonetic Alphabet is typically used for transcribing English [50]; the following discussion follows this convention. Vowel sounds such as ɪ ('i' in 'bit'), e ('e' in 'bet'), and æ ('a' in 'bat') contain only voiced speech. Fricatives such as f ('f' in 'frog'), or ʃ ('s' in 'sun') contain only unvoiced speech. Consonants such as ʤ ('j' in 'June') or v ('v' in 'various') can contain a mixture of voiced and unvoiced sounds. The classifier presented here is incapable of separating out sounds into these different types of voiced and unvoiced sounds. Rather, it only considers whether a sound contains a sufficient amount of periodic data to be reasonably considered as voiced. This behaviour is adequate for the work considered here, as any sufficiently voiced sound will contain meaningful information about the dynamics of the speech production system. The model fitting methods presented in this report will be able to average out unvoiced noise from a sufficiently voiced clip of audio data.

Secondly, one must consider the validity of the Gaussian fit. Here, the prediction error is assumed to follow a Gaussian distribution for voiced data. Figure 2.3 shows a histogram of Monte-Carlo realisations for prediction error X. From this, it appears that a Gaussian model may be appropriate for the voiced data. This can be examined theoretically. Let $Y_t = x_t - \sum_{i=1}^{k}[a_i \cos(i\omega_0 t) + b_i \sin(i\omega_0 t)]$ be the harmonic model residual of the time-t audio sample. Assume that for voiced data, the harmonic model captures the majority of the waveform shape. Any non-zero residuals must therefore be primarily a result of noise. Assume that for voiced data, this noise is Gaussian. Thus, $Y_t \sim \mathcal{N}(0, \sigma_Y^2)$, with Y_t independent and identically distributed (i.i.d.). By definition, $X = \frac{1}{N}\sum_{t=1}^{N} Y_t^2$. The transformed variable $Z = \frac{NX}{\sigma_Y}$ is a sum of squared i.i.d. standard normal random variables, meaning that $Z \sim \chi_N^2$. For large N, χ_N^2 can be approximated by a normal distribution [42]. Z, and hence Y, can thus be modelled with a normal distribution. This result demonstrates that it is reasonable to model the prediction errors for voiced data as following a Gaussian distribution. More generally, one can observe that X is the sample mean of a set of i.i.d. random variables. By the central limit theorem, X will approach a Gaussian distribution for large N.

2.2 Embedology

As discussed in section 1.2, evidence suggests that changes in the emotional state of a speaker will produce changes in vocal tract parameters. It is assumed that these changes will cause different dynamics within the speech production system. Nevertheless, it is not immediately obvious how to study these dynamical changes. Here, results from differential topology are used to motivate the concept of reconstructing a state space from audio data. A reconstructed state space is sought, with the requirement that the dynamics within the reconstruction are identical to the dynamics of the speech production system. Hence, the dynamics of the speech production system can be studied from audio recordings. The resulting dynamics can be compared over a range of emotions, and the suitability of different classification features can be examined. This section discusses the theoretical and practical considerations for achieving this.

2.2.1 Attractor Reconstruction

A theoretical analysis of a dynamical system is only possible when the governing equations of the system are known. In experimental work, a model is rarely available. Instead, the system must be investigated from observed data. The typical approach to this problem is to reconstruct a state space from the data, and to perform the desired analysis on this reconstruction. Here, a reconstruction of a state space is defined as being some representation of the recorded data, such that the dynamics of the representation are equivalent to those of the original system. The dynamics are said to be equivalent if there exists a homeomorphism between the phase portraits of the systems. That is, there exists a smooth, invertible transformation between the dynamics of the original system, and the dynamics of the reconstruction. Topologically invariant properties such as fractal dimensions, Lyapunov exponents, and equilibrium eigenvalues are captured by the reconstruction [7]. Typically, the dynamics of a system will lie either on an m-dimensional manifold $\mathcal{M}$, or on a fractal attractor. The analysis covered here assumes trajectories lie on a manifold; nevertheless, Sauer et al. demonstrate that the results discussed here extend to fractal attractors [38].

Let $\varphi : \mathbb{R}^m \mapsto \mathbb{R}$ be a scalar-valued observation function on the manifold $\mathcal{M}$. While the dynamics of interest lie on $\mathcal{M}$, an experimenter will typically only have access to observations $\varphi(\mathcal{M})$ of the system. Thus, a method of reconstructing the original system dynamics becomes necessary. Observed data are generally recorded in the form of a one-dimensional time series. An alternative representation of the observed time series data is sought, such that the dynamics of the system are preserved in the representation.

A manifold is embedded in a Euclidean space when every point on the manifold maps to a unique point in the space. Whitney's weak embedding theorem states that an m-dimensional manifold can be embedded in the Euclidean space $\mathbb{R}^{2m+1}$ [51]. (Note that Whitney's embedding theorem provides only an upper bound on the minimum number of spatial dimensions required for an embedding.) This theorem proves useful for the problem of state space reconstruction. Say a set of $2m + 1$ independent signals can be observed from a system. The value of each signal is determined entirely by the state of the system, at any given time. Thus, some map exists between the set of states, and the set of observed signals. Whitney's theorem shows that, as the $2m + 1$ independent signals contain enough information to embed the dynamics of the system, they must reconstruct the state space [39]. The problem of state space reconstruction therefore becomes one of finding $2m + 1$ independent signals.

Takens considers this problem, and provides sufficient conditions for reconstructing an attractor from a scalar observation function [47]. Consider the m-dimensional system $\dot{\mathbf{x}} = \boldsymbol{F}(\mathbf{x})$,

where $\mathbf{x} \in \mathbb{R}^m$ is a state vector, and $\boldsymbol{F} : \mathbb{R}^m \to \mathbb{R}^m$ defines the dynamics of $\mathbf{x}$. Let $\varphi(\mathbf{x})$, $\varphi : \mathbb{R}^m \to \mathbb{R}$ be some generically chosen observation function on the system. Consider a set S of $2m+1$ dimensional vectors, each of the form

$$\mathbf{v}(t) = \left(\varphi\left[\mathbf{x}(t)\right],\ \varphi\left[\mathbf{x}(t-\tau)\right],\ \ldots,\ \varphi\left[\mathbf{x}(t-2m\tau)\right]\right)^T\ .$$

That is, the vector $\mathbf{v}(t)$ contains $2m+1$ time-delayed observations of $\mathbf{x}$, with observations taken at a constant delay time τ. Assume that the dynamics of $\boldsymbol{F}$ lie on a compact attractor, on an m-dimensional manifold. The set S of $2m+1$ dimensional lag vectors is diffeomorphic to this m-dimensional manifold - there exists a smooth, invertible, differentiable transformation between the manifold containing $\boldsymbol{F}$, and the set S of lag vectors. Thus, the dynamics of $\boldsymbol{F}$ are fully reconstructed by the delay vectors $\mathbf{v}(t)$.

Takens' embedding theorem provides a framework for reconstructing dynamics from continuous-time observations of a system. Nevertheless, experimental data are not typically continuous; rather, the recorded data form a discrete time series of observations. To reason about discrete observations on $\boldsymbol{F}$, the reasoning presented by Szalai et al. [46] is followed. Consider again the system $\dot{\mathbf{x}} = \boldsymbol{F}(\mathbf{x})$. Let the solution to this system be the flow map $\Xi_t :\ \mathbf{x}_0 \mapsto \mathbf{x}(t)$. Ξ_t is thus the evolution operator from state $\mathbf{x}_0$ to $\mathbf{x}_t$, and $\mathbf{x}_{i+1} = \Xi_T(\mathbf{x}_i)$ is the time-T evolution of state $\mathbf{x}_i$. Consider some $\mathbf{x}_t = \mathbf{x}(t)$. Now, $\mathbf{x}_i = \Xi_{\tilde{\tau}}\mathbf{x}_{i-\tilde{\tau}}$. The reconstruction vectors $\mathbf{v}_i$ now become

$$\begin{aligned}\mathbf{v}_{i+2m\tilde{\tau}} &= (\phi[\Xi_{2m\tilde{\tau}}\mathbf{x}_i],\ \phi[\Xi_{(2m-1)\tilde{\tau}}\mathbf{x}_i,\ \ldots,\ \phi[\mathbf{x}_i])^T \\ &= \left(\varphi\left[\mathbf{x}_j\right],\ \varphi\left[\mathbf{x}_{j-\tilde{\tau}}\right],\ \ldots,\ \varphi\left[\mathbf{x}_{j-2m\tilde{\tau}}\right]\right)^T\ ,\end{aligned} \tag{2.3}$$

for delay-step $\tilde{\tau}$. Thus, the discrete-sampled delay vectors $\mathbf{v}_i$ exhibit the same dynamics as the continuous-time vectors. Consider the discrete mapping $\mathbf{v}_{i+1} = \boldsymbol{F}(\mathbf{v}_i)$. Any topological feature of the dynamics of Ξ_t will be shared by the discrete map F [46]. Thus, the map $\boldsymbol{F}$ reconstructs the dynamics of the system.

To apply this result to the problem of emotion recognition, delay vectors are constructed from audio recordings. Voiced speech recordings are considered as containing discrete measurements from an observation function on the speech production system. A mapping $\boldsymbol{F} : \mathbf{v}_i \mapsto \mathbf{v}_{i+1}$ is constructed. The dynamics of the mapping $\boldsymbol{F}$ are topologically equivalent to the dynamics of the speech production system. Thus, the dynamics of the vocal tract can be studied explicitly. As discussed, it is conjectured here that the dynamics of the vocal tract will contain sufficient information for classifying the emotions of a speaker.

2.2.2 Embedding Parameters

While Takens' theorem demonstrates the theory required for reconstructing an attractor, it fails to address how this is best achieved in practice. A reconstruction of speech production dynamics first requires the construction of an appropriate set of delay vectors. This, in turn, requires the experimenter to choose values for the embedding dimension d, and the delay time $\tilde{\tau}$. For a system of known dimensionality, this is trivial. Nevertheless, experimental work typically lacks known ground truth - the experimenter rarely has knowledge of the dimensionality or evolution equations of a system. For experimental data, the quality of a reconstruction will depend entirely on the choice of the embedding dimension d and lag time $\tilde{\tau}$ [23]. Thus, careful consideration must be paid to the choice of these values, to ensure the calculated dynamical parameters, such as Lyapunov exponents or equilibrium eigenvalues, accurately represent those of the original system. Here, practical methods are considered for determining these parameters.

2.2.2.1 Delay time

The work of Takens implies that any time delay can be used to reconstruct a state space. Nevertheless, the embedding theorem assumes arbitrarily precise state measurements, which is unrealistic for practical applications. Casdagli et al. consider this problem, and introduce the concepts of redundance and irrelevance [6]. A careful analysis of these concepts helps to provide guidelines for state space reconstruction in cases when the arbitrary precision assumption is relaxed.

Consider choosing a small value for $\tilde{\tau}$. Each delay coordinate will therefore be sampled at a near-by point in time. If $\tilde{\tau}$ is sufficiently small, there will be a strong correlation between the values in each coordinate direction of any given delay vector. The reconstructed dynamics will thus be squeezed along the identity line. In the presence of noise, this may make it impossible to study the dynamics of the system. The strong correlation between coordinate values means that each coordinate contains redundant information; Casdagi et al. therefore refer to this small-$\tilde{\tau}$ phenomenon as 'redundance'.

Consider now choosing a large value for $\tilde{\tau}$. A nonlinear system may exhibit chaotic dynamics; such systems exhibit sensitive dependence to initial conditions, and therefore cannot be predicted accurately over long time scales. The size of the predictive window is determined by the largest Lyapunov exponent. Say the delay time $\tilde{\tau}$ exceeds this predictive timescale. The dynamics exhibited in each reconstruction vector coordinate become causally disconnected from the dynamics in the other coordinates. In this scenario, simple topologies can look overly complex, as a result of the dynamics in each coordinate direction bearing little relevance to those in the other coordinate directions. Casdagli et al. refer to this phenomenon as 'irrelevance'.

The optimal value of $\tilde{\tau}$ must balance redundance and irrelevance. Several solutions have been proposed for achieving this [21]. Redundance can be minimised through consideration of the autocorrelation function of the scalar-valued observation signal. The autocorrelation of a signal is defined as the correlation between pairwise-lagged observations. Given the time series $\{X_i\} = (x_0, x_1, x_2, \dots)$, the autocorrelation at lag k is the linear correlation between X_i, and the time-delayed series $\{Y_i\} = (x_k, x_{k+1}, x_{k+2}, \dots)$. Lag time $\tilde{\tau}$ can be chosen as the first zero of the autocorrelation function. This seeks to minimise redundance, by minimising the linear correlation between each coordinate; irrelevance is kept as low as possible by choosing the smallest such $\tilde{\tau}$ to give a zero autocorrelation.

While conceptually simple, this method neglects any nonlinear correlations. Fraser et al. [9] address this issue, by considering instead the general relatedness between the time series $\{X_i\}$ and its k-delayed counterpart $\{Y_i\}$. Mutual information is used to quantify the generalised correlation between $\{X_i\}$ and $\{Y_i\}$. The lag k that minimises the mutual information between the two series is chosen as the delay time $\tilde{\tau}$. In practice, the mutual information function typically has several minima; the first of these is chosen, so as to balance any decrease in redundance against the increase in irrelevance.

Kim et al. recommend using correlation integrals instead of the autocorrelation or mutual information methods [21]. However, Liebert et al. demonstrate that the method of correlation integrals is essentially the same as the mutual information approach, and hence yields the same value of $\tilde{\tau}$ [23]. Correlation integrals have the benefit of requiring fewer data points, however the audio recordings in question contain a sufficiently large number of data points for this to not be an issue.

The work presented here utilises the mutual information method of Fraser et al. [9]. Audio recordings are split into voiced and unvoiced sections, using the method presented in section 2.1. The longest continuous section of voiced speech is retained for analysis. The mutual information is calculated between the voiced audio signal and its lagged counterpart. The result gives mutual information as a function of lag k. The first k that produces a minimum in the mutual information function is chosen as the lag time $\tilde{\tau}$. This always produces a value of $\tilde{\tau} = 19$ samples, independently of the emotion of a given speaker. The value of $\tilde{\tau} = 19$ is assumed for all further analysis.

2.2.2.2 Embedding Dimension

Takens' theorem states that the topology of an attractor, whose dynamics lie on an m-dimensional manifold, can be fully reconstructed by $2m + 1$-dimensional lag vectors [47]. Nevertheless, the dimensionality m of an observed dynamical system is typically unknown for experimental data, and must therefore be determined numerically. A naïve approach is to note that the topology remains unfolded for all embedding dimensions $d > 2m$. Thus, choosing an arbitrarily large value for d will necessarily embed the dynamics. Although theoretically valid, this method is unsatisfactory in practice, as it is computationally inefficient to process large quantities of high-dimensional data. Furthermore, any extracted dynamical features will typically be of high dimensionality, too. This produces a sparse feature set, which is difficult to classify from. Finally, each additional dimension will add more noisy data to the reconstruction. Larger embedding dimensions therefore contain more uncertainty in the reconstruction. As a result, it is desirable to choose the lowest possible embedding dimension for reconstructing an attractor.

One method for achieving this is to compute an invariant (such as fractal dimension or Lyapunov exponents) on the attractor, across a range of embedding dimensions [20]. The calculated invariant will remain constant for all embedding dimensions that reconstruct the dynamics. However, this approach is computationally expensive. Kennel et al. address this by introducing the method of false nearest neighbours, to more efficiently find the embedding dimension d [20]. Consider a set of delay vectors. The neighbours of a given point in the set are termed true neighbours if they lie within a neighbourhood of each other as a result of the topology of the system dynamics; they are termed false neighbours if the points exist within a neighbourhood of each other only as a result of projecting the attractor down to a smaller geometric space. The attractor is fully unfolded if and only if no false neighbours exist. If the relative increase in Euclidean distance between two points exceeds some distance tolerance, when going from embedding dimension d to embedding dimension $d + 1$, the points are considered to be false neighbours in dimension d. Let R_k be the Euclidean distance between a pair of nearest-neighbour points, in embedding dimension k. The pair are false neighbours in dimension d, if

$$\frac{R_{d+1}^2 - R_d^2}{R_d^2} > R_{tol}^2 \ ,$$

for some tolerance R_{tol}. The state space is considered fully reconstructed when no false neighbours exist.

The method of false neighbours is regularly used for state space reconstruction. Nevertheless, it is not without issues. Data become more sparse as the dimensionality increases; as a result, it becomes less meaningful to find pairs of nearest neighbours. Consequently, the algorithm loses effectiveness for large embedding dimensions. Furthermore, the algorithm requires a choice of tolerance R_{tol}. Rhodes et al. derive a relationship between the gradient of the observation

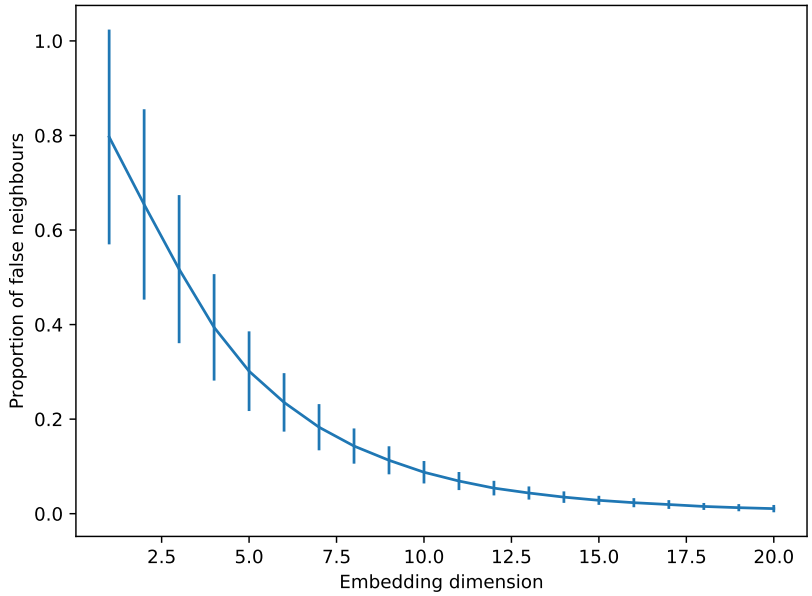


Figure 2.4: Proportion of false neighbours as a function of embedding dimension, averaged over approximately 1400 tests, taking $R_{tol} = 2$. Error bars show the standard deviation of the calculated value.

function and the proper choice of threshold R_{tol} [34]. As the observation function is typically unknown for experimental systems, the analytical method for choosing R_{tol} becomes impossible to apply. Instead, the threshold must be chosen subjectively. The validity of the false neighbours results depends entirely on whether the choice of threshold is reasonable, which cannot be tested explicitly. Finally, the method of false neighbours is not robust against noisy data. Rhodes et al. observe that for noisy data, the proportion of false neighbours in any given dimension increases with the number of data points [34]. An adjustment to the algorithm is proposed to help deal with noisy data, however this requires subjective choices for an additional two parameters. The quality of the reconstruction is once again determined by the quality of choice of these parameters, which is impossible to assess.

Note that simply filtering the data to remove noise is not a desirable approach - Badii et al. demonstrate that filtering a chaotic signal causes changes to both the Lyapunov exponents and the fractal dimension of the reconstructed attractor [2]. The aim of attractor reconstruction is to study parameters such as these; it is therefore undesirable to artificially alter the parameters through filtering.

These issues make the method of false neighbours difficult to apply to the data considered here. Figure 2.4 shows the results of applying the false nearest neighbours algorithm, as implemented by Hegger er al. [12], to sections of voiced audio data. Approximately 1400 audio sections are considered, each with a typical length of around 7000 samples. The threshold R_{tol} is set to two. Error bars show the standard deviation of the proportion of false neighbours.

The plot suggests that an embedding dimension somewhere between 15 and 20 is required to unfold the dynamics of the speech production system. This result should be treated with caution - low-dimensional vocal chord models are known to synthesise natural-sounding speech [5], suggesting that the speech production dynamics can be modelled effectively with a low-dimensional system. This in turn implies that the high dimensionality found by the false

neighbours method may actually be an artefact of noise in the signal. While the proportion of false neighbours will differ for a different choice of R_{tol}, there is no way of knowing what the correct value of R_{tol} is, and hence it is difficult to determine the correct embedding dimension. To overcome this issue, a novel method for calculating embedding dimensions is proposed here. This method is tested first on systems with known embedding dimensions, to validate the efficacy of the method. It is then applied to the audio data, to produce a more trustworthy estimate of the embedding dimension to that achieved by the method of false neighbours.

The false nearest neighbours algorithm asks whether a reconstructed state vector contains enough information to predict the next state vector, based only on properties of the data [34]. The method developed here seeks to reason explicitly about the predictive capabilities of a reconstructed state vector, and uses this to determine the correct embedding dimension. To achieve this, a nonlinear update mapping is fitted to the set of delay vectors, for a range of embedding dimensions. The prediction error of this model is used to quantify the predictive power of the set of delay vectors. The minimum embedding dimension is determined by observing when the model prediction error ceases to change, with increasing numbers of dimensions.

For a given test dimension, the model is fitted as follows. Consider the set of delay vectors $\mathbf{v}_i \in \mathbb{R}^n$, as defined by equation (2.3). Let the jth coordinate entry of $\mathbf{v}_i$ be $x_{i,j}$. Hence, $\mathbf{v}_i = [x_{i,1},\ x_{i,2},\ x_{i,3},\ \ldots,\ x_{i,n}]^T$. Define the order-$k$ monomial vector $\mathbf{v}_i^k$ of delay vector $\mathbf{v}_i$ to be the vector of all order-k monomials over the set of $x_{i,j}$. That is,

$$\begin{aligned}
\mathbf{v}_i^2 &= [x_{i,1}^2 \ \ x_{i,2}^2 \ \ \ldots \ \ x_{i,1}x_{i,2} \ \ x_{i,1}x_{i,3} \ \ \ldots \ \ ]^T, \\
\mathbf{v}_i^3 &= [x_{i,1}^3 \ \ x_{i,2}^3 \ \ \ldots \ \ x_{i,1}^2x_{i,2} \ \ x_{i,1}^2x_{i,3} \ \ \ldots \ \ ]^T, \\
&\ldots
\end{aligned}$$

An order-k polynomial map is sought, such that for some matrix A,

$$\mathbf{v}_{i+1} = A \begin{pmatrix} 1 \\ \mathbf{v}_i \\ \mathbf{v}_i^2 \\ \vdots \\ \mathbf{v}_i^k \end{pmatrix} = A\mathrm{P}_k(\mathbf{v}_i) \ ,$$

where $\mathrm{P}_k(\mathbf{v}_i)$ is the vector of monomials up to order k, for state vector $\mathbf{v}_i$. Define the matrices V_i and V_{i+1} as

$$\begin{aligned}
V_i &= [\mathrm{P}_k(\mathbf{v}_1) \mid \mathrm{P}_k(\mathbf{v}_2) \mid \mathrm{P}_k(\mathbf{v}_3) \mid \ \ldots \ \mid \mathrm{P}_k(\mathbf{v}_{N-1})]^T \ , \\
V_{i+1} &= [\mathbf{v}_2 \mid \mathbf{v}_3 \mid \mathbf{v}_4 \mid \ \ldots \ \mid \mathbf{v}_N] \ .
\end{aligned}$$

Now, $V_{i+1} = AV_i$. By means of a pseudo-inverse argument, an estimate $\hat{A}$ of A can be calculated as

$$\hat{A} = V_{i+1}V_i^T(V_iV_i^T)^{-1} \ , \tag{2.4}$$

giving a mean square relative prediction error E on the fitted data as

$$E = \frac{1}{N-1}\sum_{i=1}^{N-1} \frac{\left\|\mathbf{v}_{i+1} - \hat{A}\mathrm{P}(\mathbf{v}_i)\right\|_2^2}{\left\|\mathbf{v}_{i+1}\right\|_2^2} \ .$$

Note that any vectors satisfying $\left\|\mathbf{v}_i\right\|_2 = 0$ must be removed before completing this calculation.

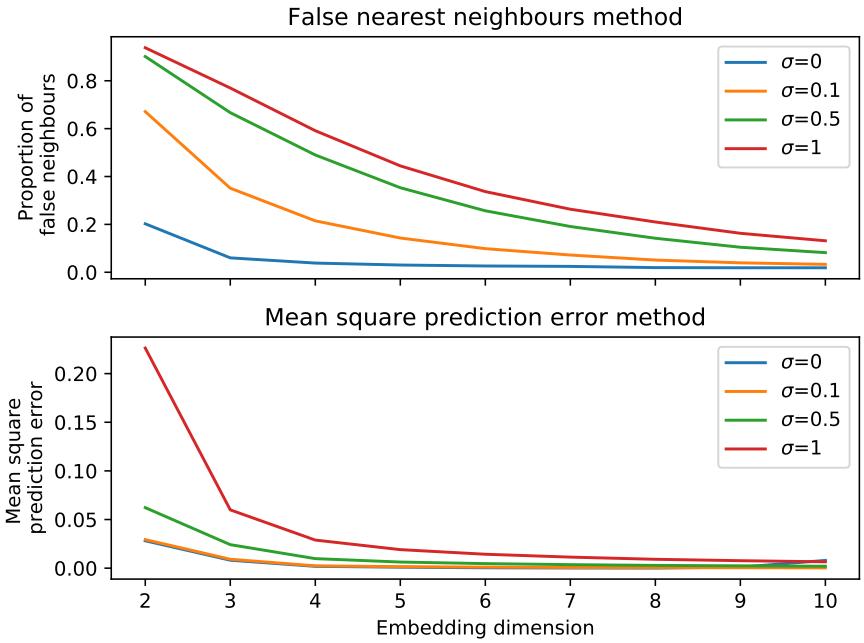


Figure 2.5: Proportion of false neighbours, and nonlinear map prediction error, for a noise-corrupted Rössler attractor [36]. For Rössler system state vector $\mathbf{x}$, the observation function is given by $\mathbf{x} \cdot \hat{\mathbf{i}} + \epsilon_\sigma$, where $\epsilon_\sigma \sim \mathcal{N}(0, \sigma^2)$.

For a least-squares-optimal estimate $\hat{A}$, the prediction error E quantifies a combination of the noise in the system, and how accurately the delay vector update map $\boldsymbol{F} : \mathbb{R}^n \mapsto \mathbb{R}^n$, $\boldsymbol{F}(\mathbf{v}) = A\mathrm{P}_k(\mathbf{v})$ can be estimated from the data. For a stochastic system containing only independent and identically distributed noise terms, the error E will be constant, and independent of the embedding dimension d. As the contribution of noise to the prediction error is independent of the embedding dimension, any change in the prediction error with dimensionality must arise as a result of a change in the predictive power of the delay vectors. The predictive power, and hence the prediction error, will cease to change when the system reaches the minimum embedding dimension. The vectors contain all necessary information to determine the state of the system at an arbitrary time in the future. This predictive power remains for all dimensions greater than the embedding dimension, but ceases to exist below it.

The whole algorithm is as follows:

- construct a set of delay vectors, for a given test dimension d;
- fit a nonlinear update map $\mathbf{v}_{i+1} = A\mathrm{P}_k(\mathbf{v}_i)$;
- calculate the mean square relative prediction error of the map;
- increment the test dimension d;
- repeat until the prediction error ceases to change when incrementing d;
- d is now the minimum embedding dimension of the system.

For noise-free dynamics on an arbitrarily smooth manifold, Taylor's theorem guarantees that the system dynamics can be reconstructed to an arbitrarily high degree of accuracy by the polynomial map $\boldsymbol{F}(\mathbf{v}) = A\mathrm{P}_k(\mathbf{v})$. This result arises from the fact that any arbitrarily smooth

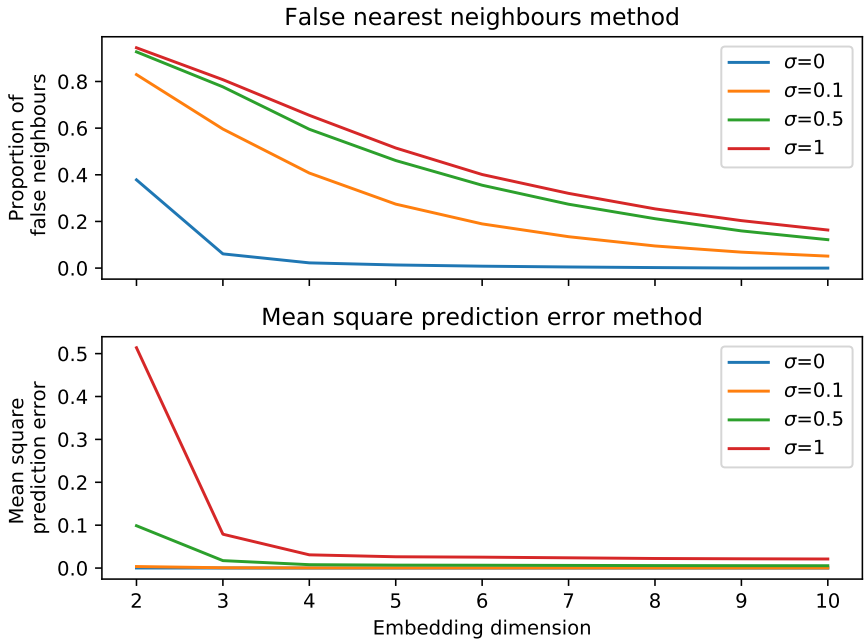


Figure 2.6: Proportion of false neighbours, and update map prediction error, for a noise-corrupted Lorenz attractor [25]. For Lorenz system state vector $\mathbf{x}$, the observation function is given by $\mathbf{x} \cdot \hat{\mathbf{i}} + \epsilon_\sigma$, where $\epsilon_\sigma \sim \mathcal{N}(0, \sigma^2)$. $R_{tol} = 2$.

manifold can be expressed as a sufficiently high-ordered Taylor series. Nevertheless, accurately fitting such a map would require an infinite amount of data. To minimise the number of fitted parameters, and hence required data points, a low-order map is desired. A cubic-polynomial system is the simplest system to generically exhibit periodic orbits. It also contains a sufficiently rich set of dynamics to exhibit all codimension-1 bifurcations. The work presented in this section therefore assumes the third-order map $\boldsymbol{F}(\mathbf{v}) = A\mathrm{P}_3(\mathbf{v})$.

To test the effectiveness of the algorithm, the method of prediction errors is compared against the method of false neighbours, for systems with known embedding dimensions, under artificial noise. Figure 2.5 shows how the proportion of false neighbours and the mean square relative prediction error E changes as a function of embedding dimension, when reconstructing the Rössler attractor [36] from observations of its x-coordinate. Experimental noise is replicated by simulating the system as is, then adding normally distributed white noise terms $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ afterwards. The method of false neighbours fails to find an appropriate embedding dimension for even the smallest amount of noise. The prediction error method developed here succeeds in finding the correct embedding dimension in the zero-noise and low-noise cases. Even for large noise, the prediction error method succeeds in producing a reasonable embedding dimension, suggesting either three or four dimensions are required to embed the system. The false neighbours method fails to produce any reasonable estimate of embedding dimension, for any of the noise cases.

Figure 2.6 shows the same methodology applied to the Lorenz system [25]. The method of false neighbours again fails to produce a reasonable embedding dimension for even the smallest noise case. By comparison, the method of errors finds the correct embedding dimension for even the largest noise case. Nevertheless, the method of errors struggles on the low-noise data. Figure 2.7 shows a reconstruction of the attractor from delay vectors, and a simulation of the fitted polynomial map $A\mathrm{P}_3(\mathbf{v})$. The dynamics of the fitted map are visually very similar to

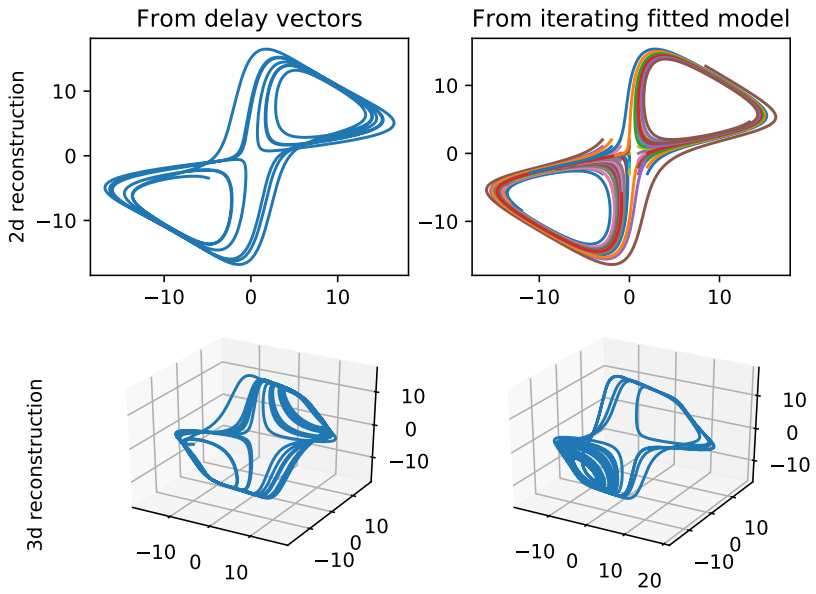


Figure 2.7: Reconstruction of the Lorenz system [25] from delay vectors, and from iterating the fitted polynomial map $A\mathrm{P}_3(\mathbf{v})$, for two and three embedding dimensions. The two-dimensional case produces a geometrically similar polynomial map to the delay vector reconstruction, whereas the three-dimensional case produces a topologically equivalent map.

those of the delay vectors. However, the map dynamics exhibit a different topology to that of the delay vectors. The method of false neighbours considers topology explicitly, and therefore is able to determine that the reconstruction is not an embedding. Conversely, the method of errors does not consider topology, and hence produces a system with geometrically similar but topologically different dynamics to that of the true reconstruction.

To understand why this failure mode occurs, one must consider the fractal nature of the system in question. McGuinness seeks to compute a fractal dimension for the Lorenz system [26]. The box counting dimension is estimated as 1.98 ± 0.02. The same paper notes that the Kaplan-Yorke conjecture [19] implies a fractal dimension of 2.06. Clearly, the Lorenz system has a fractal dimension close to 2. It is therefore unsurprising that the method of errors recommends an embedding dimension of two. The unfolded trajectories lie on a nearly-two-dimensional attractor, meaning little extra information is gained by adding a third delay coordinate. As the dynamics can nearly be unfolded in a two-dimensional embedding space, the method of errors is able to produce a low-error fit in a space that does not fully unfold the dynamics. The method hence exhibits a failure mode when the fractal nature of an attractor allows for the creation of a flow map with a near-identical vector field to the unfolded dynamics, albeit with a different topology.

Figure 2.8 shows the results of applying the method of errors to voiced audio data. Error bars show the standard deviation of the prediction error, as calculated over approximately 60 audio clips. The prediction error drops to a constant value at an embedding dimension of $d = 3$. The error bars show that the uncertainty in the prediction error also becomes constant at $d = 3$, suggesting that the observed results are characteristic of a successful embedding, and not a result of the previously discussed failure mode. This result also agrees with the conventional literature, which states that speech production is a low-dimensional dynamical system [5]. An

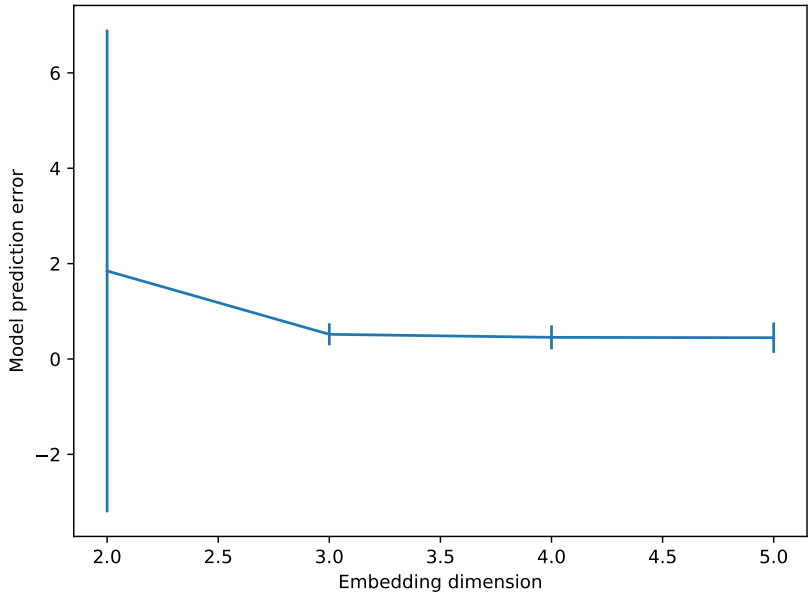


Figure 2.8: Results of the prediction error dimensionality test, averaged over approximately 1400 audio samples. Error bars show the standard deviation of the calculated value.

embedding dimension of $d = 3$ is henceforth assumed.

Here, an embedding dimension for the speech production system was sought. Appropriate experimental methods were discussed. Issues with the method of false neighbours were identified, and a novel alternative was proposed. The method was compared to systems with known ground-truth, to establish the effectiveness of the algorithm. The failure modes of the new method were identified. Finally, the method was applied to experimental data. It is observed that the method of errors derived here typically outperforms the method of false neighbours, when dealing with noisy time series data. In the next section, this embedding dimension result will be applied to help construct a model of vocal chord dynamics.

2.3 Dynamical analysis

Attractor reconstruction generates a state space that exhibits topologically equivalent dynamics to those of the system of interest. Nevertheless, modelling methods are required to be able to reason about these dynamics. The fitting and analysis of a state space evolution operator is considered here. A set of dynamical features are then sought from the fitted models, to use in the classification of speaker emotions.

2.3.1 Linearised dynamics

Modal decomposition describes a dynamical system by the natural vibrations it exhibits [8]. Speech production arises from the vibration of the vocal chords; one may therefore examine the dynamics of speech production using tools from modal decomposition. Here, a speech waveform is projected onto a point in a Hilbert space. The coordinates of this projected point characterise the dynamics of the system. To achieve this, a linear model is fitted to the set of delay vectors. Consider the delay vectors $\mathbf{v}_i$, $\mathbf{v}_{i+1}$, where the vectors lie within a small neighbourhood of the origin. A matrix M is sought, such that $\mathbf{v}_{i+1} = M\mathbf{v}_i$. The eigenvectors of M describe the normal modes of the system; the associated eigenvalues describe the dynamics of these modes.

A strong precedent exists for this method. Schmid presents the method of Dynamic Mode Decomposition (DMD) [41]. DMD extends the linearisation method discussed here, to allow efficient computation on high-dimensional data, and to extract the most significant dynamics of a system. Due to the low dimensionality of the data considered here, DMD is not deemed necessary for this work. Instead, arguments from linear algebra are presented for extracting the system dynamics from reconstructed delay vectors.

Assume the dynamics of the speech production system can be reasonably approximated by a linear model. The work here follows a similar approach to the nonlinear model fitting seen in section 2.2.2.2. For a given audio file, sections of voiced data are extracted using the methodology presented in section 2.1. A set of delay vectors $\mathbf{v}_i$ are created from the largest section of voiced data in a given audio recording. For delay vectors $\mathbf{v}_i$, $i \in \{0, 1, \ldots, n\}$, define the matrices V_i, V_{i+1} as

$$V_i = [\mathbf{v}_0 \mid \mathbf{v}_1 \mid \mathbf{v}_2 \mid \ \ldots \ \mid \mathbf{v}_{n-1}] \ ,$$
$$V_{i+1} = [\mathbf{v}_1 \mid \mathbf{v}_2 \mid \mathbf{v}_3 \mid \ \ldots \ \mid \mathbf{v}_n] \ .$$

Now, $V_{i+1} = MV_i$. The matrix M can be fitted using a pseudo-inverse argument, giving

$$\hat{M} = V_{i+1}V_i^T(V_iV_i^T)^{-1} \ .$$

Note that, unlike in section 2.2.2.2, the update matrix M is square. This allows the eigenvalues and eigenvectors to be calculated. If different emotions produce notable changes in the linearised system dynamics, these changes may show up in the eigenvalues.

Figure 2.9 shows the distributions of the modulus and argument of each eigenvalue, for a linear model fitted to the longest section of voiced data in each RAVDESS database entry. There is no statistically significant difference between the eigenvalue distributions of each emotion. This suggests that the dynamics of the linearised model are insufficient for categorising speaker emotions.

Nevertheless, it is possible that this analysis is hindered by the representation of the data. To address this, the locations of each eigenvalue is plotted on the complex plane, for a range of fitted models. A large number of eigenvalue plots are shown in appendix D, with sections of the unit circle drawn in for scale. Eigenvalues within the unit circle imply a stable equilibrium at the origin. There is just as much difference between eigenvalues for speech with the same emotion, as there is between different emotions. Consequently, the dynamics of a linear state space model can be concluded as contain insufficient information for studying the emotions of a speaker. Either emotions cannot be extracted from the dynamics of the delay vectors, or a more advanced dynamical analysis is required.

An embedding dimension of three is considered here. For a real-valued matrix M in three embedding dimensions, only one oscillatory frequency component can be modelled. The matrix M can have either two or no complex conjugate eigenvalues, and hence, only one oscillatory frequency. Although the model can contain a periodic orbit, such behaviour is not generic. Thus, the linear model is unlikely to have a sufficiently rich set of dynamics to model the vocal chords. Simple nonlinear maps can generically contain periodic orbits. Even a one-dimensional nonlinear map can contain a countable infinity of periodic orbits. Nonlinear models therefore contain significantly more dynamical richness than linear models. It is assumed here that a nonlinear model will be required for a full dynamical analysis. A nonlinear dynamical analysis is proposed in the following section, to investigate this.

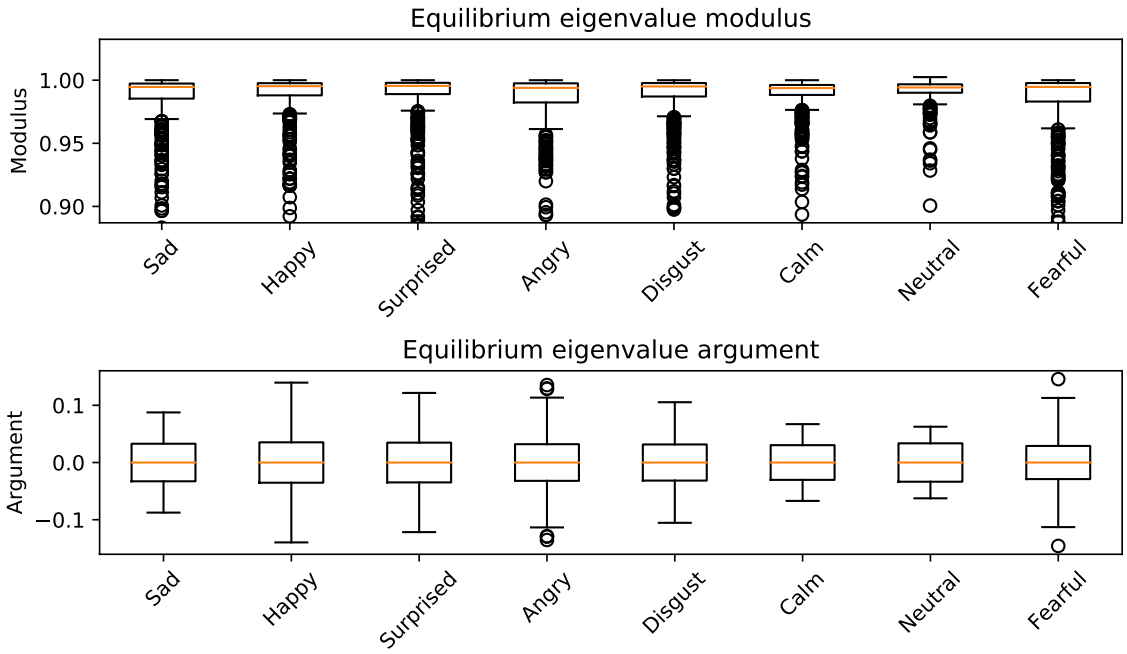


Figure 2.9: Distribution of eigenvalue modulus and arguments for different emotions, as found using a linear trajectory model. Circles indicate points more than twice the interquartile range away from the mean.

2.3.2 Nonlinear model fitting

A linear dynamical model is insufficient for examining speaker emotions. Instead, a nonlinear model must be considered. Such a model is guaranteed fit the data at least as well as the linear model discussed previously. Thus, while more difficult to analyse, a nonlinear model has the potential to extract more information from the system than can be achieved through linearisation alone.

For a limit cycle arising from a Hopf bifurcation, small-amplitude oscillations are approximately sinusoidal, with the frequency given by the imaginary component of the equilibrium eigenvalues [45]. These small-amplitude oscillations can therefore be characterised entirely by the eigenvalues of the relevant system equilibrium. The state space reconstruction method contains some similar frequency information to that extracted through a Fourier transform, however it is hoped that the dynamics-based approach will provide a more readily usable data representation than can be achieved through Fourier analysis.

Consider the nonlinear update map introduced in section 2.2.2.2. The pseudo-inverse fit for $\hat{A}$, as given in equation (2.4), gives a least-squares-optimal fit for $\hat{A}$. Over the set of delay vectors $\mathbf{v}_i$, the error $\left\|\mathbf{v}_{i+1} - A\mathrm{P}_k(\mathbf{v}_i)\right\|_2^2$ is minimised. Thus, the one-step prediction error of any point-update in the solution space is minimised. Such an optimisation is referred to here as a local optimisation. Let $\mathrm{P}_k(\mathbf{v}_i)$ be the vector of all monomials in $\mathbf{v}_i$ up to order k. Consider the update map $\boldsymbol{F} : \mathbb{R}^n \mapsto \mathbb{R}^n$, $\boldsymbol{F} = A\mathrm{P}_k(\mathbf{v}_i)$, for some matrix A. The least-squares local optimisation approach is a poor choice of methods for fitting $\boldsymbol{F}$. Instead, the error should be minimised over whole trajectories. For some initial condition $\mathbf{v}_0$, the trajectory-error $\left\|\mathbf{v}_t - \boldsymbol{F}^t(\mathbf{v}_0)\right\|_2^2$ should be minimised, for some arbitrarily large t. The resulting model will thus be optimised over the entire solution space. Such an optimisation is referred to here as a global optimisation.

It may be desirable to increase the local error in some subset of the solution space, if this produces a decrease in the global error over the set of trajectories. Such a scenario is likely to occur when a particular area of the state space contains highly noise-contaminated data. In this case, the locally optimal and globally optimal models for $\boldsymbol{F}$ will differ from each other. Here, an optimisation procedure is proposed, to approximate a globally optimal fit for $\boldsymbol{F}$. The nonlinear nature of the model means no elegant method exists for writing down the nth step map $\boldsymbol{F}^n(\mathbf{v}_0)$. As a result, a trajectory-optimal model cannot be easily fitted. Instead, an initial model is fitted and recursively improved, so as to approach global optimality.

To achieve this, successively better approximations of A are bootstrapped, to compute a new value of A that is closer to global optimality. First, a locally optimal approximation of A is found, using equation (2.4). This forms a one-step-optimal mapping A, denoted A_1. Next, the ith mapping A_i is used as an approximation of the $i+1$'th mapping A_{i+1}. This approximation is then improved through gradient descent.

The globally optimal map $\boldsymbol{F}$ can be found by minimising the sum of discounted future errors. Let E be the error function for map $\boldsymbol{F}$. For discount factor $\lambda > 0$, the error E is defined as

$$E = \sum_{i=1}^{N} \lambda^i \left\| \boldsymbol{F}^i(\mathbf{v}_0) - \mathbf{v}_i \right\|_2^2 = \sum_{i=1}^{N} \lambda^i \varepsilon_i \ , \tag{2.5}$$

where $\varepsilon_i = \left\| \boldsymbol{F}^i(\mathbf{v}_0) - \mathbf{v}_i \right\|_2^2$ is the error accumulated over a length-i trajectory. Let $A = [a]_{r,j}$; let $\hat{\mathbf{v}}_i = F^i(\mathbf{v}_0)$ be the approximation of $\mathbf{v}_i$ found by iterating the map $\mathbf{v}_{i+1} = \boldsymbol{F}(\mathbf{v}_i)$ from initial condition $\mathbf{v}_0$; let $\hat{v}_{r,i}$ be the rth row of the column vector $\hat{\mathbf{v}}_i$. The gradient of E with respect to $a_{r,j}$ is sought. Consider first the gradient of E with respect to the ith step error ε_i. By the chain rule,

$$\frac{\partial \varepsilon_i}{\partial a_{r,j}} = \frac{\partial \varepsilon_i}{\partial \hat{v}_{r,i}} \frac{\partial \hat{v}_{r,i}}{\partial a_{r,j}} \ .$$

Let $\mathrm{P}_k^n(\mathbf{v}_i)$ be the nth entry of the monomial vector $\mathrm{P}_k(\mathbf{v}_i)$. By definition of $\boldsymbol{F}$,

$$\hat{\mathbf{v}}_{i+1} = \begin{pmatrix} \hat{v}_{1,i+1} \\ \hat{v}_{2,i+1} \\ \hat{v}_{3,i+1} \\ \cdots \end{pmatrix} = \begin{pmatrix} \sum_j a_{1,j} \mathrm{P}_k^j(\mathbf{v}_i) \\ \sum_j a_{2,j} \mathrm{P}_k^j(\mathbf{v}_i) \\ \sum_j a_{3,j} \mathrm{P}_k^j(\mathbf{v}_i) \\ \cdots \end{pmatrix} \ .$$

From this, it follows that

$$\frac{\partial \hat{v}_{r,i}}{\partial a_{r,j}} = \mathrm{P}_k^j(\mathbf{v}_{i-1}) \ .$$

Furthermore, $\varepsilon_i = \sum_r (\hat{v}_{r,i} - v_{r,i})^2$, so

$$\frac{\partial \varepsilon_i}{\partial \hat{v}_{r,i}} = 2(\hat{v}_{r,i} - v_{r,i}) \ .$$

These are combined to get

$$\frac{\partial \varepsilon_i}{\partial a_{r,j}} = \frac{\partial \varepsilon_i}{\partial \hat{v}_{r,i}} \frac{\partial \hat{v}_{r,i}}{\partial a_{r,j}} = 2(\hat{v}_{r,i} - v_{r,i}) \mathrm{P}_k^j(\mathbf{v}_{i-1}) \ .$$

Finally, note that $E = \sum_{i=1}^{N} \lambda^i \varepsilon^i$. Thus,

$$\frac{\partial E}{\partial a_{r,j}} = \sum_{i=1}^{N} \lambda^i \frac{\partial \varepsilon_i}{\partial a_{r,j}} = \sum_{i=1}^{N} 2\lambda^i (\hat{v}_{r,i} - v_{r_i}) \mathrm{P}_k^j(\mathbf{v}_{i-1}) \ ,$$

giving the desired error gradient.

To utilise this result, an expression for the error gradient E in terms of the entire matrix A is defined. Let the gradient matrix G of the ith-step error be defined as

$$G = \frac{\partial \varepsilon_i}{\partial A} := \left[\frac{\partial \varepsilon_i}{\partial a_{r,j}} \right]_{r,j} . \tag{2.6}$$

That is, the r, jth entry of the gradient matrix gives the gradient of ε_i with respect to the r, jth entry of A. From this, the optimisation algorithm is defined as follows.

- Solve equation (2.4) for a locally optimal estimate A_1.
- Starting with a trajectory length of $n = 2$, and initially approximating A_{n+1} with A_n,
  - Choose a randomly selected initial condition $\mathbf{v}_0$ from the set of delay vectors.
  - Simulate a trajectory $\{\mathbf{v}_0, \hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \ldots, \hat{\mathbf{v}}_n\}$ by iterating the currently fitted map $\hat{\mathbf{v}}_{i+1} = \boldsymbol{F}(\hat{\mathbf{v}}_i)$; record the monomial vectors $\mathrm{P}_k(\mathbf{v}_i)$.
  - Calculate the ith step error vector $\hat{\mathbf{v}}_i - \mathbf{v}_i$, for $i \in \{1, 2, \ldots, n\}$.
  - Use the ith monomial vector and error vector to form the ith error matrix, as defined in equation (2.6).
  - For some learning rate η, apply a gradient descent step to obtain the updated matrix A'_n, such that
$$A_n \mapsto A'_n = A_n - \eta \sum_{i=1}^{N} \lambda^i \frac{\partial \varepsilon_i}{\partial A_n} .$$
  - Repeat for the desired number of epochs.
- Increase the trajectory length n by one.
- Repeat until the desired trajectory length has been optimised for.

This algorithm bootstraps on the ith-step-optimal estimate of A, by using it to approximate the $i+1$th-step-optimal A. This approximation is then refined using gradient descent. By choosing a new, random initial condition for each gradient descent step, the model is prevented from over-fitting to a local sub-trajectory. As the trajectory length n becomes large, the optimised estimate of A will approach global optimality.

By definition, nearby trajectories in a chaotic system diverge exponentially fast, on average. An error $\left\|\hat{\mathbf{v}}_i - \mathbf{v}_i\right\|_2$ in the ith approximation $\hat{\mathbf{v}}_i$ can be considered as a perturbation from the actual trajectory location $\mathbf{v}_i$. For a chaotic system, this perturbation, and hence the trajectory error, will grow exponentially fast. This causes the computed gradients to rapidly explode, as the trajectory size n increases. To counter this effect, a small discount factor λ can be chosen. This gives less weight to points in the trajectory that, as a result of chaos in the system, are difficult to fit accurately. Furthermore, the algorithm should be restricted to small trajectory lengths, to reduce the aforementioned gradient explosion problem. Finally, gradient clipping [29] can be used to restrict the maximum allowed size for a gradient descent step. The author recommends a small learning rate, of order 1×10^{-7}, a maximum trajectory length of up to $n = 10$, clipping gradients to $a_{r,j} \in [-2, 2]$, repeating the gradient descent steps for upward of 10,000 epochs, and using a fifth-order polynomial map. Note that these requirements make the algorithm computationally expensive.

Appendix F shows plots of iterated trajectories, under both the locally and globally optimised nonlinear models, for a range of audio recordings. Also plotted is the delay vector trajectory to which the models are fitted. The first delay vector is taken as an initial condition for both of the update maps. The maps are fitted, then iterated from this initial condition, to produce simulated trajectories. By visual inspection, the trajectories of the locally optimal model are seen to match the training data more often than the globally optimal trajectories do. While several plots show the simulated trajectories to differ greatly from the training data, it is possible that a different initial condition would produce a trajectory that matches the data better. Nevertheless, a large number of simulated trajectories cannot be displayed neatly, so only a single initial condition is iterated for the presented plots.

2.3.3 Finding fixed points

The equilibria of a dynamical system contain important information about the system dynamics. For small-amplitude limit cycles, the behaviour of the associated oscillations can be characterised by the eigenvalues of the equilibrium from which the limit cycle appeared. The fixed points of the fitted locally optimal model $\boldsymbol{F} = A\mathrm{P}_5$ are considered here, for use as features in emotion classification. A fixed point is defined as some point $\mathbf{v}^*$, such that $\mathbf{v}^* = A\mathrm{P}_5(\mathbf{v}^*)$. While all equilibria are fixed points, not all fixed points are equilibria. Non-equilibrium fixed points may arise when the frequency of a periodic orbit exactly matches the data sampling frequency. Nevertheless, it is assumed that this behaviour is unlikely; the two terms are therefore used interchangeably here. The nonlinear map $\boldsymbol{F} = A\mathrm{P}_5$ can contain a large number of equilibria; here, an algorithm is presented to extract any fixed points of the model that are likely to be meaningful.

The accuracy of the fitted map $\boldsymbol{F}$ can only be guaranteed for neighbourhoods that are well-represented by the data to which the map is fitted. As a result of extrapolation errors, fixed points can arise outside the range of the available data; it cannot be guaranteed that these fixed points are representative of the system dynamics. Consequently, a two-step algorithm is proposed, to identify fixed points, and remove those that are not well-represented by the data.

Consider a set of delay vectors $\mathbf{v}_i$. A set of intervals $I_j = [l_j, u_j]$ exist, such that the jth component of all delay vectors $\mathbf{v}_i$ lie in the interval I_j. Hence, all delay vectors lie within some hypercube $\mathcal{Q}$, with the limits of $\mathcal{Q}$ in the jth coordinate direction being defined by the interval I_j. As the hypercube contains all the data available for fitting, all well-fitted equilibria must lie within a neighbourhood of $\mathcal{Q}$. To exploit this fact, a large set of points are chosen at random. These points are drawn from a uniform distribution on $\mathcal{Q}$. Any fixed point of $\boldsymbol{F}$ must satisfy the objective function $e(\mathbf{v}) = \boldsymbol{F}(\mathbf{v}) - \mathbf{v}$. Thus, a root finding algorithm is used to solve $e(\mathbf{v})$ for the fixed points $\mathbf{v}^*$ of the nonlinear map. The randomly selected points are used as initial guesses for the root finding procedure. For a sufficiently dense set of initial conditions, a root finding algorithm will always start with an initial guess arbitrarily close to any given equilibrium within $\mathcal{Q}$.

Multiple initial guesses may exist within the neighbourhood of a given equilibrium. Consequently, the same equilibrium may be found multiple times. To address this, a candidate equilibrium is termed ε-unique, if and only if no other candidate equilibria lie within a ball of radius ε, centered at the candidate equilibrium of interest. A set of candidate equilibria is said to be ε-unique if and only if no two equilibria within the set lie within a Euclidean distance of ε from each other. An efficient algorithm for finding an ε-unique set of equilibria is presented in appendix B.

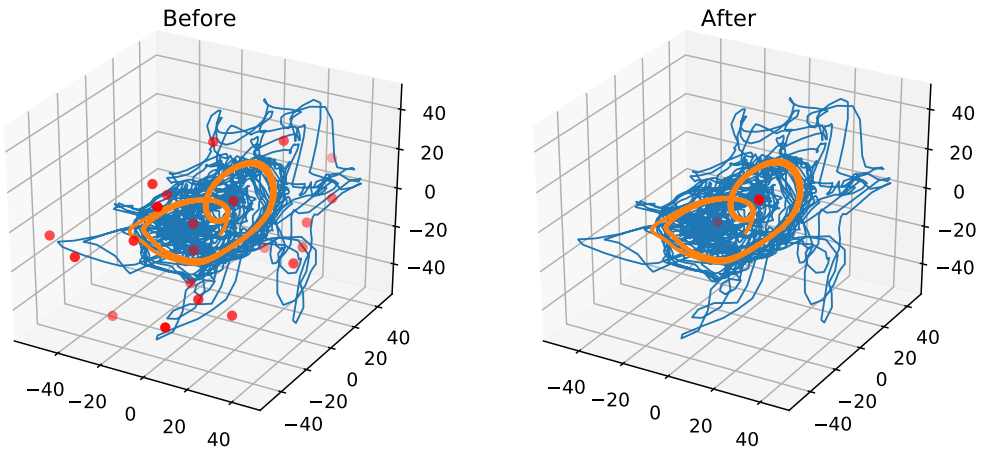


Figure 2.10: Before and after the removal of spurious map equilibria, for a polynomial map fitted to a trajectory of delay vectors. The trajectory of the delay vectors are shown in blue; a simulation of the fitted model is shown in orange; equilibria are shown in red.

Given a set of ε-unique equilibria, one must now decide which equilibria are well-represented by the data, and which are not. It is assumed that any equilibrium sufficiently surrounded by the fitting data will be trustworthy. This assumption is exploited when determining the trustworthiness of an equilibrium.

Consider some equilibrium $\mathbf{v}^*$. One wishes to determine if $\mathbf{v}^*$ is likely to be of interest when studying the dynamics of the system. Only equilibria that are trustworthy are deemed dynamically interesting. To determine if the equilibrium is trustworthy, one can consider a ball around the equilibrium, with a radius chosen so as to include some fixed percentage of the training data. The equilibrium is likely to be trustworthy if it is evenly surrounded by data points. To quantify this, the mean-field location of all data points within the ball is considered. If the data points are spread symmetrically across the ball, and hence, evenly surround the equilibrium, the mean location of the data points will lie at the center of the ball - at the equilibrium. Conversely, if the equilibrium lies outside the range of the data, and hence is untrustworthy, the data points will be clustered towards one side of the ball. Consequently, the mean-field location of the data points will lie away from the equilibrium. The degree of symmetry in the distribution of nearby data points can therefore be quantified by the distance between the equilibrium, and the mean-field location of all points within some ball centered at the equilibrium. An equilibrium is discarded as untrustworthy if this distance is large, relative to the size of the ball.

The equilibrium-searching algorithm can be summarised as follows.

- Draw a large number of initial guesses at random, from the smallest hypercube containing the training data.
- Run a root-finding algorithm such as Newton's method on each of the initial conditions, to form a set of candidate equilibria.
- Use an ε-uniqueness algorithm to extract an ε-unique subset of candidate equilibria.
- For each equilibrium within this subset, find the nearest n data points to the equilibrium.
- Calculate the mean location of these data points.

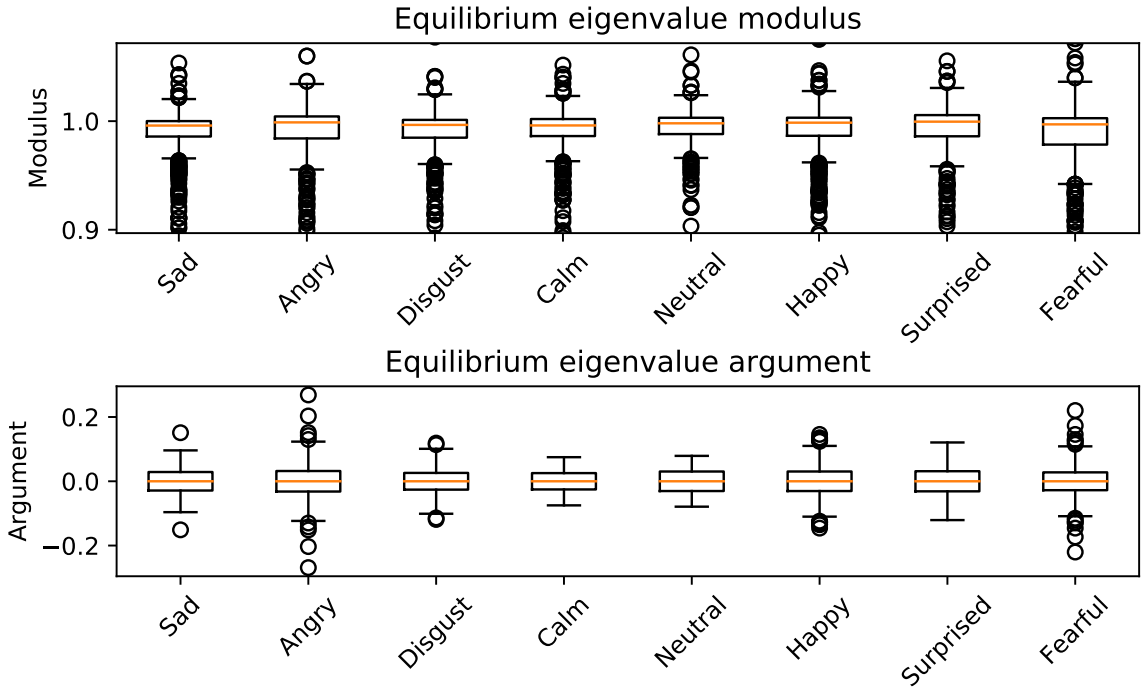


Figure 2.11: Distribution of equilibrium eigenvalue modulus and arguments for different emotions, as calculated from the trustworthy equilibria of 1400 voiced audio recordings. Circles indicate points more than twice the interquartile range away from the mean.

- If the distance from the equilibrium to the mean location is large, relative to the size of the ball, reject the equilibrium as untrustworthy. Otherwise, the equilibrium is accepted as dynamically interesting.

Figure 2.10 shows the an example of equilibrium locations before and after the application of the trustworthiness algorithm. The algorithm successfully removes all the spurious equilibria, and retains the two realistic equilibria.

2.4 Feature extraction

While the nonlinear model fitted in section 2.3.2 contains information about the dynamics of the vocal chords, it is not immediately obvious how to reason about these dynamics. One cannot simply use the parameters of a fitted model for emotion classification. There are 56 monomials with a degree of at most five, in three dimensions. The associated nonlinear model $\boldsymbol{F}(\mathbf{v}) = A\mathrm{P}_5(\mathbf{v})$ therefore contains 168 parameters in total. This is too high of a dimensionality to classify from. Instead, a lower-dimensional feature set must be extracted from the reconstructed attractor. This section considers possible dynamical features to achieve this.

2.4.1 Equilibria

The simulated trajectories shown in appendix F demonstrate that the nonlinear model often produces limit cycles. For small amplitude oscillations, one can characterise a limit cycle entirely from the eigenvalues of the linearised equilibrium. Thus, the fitted nonlinear model is linearised about the previously found equilibria. The eigenvalues of this linearisation are treated as dynamical features.

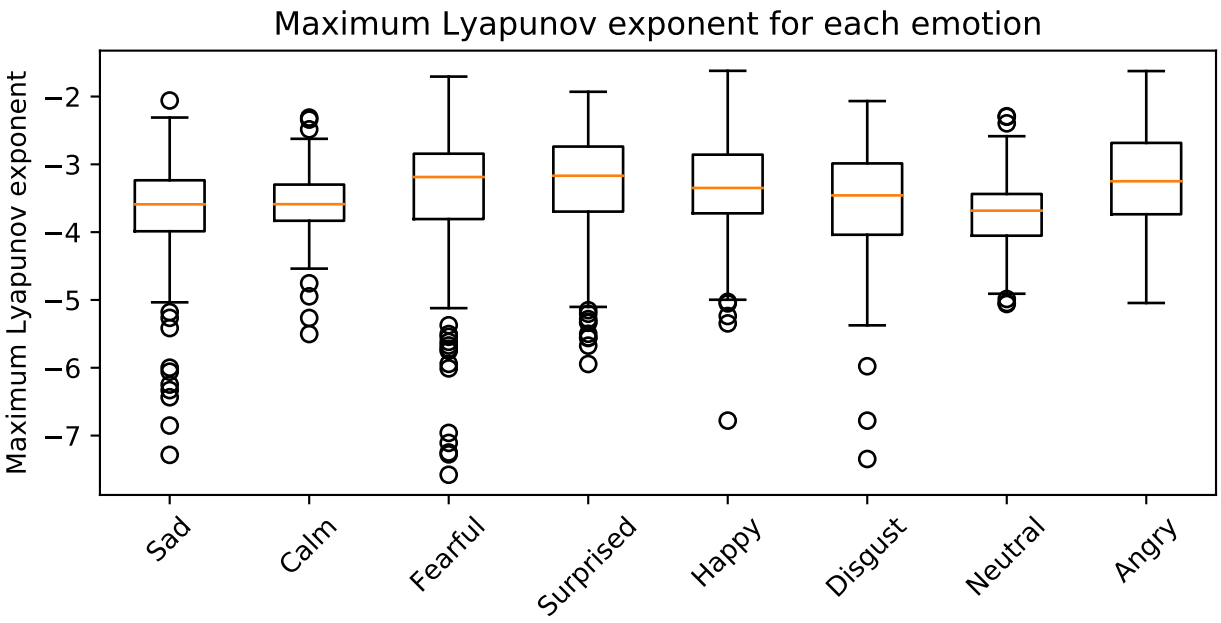


Figure 2.12: Maximum Lyapunov exponent of the longest window of voiced audio data, calculated with the Rosenstein method [35] on 1435 audio samples. Circles indicate points more than twice the interquartile range away from the mean.

The longest voiced window in an audio recording is found. A nonlinear model is fitted to the reconstructed dynamics. The equilibria of the model are found, and the associated eigenvalues are calculated. Figure 2.11 shows the distribution of eigenvalue moduli and arguments, for 1400 audio recordings. The plot suggests that there is no discernable difference between the eigenvalues of the equilibria for each emotion. Appendix E shows plots of the eigenvalue locations on a complex plane, along with sections of the unit circle. As with the linear model eigenvalues, there is as much difference between plots for a single emotion as there is between different emotions. Thus, the eigenvalues of the fitted nonlinear model equilibria contain insufficient information for classifying emotion. Note that, unlike the linear model, the nonlinear model typically has a pair of complex conjugate eigenvalues outside of the unit circle. This indicates that a limit cycle exists around the equilibrium, with a frequency given approximately by the imaginary component of the eigenvalues.

The trajectories shown in appendix F show successive iterations of the nonlinear map $\mathbf{v}_{i+1} = A\mathrm{P}_5(\mathbf{v}_i)$. The training data to which the model is fitted is also shown. The initial condition $\mathbf{v}_0$ is taken as the first delay vector from the data. By visual inspection, the fitted model appears to provide an adequate representation of the data approximately 75% of the time. The usefulness of equilibrium eigenvalues is therefore limited by the fact that the fitted model, and hence the calculated eigenvalues, is not always representative of the data. It may be possible to address this issue by selecting a different nonlinear model, such as a regressive neural network.

2.4.2 Lyapunov exponents

Nonlinear systems can be characterised by their Lyapunov exponents. These quantify the average rate of divergence of nearby trajectories, in a given coordinate direction. A trajectory on a chaotic attractor will have a positive Lyapunov exponent. Hence, the exponents are often used as a test for chaos. The plots shown in appendix F demonstrate that the reconstructed vocal chord dynamics often converge to a limit cycle. This suggests that the system is not chaotic, and will thus have negative Lyapunov exponents. Nevertheless, even if the exponents are not

being used to test for chaos, they are still an informative feature of the dynamics. Lyapunov exponents are a topologically invariant feature of a system, and are therefore retained in an attractor reconstruction. Here, their predictive power for emotion classification is considered.

Numerous methods exist for estimating Lyapunov exponents from data [1]. The work here considers the Rosenstein method [35], chosen for its robustness to experimental noise, and to changes in parameter values. The method estimates only the largest Lyapunov exponent. To achieve this, pairs of near-by state vectors are identified from the reconstructed state space. These pairs are tracked across trajectories, as the system evolves. The distance between the evolving trajectories is recorded. The logarithm of the distance is taken. If the trajectories diverge or converge exponentially fast on average, the rate of change with time of the logarithm of this separation distance will be constant. Hence, a linear model is fitted, to predict the logarithm of the separation distance as a function of time. The slope of this model gives the rate of exponential divergence or convergence, and thus the Lyapunov exponent. This is repeated over multiple pairs of near-by vectors, to provide an estimate of trajectory divergence over the entire attractor.

Figure 2.12 shows the distribution of largest Lyapunov exponents for each emotion. A total of 1435 audio samples are considered. The longest window of voiced speech is extracted from each audio file. The Rosenstein method is applied, to estimate the largest Lyapunov exponent. As predicted, all the Lyapunov exponents are negative. There is no statistically significant difference between the distribution of Lyapunov exponents for any pair of emotions. Consequently, the Lyapunov exponents alone do not contain sufficient information from which to classify the emotions of a speaker. Nevertheless, there is some difference between the distributions. This suggests that, while inadequate by itself, the largest Lyapunov exponent of an attractor may be a useful feature when considered in conjunction with other dynamical statistics.

2.4.3 Correlation dimension

Pitsikalis and Maragos succeed in classifying speech sounds from the fractal dimensions of attractor reconstructions [32]. Hence, the fractal dimension of the reconstructed speech production system must contain some information about speech production dynamics. The applicability of fractal dimensions to the problem of emotion recognition is considered here.

Consider a ball of radius ε in the Euclidean space $\mathbb{R}^m$. The Lebesgue measure k of the ball grows with $k \sim \varepsilon^m$; the exponent m gives the topological dimension of the ball. Fractal dimensions generalise this notion of dimensionality to rough, often self-similar objects. Grassberger and Procaccia introduce the correlation dimension as an easily computable approximation for the fractal dimension of a strange attractor [10]. The correlation dimension considers how the number of points within an ε-ball grows with ε. This is computed by means of the correlation sum, which quantifies the proportion of points within some ε-ball in the embedding space. For sufficiently small ε, the correlation sum is shown to grow in ε^ν. The exponent ν gives the correlation dimension. The correlation dimension is intuitive and computationally simple, and is therefore considered here as a measure of the fractal dimension of a reconstructed attractor.

Ruelle states that, for N data points, one must not believe a calculated correlation dimension that is not well below $2\log_{10}(N)$ [37]. The time series considered here contain the longest sequence of voiced data found in an audio file. They are typically of length 7000 - 12,000 samples, however the shortest fall to as low as 3000 samples. This means correlation dimensions must only be considered trustworthy if they are well below approximately 6.95. Figure 2.13 shows a box plot of the correlation dimension of attractor reconstructions for a range of emotions, based

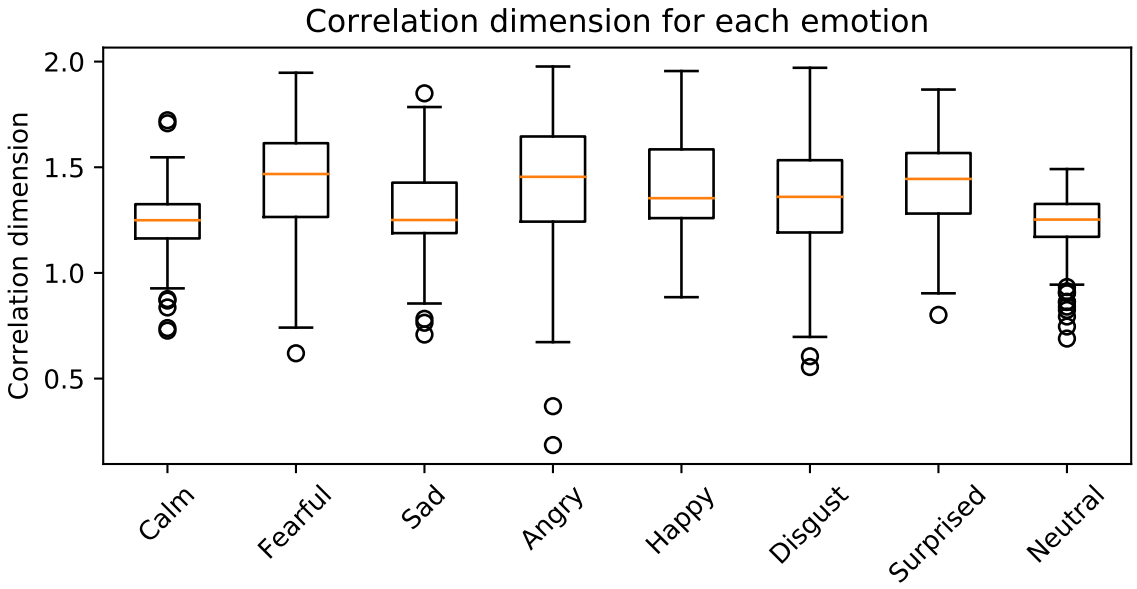


Figure 2.13: Correlation dimensions for an attractor reconstructed from the longest section of voiced data in 118 audio recordings. Circles indicate points more than twice the interquartile range away from the mean.

on 1069 speech recordings. The dimensions are seen to be sufficiently below the 6.95 threshold so as to be deemed trustworthy.

Figure 2.13 shows that, as with Lyapunov exponents, the correlation dimension alone does not contain sufficient information to classify emotions. There is no statistically significant difference between the distributions of correlation dimensions, for any pair of emotions. Nevertheless, there is a slight difference in the distributions for each emotion. The correlation dimension is therefore retained as a feature, in the hope that it will prove useful when considered alongside other dynamical features.

2.5 Classification

Section 2.4 identifies three features for quantifying speech production dynamics. These are the equilibrium eigenvalues of the speech production system, the correlation dimension of the reconstructed attractor, and the largest Lyapunov exponent. None of these features are capable of distinguishing between speaker emotions on their own. Instead, two feature spaces are defined. The first is a set of labelled two-tuples, containing the largest Lyapunov exponent, and the correlation dimension of the reconstructed state space. The second is a set of labelled four-tuples, containing the largest magnitude and angle of the fitted model equilibrium eigenvalues, in addition to the previous two features. The largest eigenvalue modulus quantifies the degree of instability of an equilibrium; the eigenvalue angle estimates the frequency of a surrounding limit cycle, where one exists. Two different classifiers are trained on the feature spaces. A K-nearest-neighbours and random forest classifier are chosen, due to their ability to accurately train and classify from small amounts of data. The two algorithms are shown in appendix C to have very similar classification accuracies.

The two algorithms are trained on feature sets for pairs of emotions. To achieve this, the dynamical features are computed from the largest window of voiced data in each recording of the RAVDESS database. For a given pair of emotions, the corresponding feature spaces are

	Angry	Calm	Disgust	Fearful	Happy	Neutral	Sad	Surprised
Angry	-	0.7411	0.5664	0.5814	0.5854	0.6286	0.6096	0.5943
Calm	0.7361	-	0.6617	0.7261	0.7118	0.6433	0.5618	0.7311
Disgust	0.5571	0.6586	-	0.5397	0.5914	0.6432	0.5248	0.5852
Fearful	0.5775	0.7289	0.5445	-	0.5972	0.7045	0.5855	0.5628
Happy	0.5904	0.6889	0.5934	0.5666	-	0.6455	0.5862	0.6407
Neutral	0.6090	0.6433	0.6241	0.7014	0.6509	-	0.5409	0.6791
Sad	0.6082	0.5704	0.5331	0.5672	0.5693	0.5382	-	0.6545
Surprised	0.5982	0.7239	0.5731	0.5838	0.6272	0.6859	0.6479	-

Figure 2.14: Average classification accuracy for pairs of strong emotions, using a classifier trained on correlation dimension and largest Lyapunov exponent. The best accuracy from K-nearest-neighbours and a random forest classifier is shown.

	Angry	Calm	Disgust	Fearful	Happy	Neutral	Sad	Surprised
Angry	-	0.6990	0.5165	0.5125	0.5776	0.5620	0.6195	0.6395
Calm	0.7105	-	0.6370	0.7145	0.6391	0.6356	0.6014	0.6714
Disgust	0.5295	0.6525	-	0.5275	0.5933	0.6347	0.5018	0.5767
Fearful	0.5250	0.7170	0.5290	-	0.5800	0.7140	0.5955	0.5495
Happy	0.5886	0.6405	0.5967	0.5914	-	0.6512	0.5578	0.6373
Neutral	0.5780	0.6494	0.6367	0.6853	0.6344	-	0.5812	0.6725
Sad	0.6405	0.6155	0.5173	0.5891	0.5778	0.5553	-	0.5968
Surprised	0.6195	0.6652	0.5671	0.5390	0.6355	0.6763	0.6177	-

Figure 2.15: Average classification accuracy for pairs of strong emotions, using a classifier trained on correlation dimension, maximum equilibrium eigenvalue modulus and argument, and largest Lyapunov exponent. The best accuracy from K-nearest-neighbours and a random forest classifier is shown.

randomly partitioned into test and training data. The test data typically contains 128 labelled feature vectors per emotion; the training data typically contains 14 feature vectors per emotion. A K-nearest-neighbours classifier and a random forest classifier are trained and scored from these features. As the feature space is randomly partitioned, the exact accuracy score varies slightly depending on the partitioning. Consequently, the fitting and testing is repeated 100 times, using a different random partition each time. The average accuracy score is recorded. Note that the accuracy score tables are not entirely symmetric, as a result of this random variation in accuracy scores.

Tables 2.14 and 2.15 show the best pairwise classification accuracy of the two algorithms, for the two- and four-dimensional feature spaces, respectively. Note that these tables only consider recordings of strong-intensity emotional speech. The pairwise classification accuracies of each individual algorithm, for both strong-only, and all audio recordings, are shown in appendix C.

The classifier performs better on the two-dimensional feature space. The plot shown in figure 2.11 shows that all emotions have a visually identical distribution of equilibrium eigenvalue moduli and angles. This suggests that the equilibrium eigenvalues contain no useful information about speaker emotions. As a result, the inclusion of these features is akin to adding random noise into the feature space. The resulting drop in classification accuracy is therefore unsurprising.

One would expect a pairwise classification accuracy of 0.5, through random guesswork alone. An accuracy below 0.5 can be turned into an accuracy above 0.5 by swapping the resulting

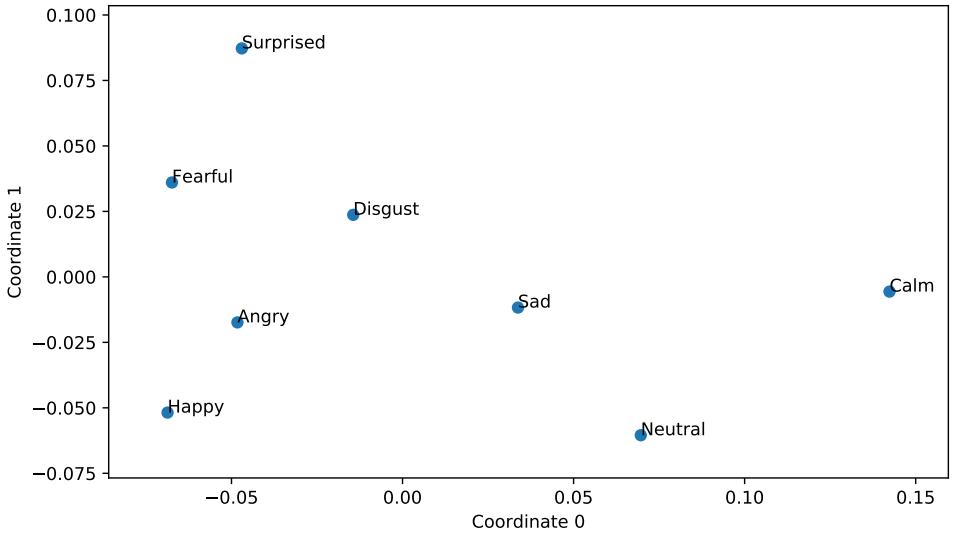


Figure 2.16: Projection of classification accuracies onto a 2D plane, for a k-nearest-neighbours classifier trained on strong emotional speech. Multidimensional scaling is used to transform a dissimilarity matrix into a set of coordinates. Further apart pairs of points are more different than nearby pairs, and can be classified more accurately.

class labels. 0.5 thus forms a minimum accuracy baseline, against which the classification performance can be judged. The classifiers sometimes perform well, relative to this baseline. For strong-intensity emotional speech, calm and angry can be distinguished correctly approximately 74% of the time, and calm and fearful can be correctly classified approximately 73% of the time. Nevertheless, some pairs of emotions demonstrate an exceedingly poor classification accuracy. Fear and disgust can only be identified at an accuracy of 54% - barely above the baseline. A possible explanation for this is that fear and disgust are both stressed emotions, and the classifier works best when detecting stressed from unstressed. This hypothesis explains why angry emotional speech can be distinguished from calm or neutral speech at a higher accuracy than for fear or disgust. The hypothesis also agrees with the discussion presented in the literature review (section 1.2), where it is proposed that it is easier to determine the presence or absence of stress in a speaker, than their specific emotional state.

To test the hypothesis, a multidimensional scaling procedure is applied. A hypothetical pair of emotions showing a 50% classification accuracy must have a very similar set of features, and are therefore indistinguishable from one another. In contrast, a pair with a 100% classification accuracy would be completely distinguishable, and therefore entirely dissimilar from each other. Pairwise classification accuracies thus quantify the degree of dissimilarity between different emotions. An emotion and itself has a dissimilarity of zero; a pair of different emotions have a dissimilarity score given by the proportion of correct classifications, minus 0.5. Multidimensional scaling transforms a dissimilarity matrix into a set of coordinates, such that the Euclidean distance between any given pair of points represents the degree of dissimilarity between those points. Figure 2.16 shows the application of multidimensional scaling to the emotional data. Near-by points on the plot show a high degree of similarity, and thus cannot be classified accurately. The previously discussed hypothesis suggests that the emotions should be split into two well-separated clusters - one for stressed emotions, and one for unstressed emotions. No clear split is observed. The multidimensional scaling results therefore disprove the hypothesis.

Nevertheless, surprised, fearful, disgust, angry, and happy are all close together in the multidimensional scaling, and well-separated from calm and neutral. These similar emotions are collected together into one class, referred to here as the excited-state class. Calm and neutral - both close together - are collected together as a second class, referred to as the ground-state class. Sad emotional speech shares similarities with both the ground-state and excited-state classes, and is hence ignored here for simplicity. To further test the hypothesis, one can calculate the classification accuracy between these two metaclasses. A k-nearest-neighbours classifier produces a 77% classification accuracy, based on 741 training examples and 185 test samples. 77% of the test and training data lie within the excited-state class, meaning the classifier could achieve an identical accuracy by guessing an audio sample to be in the excited state every time. The classifier is therefore unable to distinguish between the two best-separated metastates, further disproving the stressed-unstressed hypothesis.

Chapter 3

Discussion and conclusion

3.1 Project achievements

This report investigates the recognition of emotions in speech. Current research approaches this problem by focusing on prosodic features. These quantify speech properties such as pitch and speaking rate. It is argued here that prosodic features are insufficient for accurately recognising emotions. An alternative problem approach is proposed. This focuses on studying the dynamics of the vocal chords. The work of Tolkmitt et al. suggest that the vocal chord dynamics change under the stress-state of a speaker [48]. This project seeks to determine whether these dynamical changes provide sufficient information for emotion recognition. Here, the main successes of the project are identified.

Speech contains sounds generated from the oral cavity, and from the vocal folds. Only the dynamics of the vocal folds are considered in this project. The dynamical analysis presented here is inapplicable to unvoiced sounds. Unvoiced sound must therefore be removed, before the analysis can take place. A novel algorithm is proposed, to identify sections of voiced audio data. The algorithm is tested by applying it to randomly selected data files, and listening to the results. It is observed as having a desirably low rate of false (unvoiced) acceptance, whilst also retaining long enough windows of voiced data to be useful for further analysis.

Results from differential topology are used to reconstruct the dynamics of the speech production system from audio recordings. It is found that the conventional method of determining embedding dimension - false nearest neighbours - is difficult to apply to speech data. Consequently, an entirely new algorithm is proposed to determine embedding dimensions. This algorithm is validated on synthetic data. The failure modes of the algorithm are identified and discussed. The result of the algorithm is taken as the embedding dimension for reconstructing the speech production dynamics.

The reconstructed dynamics are topologically equivalent to those of the speech production system. A method is developed to extract information from these dynamics. This contains two parts. Firstly, a nonlinear model is fitted. A gradient descent optimisation procedure is proposed, so as to optimise a model over entire trajectories. The limitations with this procedure are considered. Secondly, a procedure is derived for extracting equilibria from the model. The nuances of numerically finding nonlinear equilibria are discussed. An algorithm is proposed, to identify all equilibria that are well-fitted by the training data. This algorithm is designed in such a way as to be robust against any spurious equilibria arising from untrustworthy extrapolations of the dynamics.

A classifier is constructed, to discriminate between pairs of emotions. The classifier demonstrates a reasonable accuracy in some cases. Strong intensity angry and calm speech can be discerned accurately 74% of the time; fearful and calm speech can be classified at a 73% accuracy. Nevertheless, the classifier is little better than random guesswork for other pairs of emotions. Sad and neutral are only identified correctly 54% of the time. A 57-59% classification accuracy is achieved on happy and sad data; this compares to 66% for the deep learning approach employed on the same data by Jannat et al. [17]. The results produced here are comparable to those achieved by the studies discussed in section 1.2, for which prosodic features are used. Thus, while the nonlinear dynamics approach does not exceed the accuracy of prosodic classifiers, neither does it lag behind them.

3.2 Future work

The work presented here considers a nonlinear model fitted to reconstructed speech system dynamics. The equilibria of this model are extracted. It is found that these equilibria contain insufficient information to categorise emotions. One may choose to address this by fitting a model to the vocal tract state instead. Linear predictive coding is often used to estimate vocal tract parameters [28]. This models the vocal folds as a buzzer, and the vocal tract as a linear filter. Inverse filtering is used to extract the vocal fold vibrations, by fitting an infinite impulse filter to the signal, and treating the residuals as the original input signal. Linear predictive coding is not considered in this report, as the windowing methods required to use it cause the vocal fold vibrations to be incorrectly and irreversibly distorted. Nevertheless, one could model the dynamics of the vocal tract itself through consideration of the fitted filer parameters.

Alternatively, one can quantify the produced speech by means of a harmonic decomposition. Consider the harmonic model presented in equation (2.1). The fundamental frequency ω_0, and the Fourier parameters a_i and b_i together contain all the information required to reconstruct a window of speech signal. The fundamental frequency and Fourier parameters define a state space for the speech production system. One may consider long-term speech trends by considering the dynamics of this state space.

Batliner et al. note that prosody ceases to be a reliable indicator of emotion, for realistic, non-acted scenarios [3]. The nonlinear dynamics approach discussed here is intended to overcome this limitation, by identifying the physiological changes in the vocal tract identified by Scherer [40]. It is not unreasonable to assume that actors do not demonstrate these physiological changes, since they are not actually experiencing the target emotion. Future work should therefore investigate how the classification accuracies achieved in this report change, when recordings exhibiting real stress are used. Note also that human listeners sometimes struggle to identify the audio recordings considered here (see table 1.1 for the proportion of correct labellings by human validators). The human recognition rate drops to as low as 29% accuracy, for normal-

intensity happy recordings in the RAVDESS data set. Such work would therefore benefit from any real-stress audio recordings being validated to a higher accuracy than RAVDESS.

The fitted model parameters are not considered for classification in this report, as a result of the high dimensionality of the data. Nevertheless, with suitable computational approaches, one may be able to classify emotions to a higher accuracy than that achieved here, by considering the nonlinear model itself as a feature vector. Future work should construct a classifier that learns from the entire nonlinear model.

Chapter 4

Bibliography

[1] Jan Awrejcewicz, Anton Krysko, Nikolay Erofeev, Vitalyj Dobriyan, Marina Barulina, and Vadim Krysko. Quantifying chaos by various computational methods. part 1: simple systems. *Entropy*, 20(3):175, 2018.

[2] R Badii, G Broggi, B Derighetti, Ms Ravani, S Ciliberto, A Politi, and MA Rubio. Dimension increase in filtered chaotic signals. *Physical Review Letters*, 60(11):979, 1988.

[3] Anton Batliner, Kerstin Fischer, Richard Huber, Jörg Spilker, and Elmar Nöth. How to find trouble in communication. *Speech communication*, 40(1-2):117–143, 2003.

[4] David A Berry, Hanspeter Herzel, Ingo R Titze, and Katharina Krischer. Interpretation of biomechanical simulations of normal and chaotic vocal fold oscillations with empirical eigenfunctions. *The Journal of the Acoustical Society of America*, 95(6):3595–3604, 1994.

[5] Peter Birkholz, BJ Kröger, and P Birkholz. A survey of self-oscillating lumped-element models of the vocal folds. *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung*, pages 47–58, 2011.

[6] Martin Casdagli, Stephen Eubank, J Doyne Farmer, and John Gibson. State space reconstruction in the presence of noise. *Physica D: Nonlinear Phenomena*, 51(1-3):52–98, 1991.

[7] J-P Eckmann and David Ruelle. Ergodic theory of chaos and strange attractors. In *The Theory of Chaotic Attractors*, pages 273–312. Springer, 1985.

[8] DJ Ewins. Basics and state-of-the-art of modal testing. *Sadhana*, 25(3):207–220, 2000.

[9] Andrew M Fraser and Harry L Swinney. Independent coordinates for strange attractors from mutual information. *Physical review A*, 33(2):1134, 1986.

[10] Peter Grassberger and Itamar Procaccia. Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, 9(1-2):189–208, 1983.

[11] Kazi Mahmudul Hassan, Ekramul Hamid, and Khademul Islam Molla. A method for voiced/unvoiced classification of noisy speech by analyzing time-domain features of spectrogram image. *Science Journal of Circuits, Systems and Signal Processing*, 6(2):11–17, 2017.

[12] Rainer Hegger, Holger Kantz, and Thomas Schreiber. Practical implementation of nonlinear time series methods: The tisean package. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 9(2):413–435, 1999.

[13] Javier Hernandez, Daniel McDuff, Xavier Benavides, Judith Amores, Pattie Maes, and Rosalind Picard. Autoemotive: bringing empathy to the driving experience to manage stress. In *Proceedings of the 2014 companion publication on Designing interactive systems*, pages 53–56. ACM, 2014.

[14] Hanspeter Herzel. Bifurcations and chaos in voice signals. *Applied Mechanics Reviews*, 46(7):399–413, 1993.

[15] Hanspeter Herzel and J Wendler. Evidence of chaos in phonatory samples. In *Second European Conference on Speech Communication and Technology*, 1991.

[16] Julia Hirschberg, Stefan Benus, Jason M Brenier, Frank Enos, Sarah Friedman, Sarah Gilman, Cynthia Girand, Martin Graciarena, Andreas Kathol, Laura Michaelis, et al. Distinguishing deceptive from non-deceptive speech. In *Ninth European Conference on Speech Communication and Technology*, 2005.

[17] Rahatul Jannat, Iyonna Tynes, Lott La Lime, Juan Adorno, and Shaun Canavan. Ubiquitous emotion recognition using audio and video data. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, pages 956–959. ACM, 2018.

[18] Jack J Jiang, Yu Zhang, and Jennifer Stern. Modeling of chaotic vibrations in symmetric vocal folds. *The Journal of the Acoustical Society of America*, 110(4):2120–2128, 2001.

[19] James L Kaplan and James A Yorke. Chaotic behavior of multidimensional difference equations. In *Functional Differential equations and approximation of fixed points*, pages 204–227. Springer, 1979.

[20] Matthew B Kennel, Reggie Brown, and Henry DI Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical review A*, 45(6):3403, 1992.

[21] H_S Kim, R Eykholt, and JD Salas. Nonlinear dynamics, delay times, and embedding windows. *Physica D: Nonlinear Phenomena*, 127(1-2):48–60, 1999.

[22] John David Michael Henry Laver. Individual features in voice quality. 1987.

[23] Wolfgang Liebert and HG Schuster. Proper choice of the time delay for the analysis of chaotic time series. *Physics Letters A*, 142(2-3):107–111, 1989.

[24] Steven R Livingstone and Frank A Russo. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391, 2018.

[25] Edward N Lorenz. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2):130–141, 1963.

[26] Mark J McGuinness. The fractal dimension of the lorenz attractor. *Physics Letters A*, 99(1):5–9, 1983.

[27] Jesper Kjær Nielsen, Tobias Lindstrøm Jensen, Jesper Rindom Jensen, Mads Græsbøll Christensen, and Søren Holdt Jensen. Fast fundamental frequency estimation: Making a statistically efficient estimator computationally efficient. *Signal Processing*, 135:188–197, 2017.

[28] Douglas O'Shaughnessy. Linear predictive coding. *IEEE potentials*, 7(1):29–32, 1988.

[29] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318, 2013.

[30] James W Pennebaker, Martha E Francis, and Roger J Booth. Linguistic inquiry and word count (liwc): Liwc2001 manual, 2001.

[31] Valery Petrushin. Emotion in speech: Recognition and application to call centers. In *Proceedings of artificial neural networks in engineering*, volume 710, 1999.

[32] Vassilis Pitsikalis and Petros Maragos. Analysis and classification of speech signals by generalized fractal dimension features. *Speech Communication*, 51(12):1206–1223, 2009.

[33] Lawrence Rabiner. On the use of autocorrelation analysis for pitch detection. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(1):24–33, 1977.

[34] Carl Rhodes and Manfred Morari. False-nearest-neighbors algorithm and noise-corrupted time series. *Physical Review E*, 55(5):6162, 1997.

[35] Michael T Rosenstein, James J Collins, and Carlo J De Luca. A practical method for calculating largest lyapunov exponents from small data sets. *Physica D: Nonlinear Phenomena*, 65(1-2):117–134, 1993.

[36] Otto E Rössler. An equation for continuous chaos. *Physics Letters A*, 57(5):397–398, 1976.

[37] David Ruelle. The claude bernard lecture, 1989-deterministic chaos: the science and the fiction. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 427(1873):241–248, 1990.

[38] Tim Sauer, James A Yorke, and Martin Casdagli. Embedology. *Journal of statistical Physics*, 65(3-4):579–616, 1991.

[39] Timothy D Sauer. Attractor reconstruction. *Scholarpedia*, 1(10):1727, 2006.

[40] Klaus R Scherer. Vocal affect expression: A review and a model for future research. *Psychological bulletin*, 99(2):143, 1986.

[41] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[42] Norman C Severo and Marvin Zelen. Normal approximation to the chi-square and non-central f probability functions. *Biometrika*, 47(3/4):411–416, 1960.

[43] Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

[44] Ina Steinecke and Hanspeter Herzel. Bifurcations in an asymmetric vocal-fold model. *The Journal of the Acoustical Society of America*, 97(3):1874–1884, 1995.

[45] Steven Strogatz, Mark Friedman, A John Mallinckrodt, and Susan McKay. Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering. *Computers in Physics*, 8(5):532–532, 1994.

[46] Robert Szalai, David Ehrhardt, and George Haller. Nonlinear model identification and spectral submanifolds for multi-degree-of-freedom mechanical vibrations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2202):20160759, 2017.

[47] Floris Takens. Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980*, pages 366–381. Springer, 1981.

[48] Frank J Tolkmitt and Klaus R Scherer. Effect of experimentally induced stress on vocal parameters. *Journal of Experimental Psychology: Human Perception and Performance*, 12(3):302, 1986.

[49] Dimitrios Ververidis and Constantine Kotropoulos. Emotional speech recognition: Resources, features, and methods. *Speech communication*, 48(9):1162–1181, 2006.

[50] John C Wells. Ipa transcription systems for english. *PG Bulletin: Bulletin of teachers of English phonetics in Chile and abroad*, 9, 2001.

[51] Hassler Whitney. Differentiable manifolds. *Annals of Mathematics*, pages 645–680, 1936.

[52] Zhihong Zeng, Maja Pantic, Glenn I Roisman, and Thomas S Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE transactions on pattern analysis and machine intelligence*, 31(1):39–58, 2009.

Appendices

A: Software development

A suite of software tools have been developed to implement the analysis presented in this report. The main programs are available at GitHub: MarkBlyth/EmotionDetection. These are released under the GNU GPL3 as free, copyleft software. The programs are written in Python, and follow a functional programming design. Where relevant, parallelisation has been used, to allow efficient execution over multiple cores. Also included are the data files used to generate some of the plots presented here.

B: ε-uniqueness algorithm

Consider some set $\mathcal{S}$ of points. We wish to find some subset $\mathcal{U} \subseteq S$, such that $\forall \mathbf{u}, \mathbf{v} \in \mathcal{U}$, $\left\|\mathbf{u} - \mathbf{v}\right\|_2 \geq \varepsilon$, for some ε. Here, an efficient algorithm is presented to achieve this.

- Let $\mathcal{U} = \emptyset$.
- Repeat until $\mathcal{S} = \emptyset$:
  - Choose any element $\mathbf{u} \in \mathcal{S}$. Remove $\mathbf{u}$ from $\mathcal{S}$ and add it to $\mathcal{U}$.
  - For this choice of $\mathbf{u}$, for each $\mathbf{v} \in \mathcal{S}$, remove $\mathbf{v}$ from $\mathcal{S}$ if $\left\|\mathbf{v} - \mathbf{u}\right\|_2 < \varepsilon$.

The presented algorithm has a best-case run time of $\mathcal{O}(n)$, a worst-case run time of $\mathcal{O}(n^2)$, and an average run time of $\mathcal{O}(n \log n)$. Note that $\mathcal{U}$ is non-unique. This algorithm therefore only chooses one such possible $\mathcal{U}$.

C: Full classifier performance results

The main report considers using the K-nearest-neighbours (KNN) algorithm and a random forest algorithm, to classify pairs of strong-intensity emotions. The tables presented here extend this, by presenting the average classification accuracy for pairs of strong and normal intensity emotions, and for both a random forest classifier and KNN classifier individually. Tables are presented for both the two- and four-dimensional feature vectors.

	Neutral	Fearful	Disgust	Surprised	Calm	Angry	Sad	Happy
Neutral	-	0.6900	0.6180	0.6575	0.5769	0.5713	0.5447	0.6763
Fearful	0.6980	-	0.5120	0.5690	0.6995	0.5300	0.5695	0.5343
Disgust	0.5980	0.5235	-	0.5348	0.6565	0.4845	0.5073	0.5981
Surprised	0.6656	0.5571	0.5200	-	0.6671	0.5857	0.5950	0.6036
Calm	0.5919	0.7070	0.6765	0.6548	-	0.7065	0.6027	0.6191
Angry	0.5840	0.5390	0.4880	0.5814	0.7075	-	0.6232	0.5724
Sad	0.5576	0.5805	0.5150	0.6150	0.6123	0.6073	-	0.5435
Happy	0.6525	0.5343	0.6067	0.5905	0.6150	0.5910	0.5235	-

Figure 1: Average classification accuracy for pairs of strong emotions, using a K-nearest-neighbours classifier trained on correlation dimension, maximum equilibrium eigenvalue modulus and argument, and largest Lyapunov exponent.

	Neutral	Fearful	Disgust	Surprised	Calm	Angry	Sad	Happy
Neutral	-	0.6773	0.6480	0.6488	0.6381	0.5907	0.5482	0.6687
Fearful	0.6727	-	0.4865	0.5262	0.7110	0.5395	0.5959	0.5671
Disgust	0.6227	0.5010	-	0.5938	0.6555	0.5050	0.5068	0.5638
Surprised	0.6312	0.5352	0.5710	-	0.6571	0.6067	0.5768	0.6218
Calm	0.6500	0.6915	0.6595	0.6552	-	0.6765	0.6109	0.6295
Angry	0.6133	0.5065	0.5265	0.6390	0.6760	-	0.6127	0.5619
Sad	0.5524	0.6009	0.5068	0.5864	0.6236	0.6100	-	0.5661
Happy	0.6469	0.5600	0.5695	0.6273	0.6300	0.5605	0.5604	-

Figure 2: Average classification accuracy for pairs of strong emotions, using a random forest classifier trained on correlation dimension, maximum equilibrium eigenvalue modulus and argument, and largest Lyapunov exponent.

	Angry	Fearful	Disgust	Calm	Happy	Surprised	Neutral	Sad
Angry	-	0.5175	0.4665	0.7010	0.5810	0.5714	0.5527	0.6205
Fearful	0.5190	-	0.5155	0.6920	0.5229	0.5490	0.7000	0.5809
Disgust	0.4655	0.5105	-	0.6480	0.5971	0.5343	0.5880	0.5177
Calm	0.7075	0.7070	0.6465	-	0.6068	0.6586	0.5781	0.5814
Happy	0.5781	0.5495	0.5881	0.6355	-	0.5927	0.6687	0.5530
Surprised	0.5938	0.5481	0.5310	0.6829	0.5973	-	0.6519	0.6145
Neutral	0.5700	0.7080	0.5900	0.5813	0.6600	0.6375	-	0.5571
Sad	0.6159	0.5727	0.5141	0.5982	0.5509	0.6073	0.5535	-

Figure 3: Average classification accuracy for pairs of emotions, using a K-nearest-neighbours classifier trained on correlation dimension, maximum equilibrium eigenvalue modulus and argument, and largest Lyapunov exponent.

	Angry	Fearful	Disgust	Calm	Happy	Surprised	Neutral	Sad
Angry	-	0.5380	0.5130	0.6815	0.5590	0.6410	0.5893	0.6045
Fearful	0.5335	-	0.5150	0.6985	0.5552	0.5433	0.6840	0.6009
Disgust	0.5115	0.4995	-	0.6475	0.5743	0.5924	0.6320	0.4977
Calm	0.6845	0.7125	0.6465	-	0.6314	0.6538	0.6525	0.5895
Happy	0.5610	0.5895	0.5600	0.6609	-	0.6232	0.6775	0.5596
Surprised	0.6186	0.5376	0.5867	0.6638	0.6345	-	0.6438	0.6000
Neutral	0.5773	0.6713	0.6380	0.6419	0.6469	0.6350	-	0.5724
Sad	0.6127	0.5832	0.5018	0.6173	0.5687	0.6005	0.5582	-

Figure 4: Average classification accuracy for pairs of emotions, using a random forest classifier trained on correlation dimension, maximum equilibrium eigenvalue modulus and argument, and largest Lyapunov exponent.

	Angry	Sad	Disgust	Neutral	Calm	Happy	Fearful	Surprised
Angry	-	0.6114	0.5486	0.6071	0.7361	0.5846	0.5843	0.5961
Sad	0.6054	-	0.5393	0.5295	0.5411	0.5814	0.5572	0.6424
Disgust	0.5507	0.5172	-	0.6286	0.6517	0.6059	0.5483	0.5524
Neutral	0.5857	0.5314	0.6241	-	0.5886	0.6577	0.7159	0.6750
Calm	0.7436	0.5557	0.6607	0.5357	-	0.6832	0.7061	0.7229
Happy	0.5993	0.5810	0.5876	0.6559	0.6911	-	0.5917	0.6252
Fearful	0.5729	0.5755	0.5369	0.6905	0.7104	0.5790	-	0.5786
Surprised	0.5957	0.6428	0.5510	0.6995	0.7411	0.6276	0.5693	-

Figure 5: Average classification accuracy for pairs of strong emotions, using a K-nearest-neighbours classifier trained on correlation dimension and largest Lyapunov exponent.

	Angry	Sad	Disgust	Neutral	Calm	Happy	Fearful	Surprised
Angry	-	0.5932	0.5475	0.6133	0.7075	0.5668	0.5504	0.5671
Sad	0.5918	-	0.4903	0.5227	0.5625	0.5893	0.5610	0.6262
Disgust	0.5596	0.4831	-	0.6177	0.6390	0.5455	0.5390	0.5638
Neutral	0.6200	0.5405	0.6186	-	0.6500	0.6305	0.7068	0.6805
Calm	0.7129	0.5664	0.6624	0.6143	-	0.6604	0.7171	0.6993
Happy	0.5693	0.5838	0.5397	0.6241	0.6554	-	0.5641	0.6424
Fearful	0.5404	0.5831	0.5386	0.7018	0.7229	0.5752	-	0.5403
Surprised	0.5618	0.6300	0.5710	0.6945	0.7204	0.6317	0.5279	-

Figure 6: Average classification accuracy for pairs of strong emotions, using a random forest classifier trained on correlation dimension and largest Lyapunov exponent.

	Angry	Disgust	Happy	Sad	Neutral	Fearful	Surprised	Calm
Angry	-	0.5582	0.5736	0.5979	0.5981	0.5893	0.6089	0.7286
Disgust	0.5436	-	0.5759	0.5276	0.6400	0.5569	0.5621	0.6734
Happy	0.5807	0.5972	-	0.5790	0.6600	0.5538	0.6238	0.7061
Sad	0.6129	0.5186	0.5697	-	0.5505	0.5659	0.6407	0.5625
Neutral	0.5962	0.6123	0.6577	0.5482	-	0.7018	0.6782	0.5548
Fearful	0.5775	0.5534	0.5893	0.5583	0.7191	-	0.5734	0.7075
Surprised	0.5996	0.5572	0.6266	0.6434	0.6936	0.5872	-	0.7386
Calm	0.7389	0.6748	0.7061	0.5689	0.5776	0.7114	0.7379	-

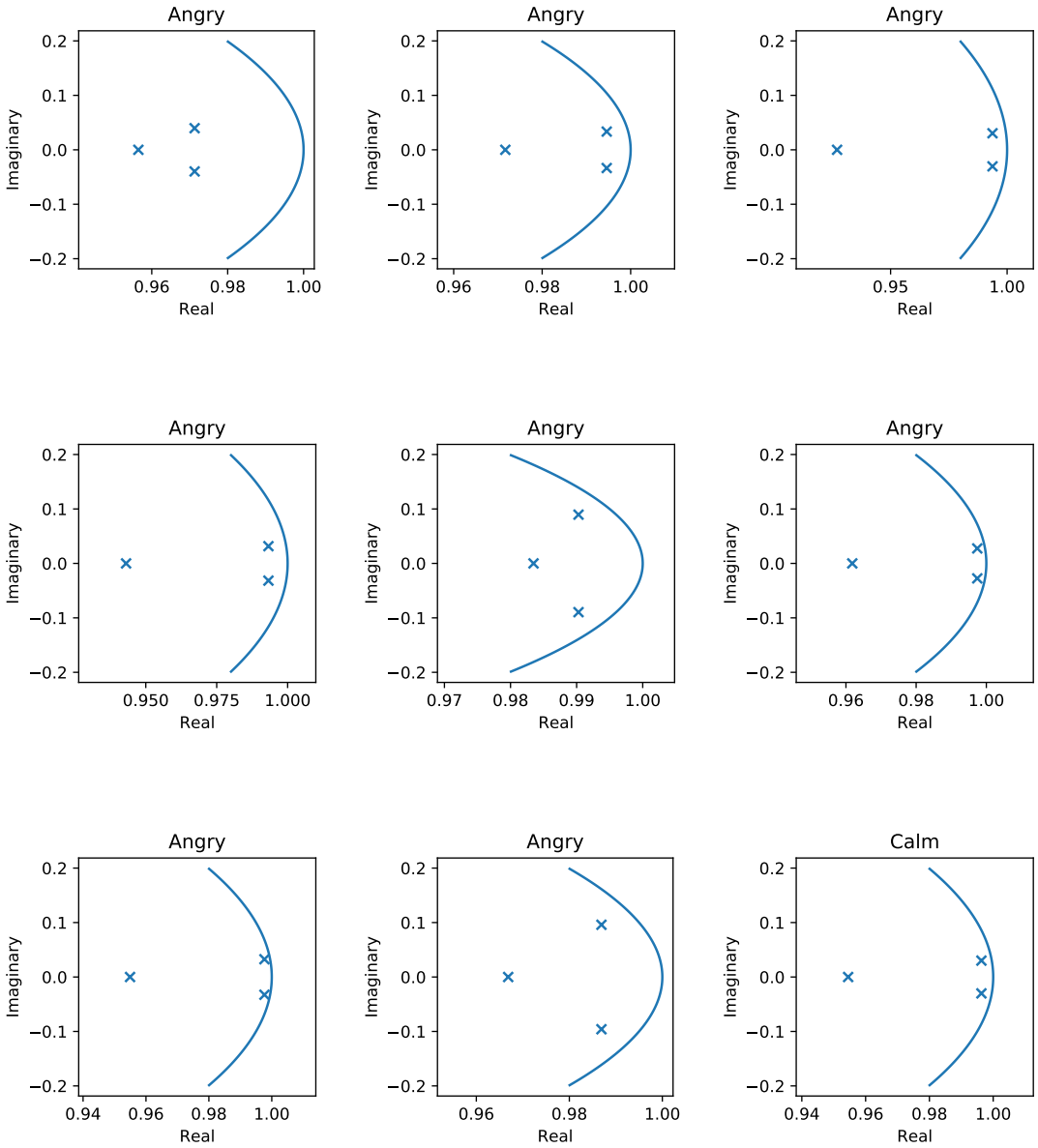
Figure 7: Average classification accuracy for pairs of emotions, using a K-nearest-neighbours classifier trained on correlation dimension and largest Lyapunov exponent.

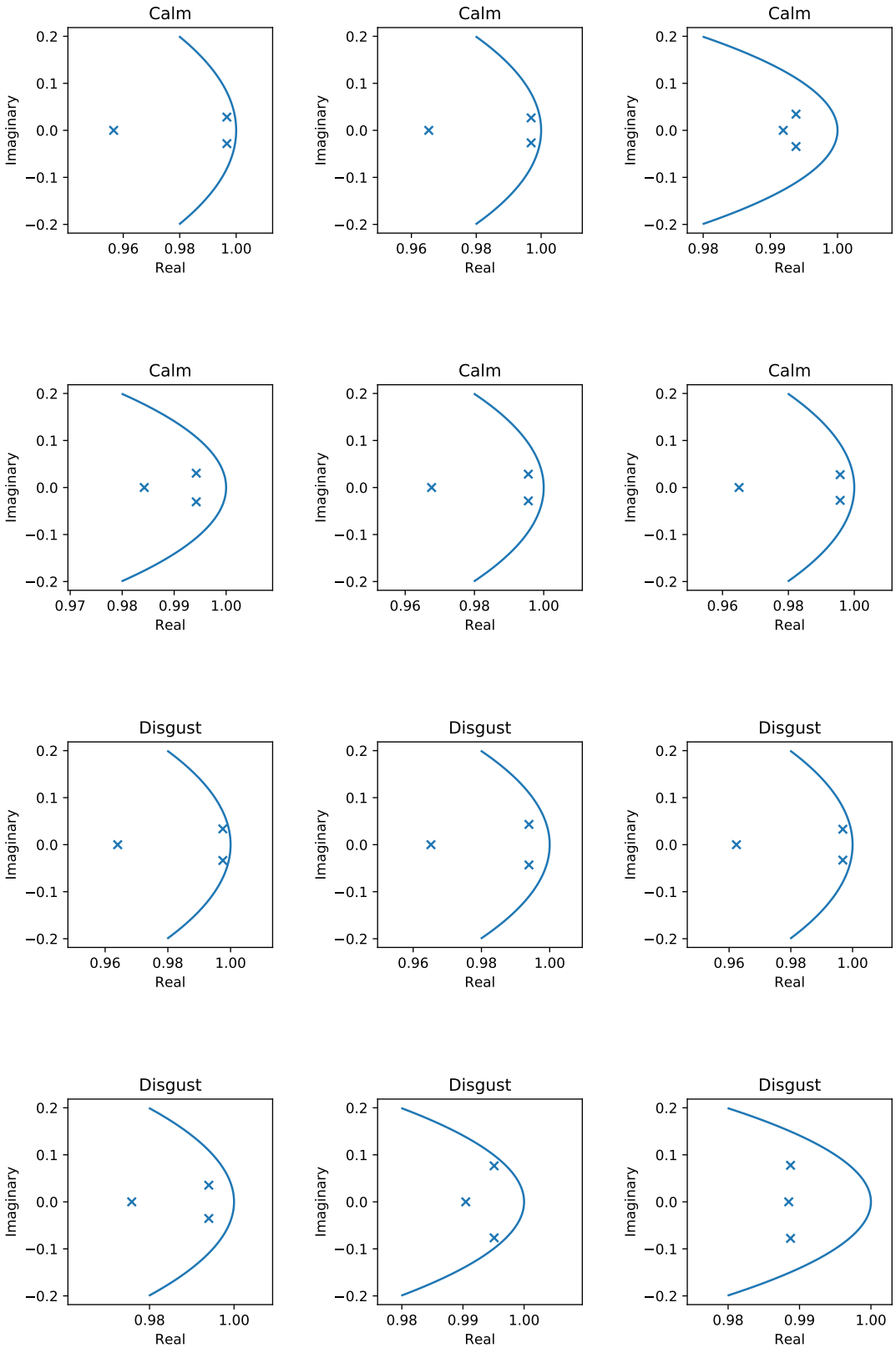
	Angry	Disgust	Happy	Sad	Neutral	Fearful	Surprised	Calm
Angry	-	0.5721	0.5532	0.5839	0.6081	0.5525	0.5793	0.6961
Disgust	0.5579	-	0.5266	0.4876	0.6345	0.5348	0.5800	0.6683
Happy	0.5621	0.5372	-	0.5797	0.6355	0.5503	0.6293	0.6732
Sad	0.6064	0.4952	0.5828	-	0.5445	0.5597	0.6214	0.5721
Neutral	0.6043	0.6177	0.6364	0.5473	-	0.7105	0.6714	0.6305
Fearful	0.5464	0.5421	0.5676	0.5669	0.7005	-	0.5390	0.7182
Surprised	0.5911	0.5762	0.6334	0.6193	0.6945	0.5407	-	0.7289
Calm	0.7021	0.6748	0.6518	0.5621	0.6333	0.7175	0.7254	-

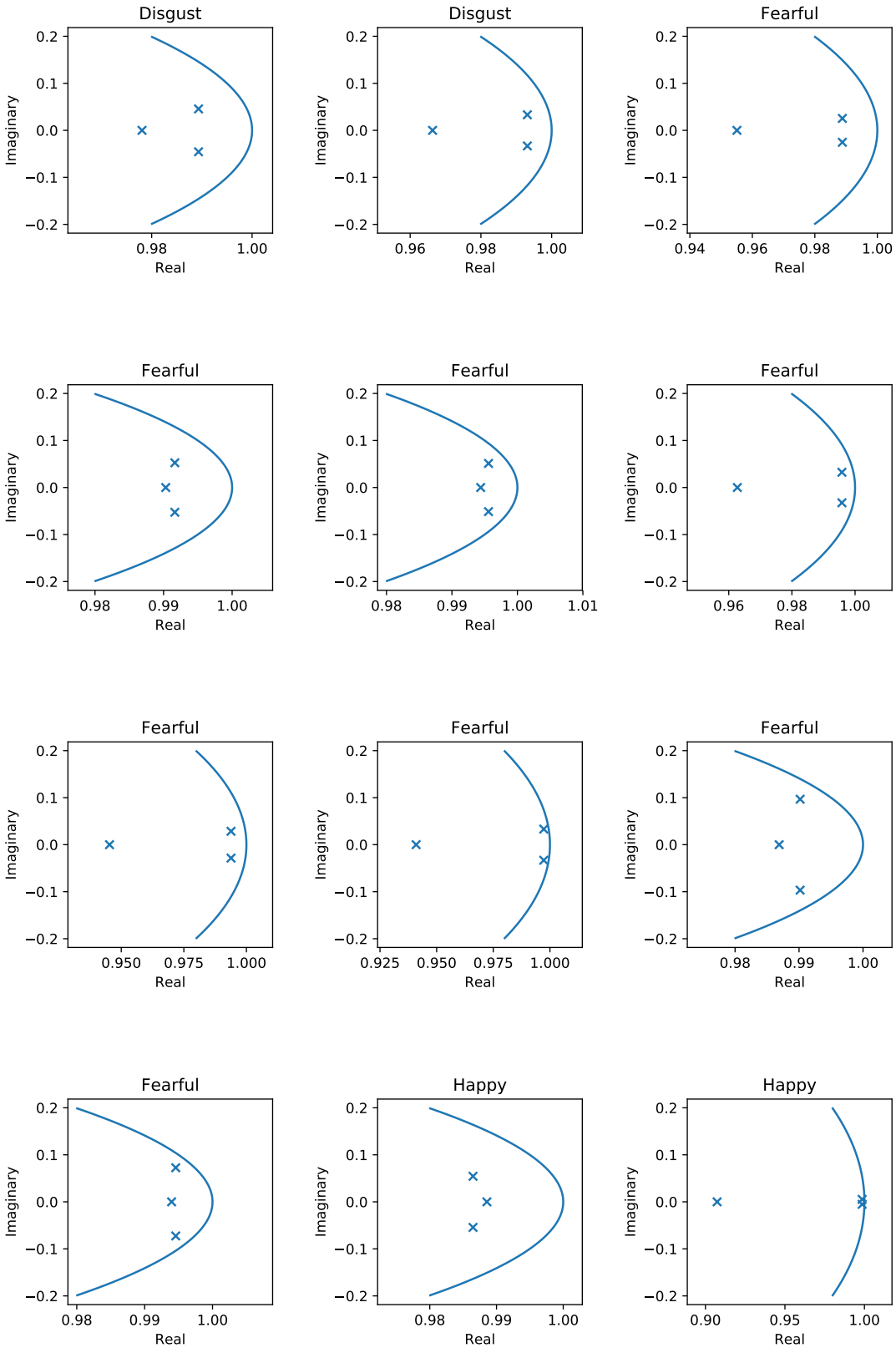
Figure 8: Average classification accuracy for pairs of emotions, using a random forest classifier trained on correlation dimension and largest Lyapunov exponent.

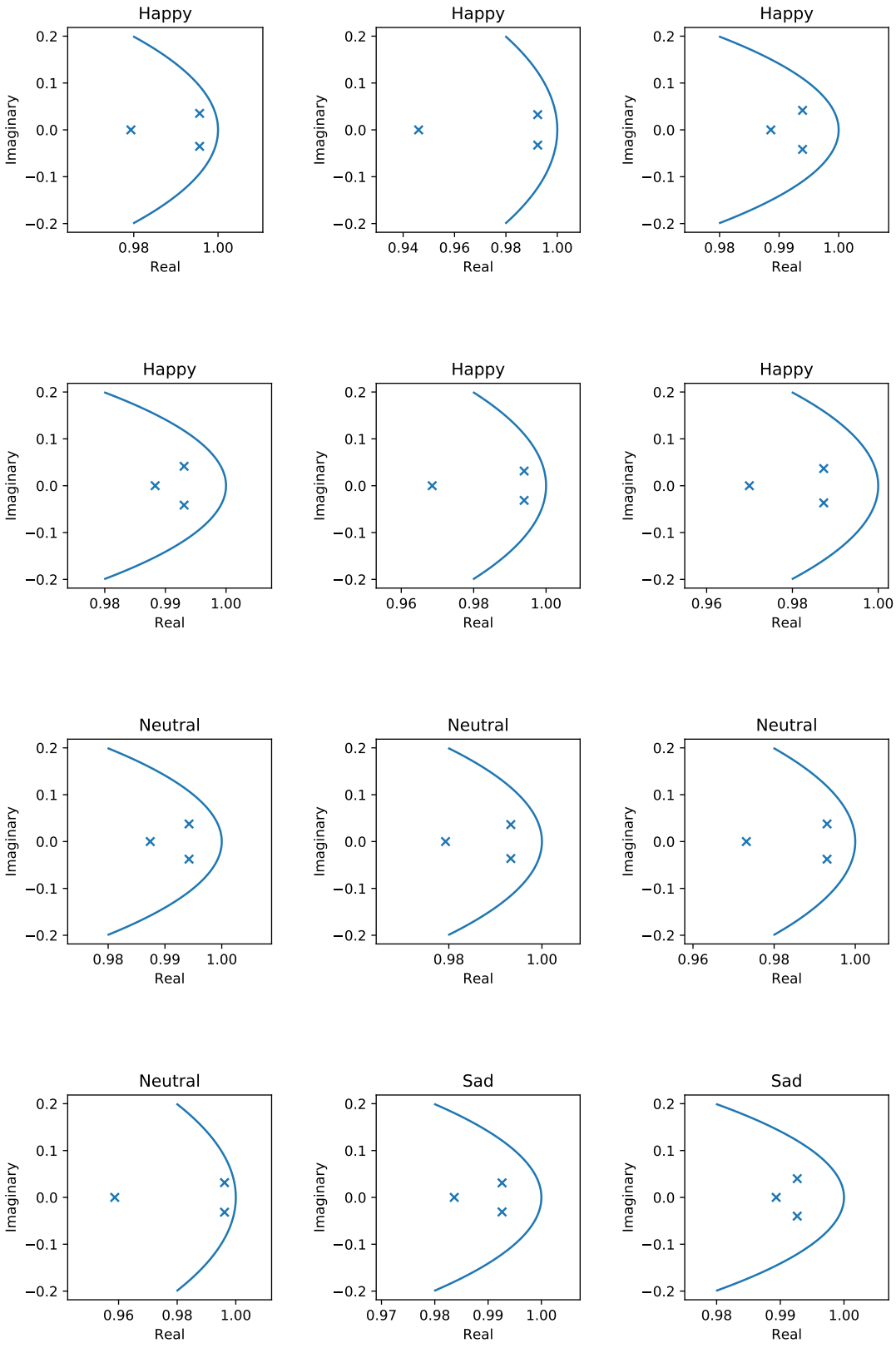
D: Linearisation eigenvalue plots

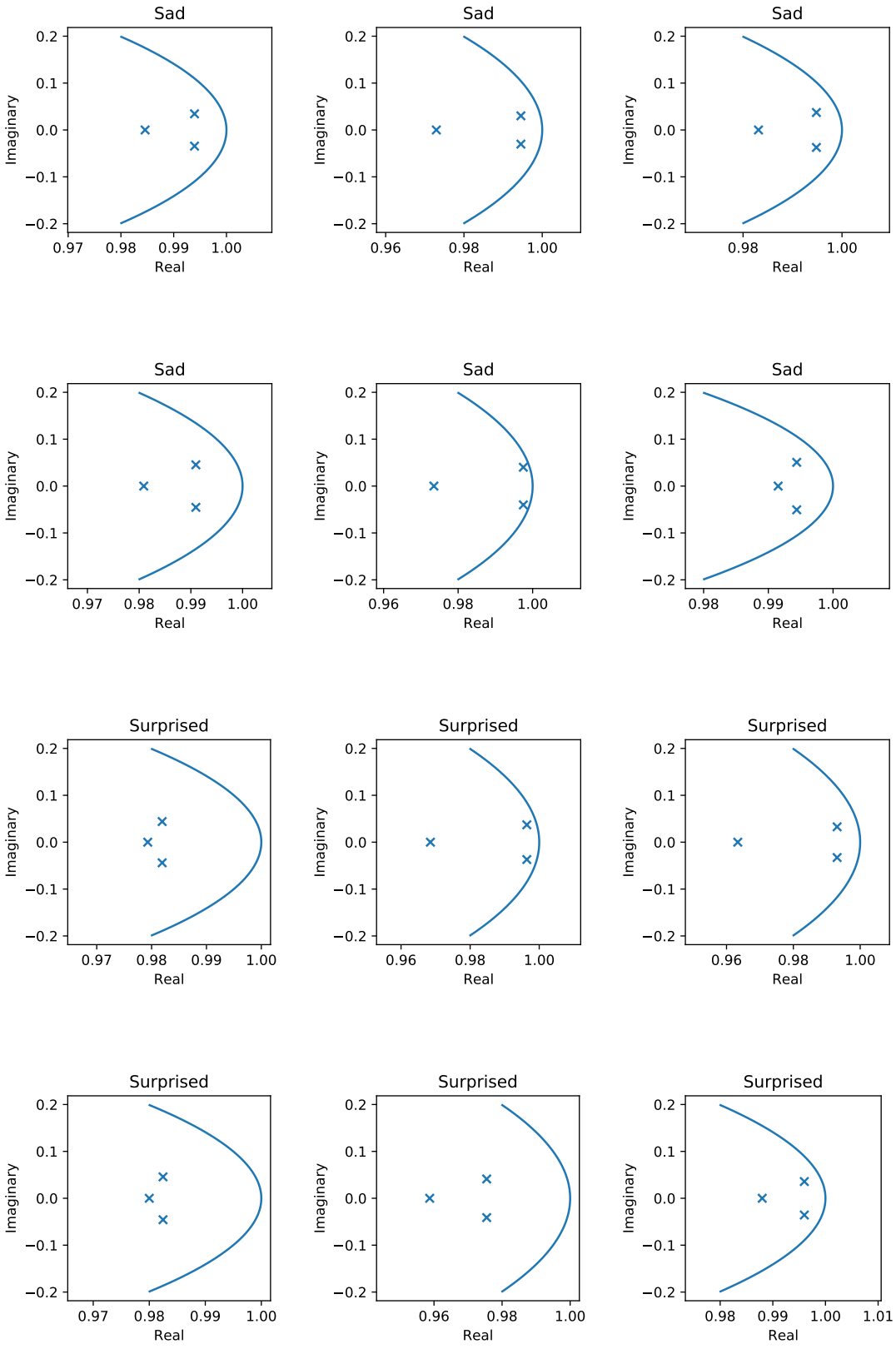
Each plot presented here shows the eigenvalues of the linear model $\mathbf{v}_{i+1} = M\mathbf{v}_i$, for delay vectors $\mathbf{v}_i$. Also shown is a segment of the unit circle. Eigenvalues within the unit circle are stable; those outside are unstable.




Calm
Calm
Calm
Calm
Calm
Calm
Disgust
Disgust
Disgust
Disgust
Disgust
Disgust
Imaginary
Real

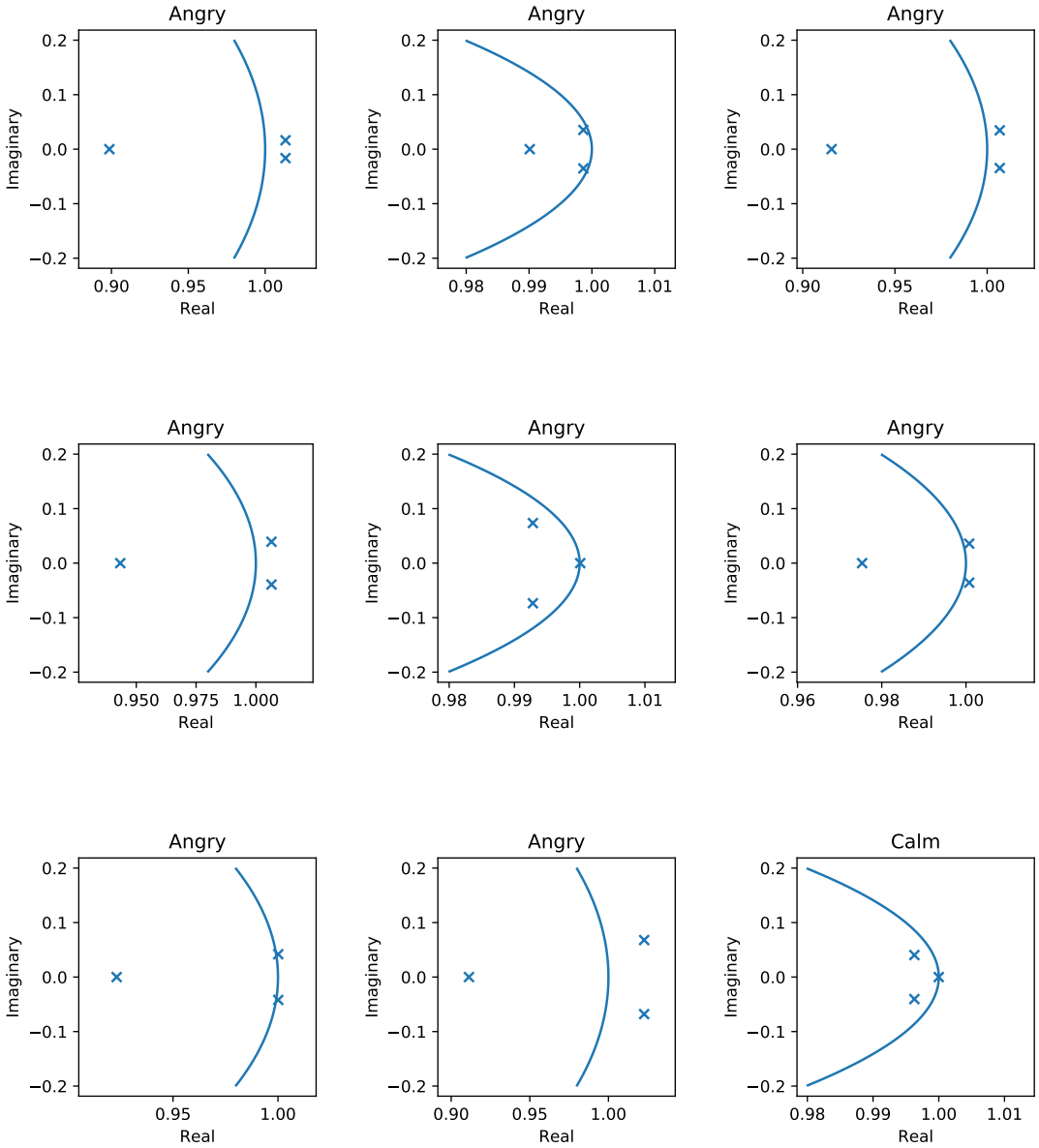

Disgust
Disgust
Fearful
Fearful
Fearful
Fearful
Fearful
Fearful
Fearful
Fearful
Happy
Happy
Imaginary
Real

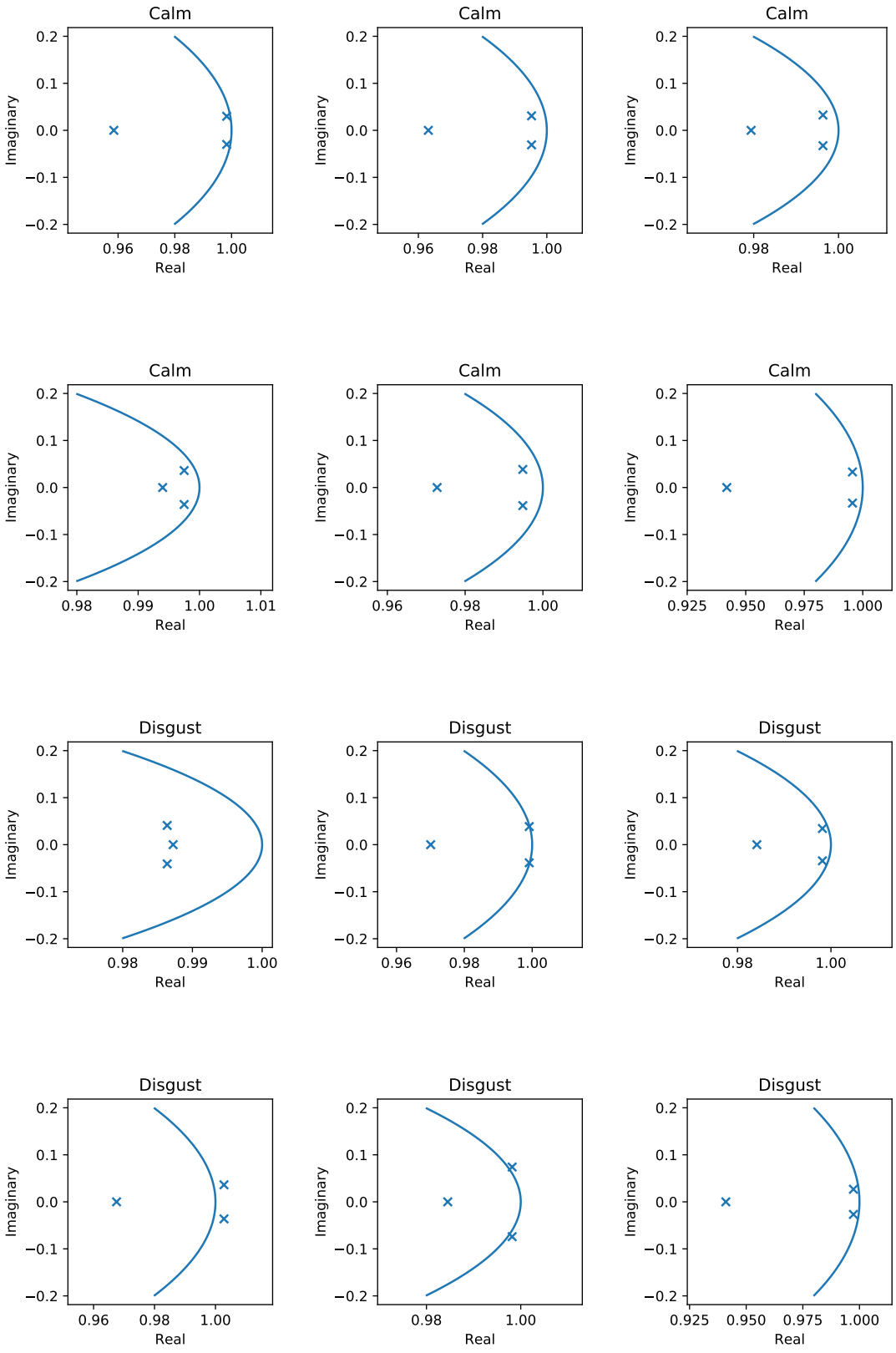

Happy
Happy
Happy
Happy
Happy
Happy
Neutral
Neutral
Neutral
Neutral
Sad
Sad
Imaginary
Real

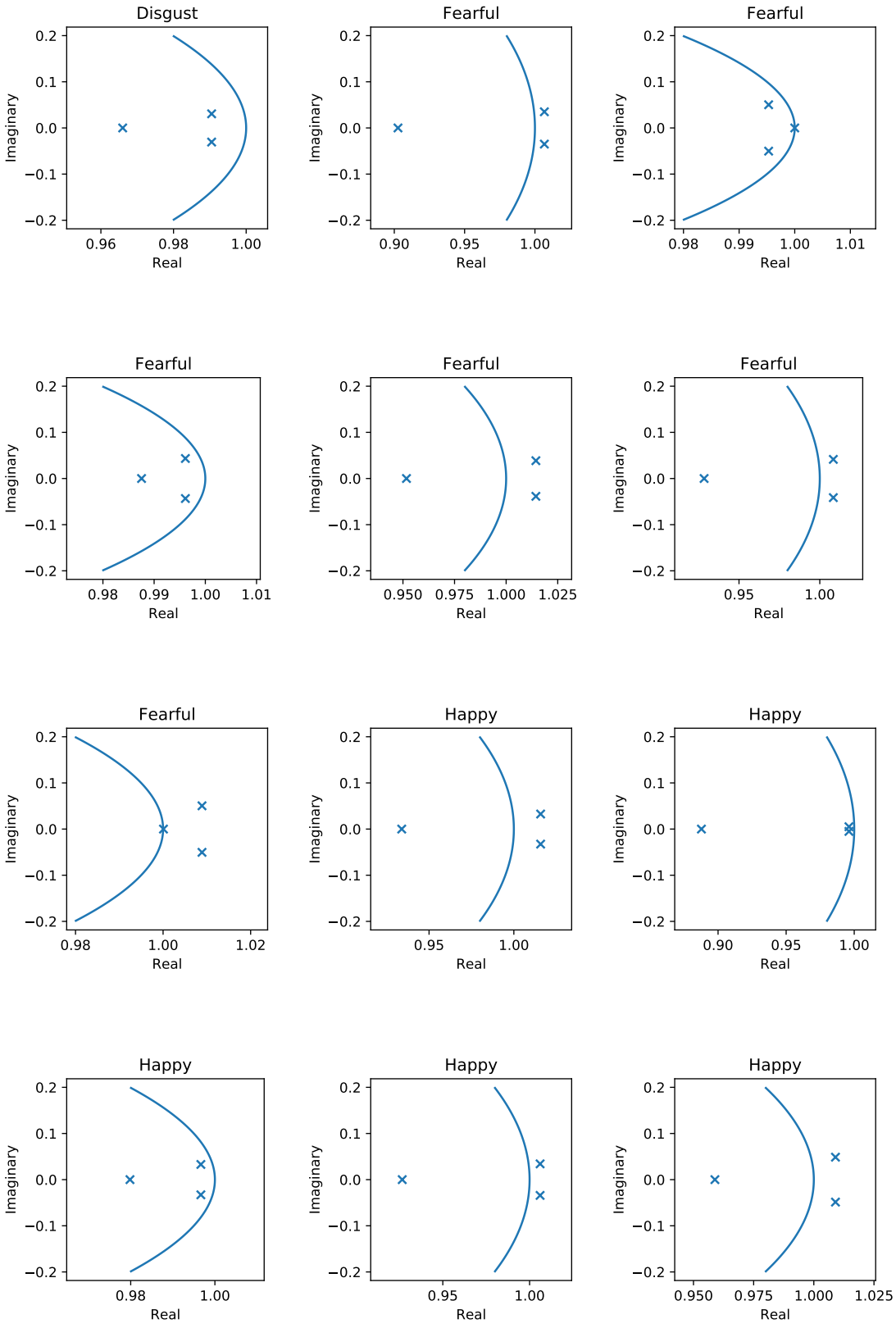

Sad
Sad
Sad
Sad
Sad
Sad
Surprised
Surprised
Surprised
Surprised
Surprised
Surprised
Imaginary
Real

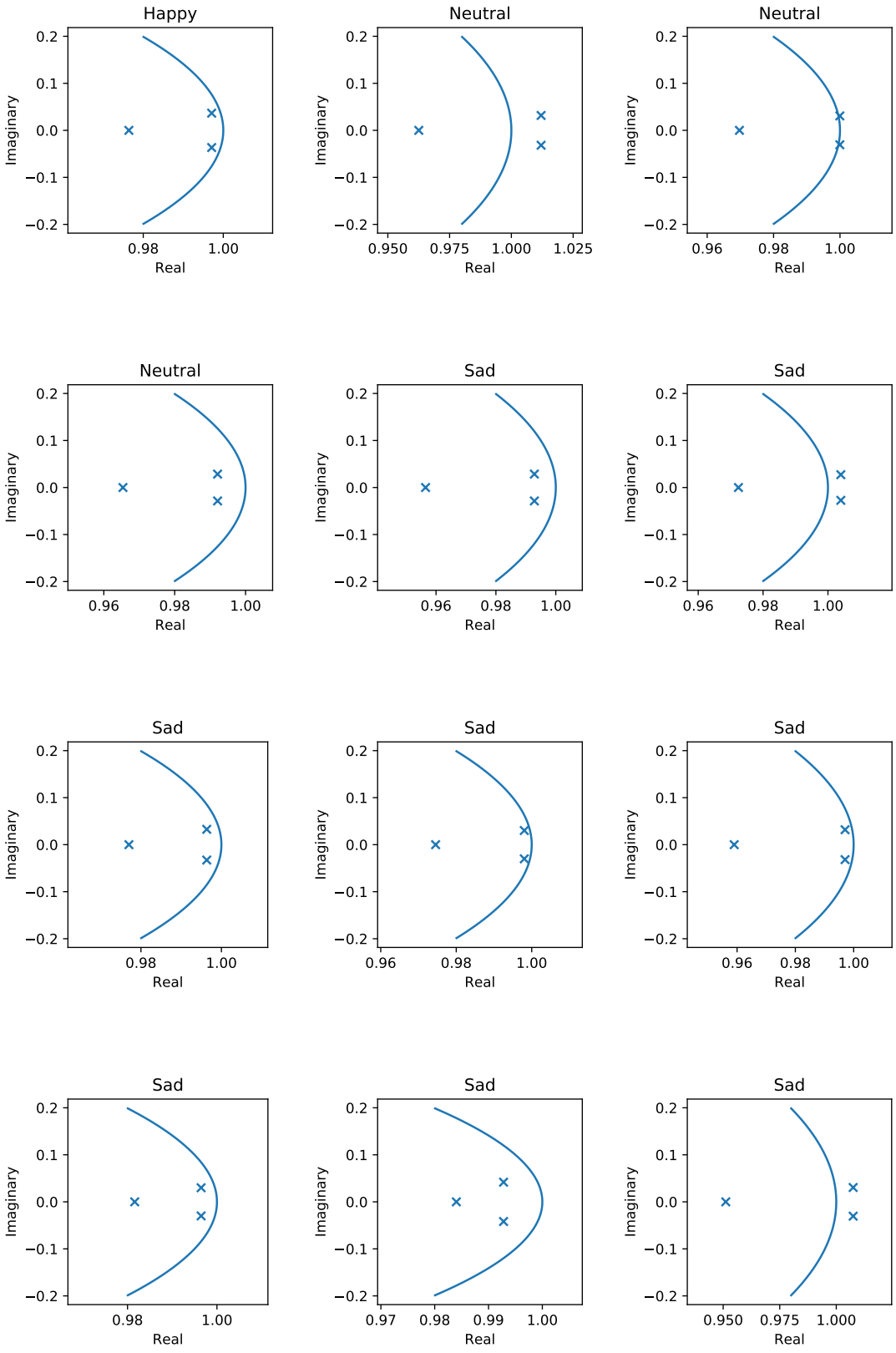
E: Nonlinear equilibrium eigenvalue plots

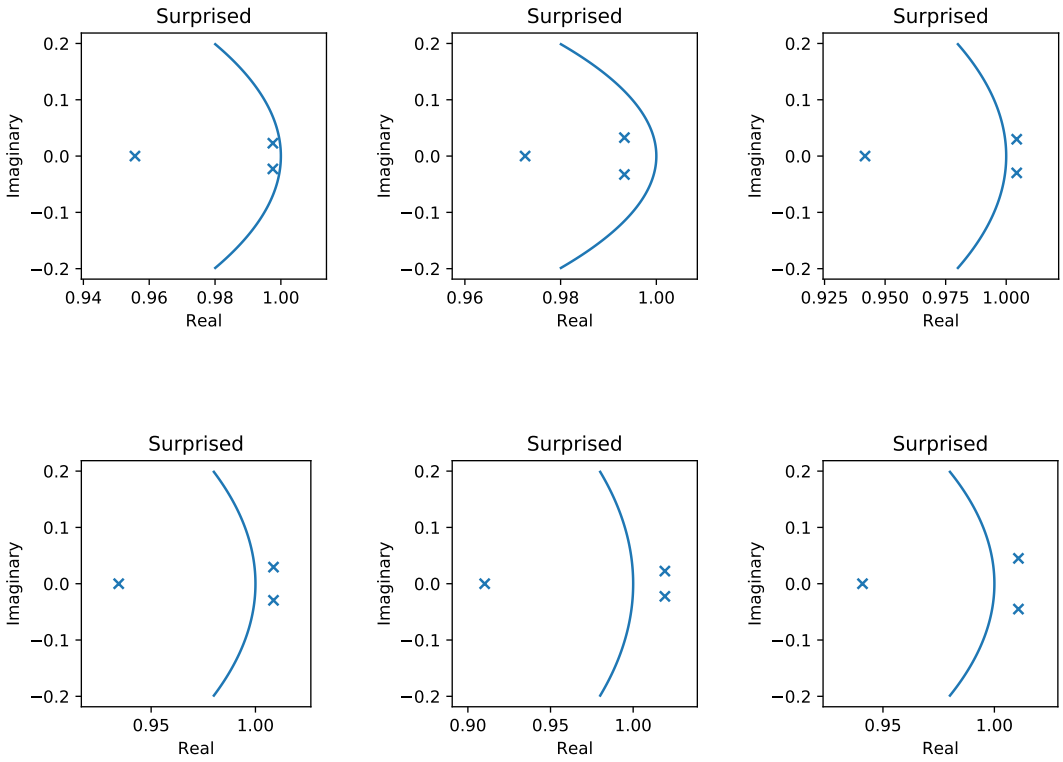
Each plot presented here shows the eigenvalues of the linearisation of the nonlinear model $\mathbf{v}_{i+1} = AP(\mathbf{v}_i)$, about a single fixed point, for delay vectors $\mathbf{v}_i$ and monomials up to degree five. Also shown is a segment of the unit circle. Eigenvalues within the unit circle are stable; those outside are unstable.




Calm
Calm
Calm
Calm
Calm
Calm
Disgust
Disgust
Disgust
Disgust
Disgust
Disgust
Imaginary
Real

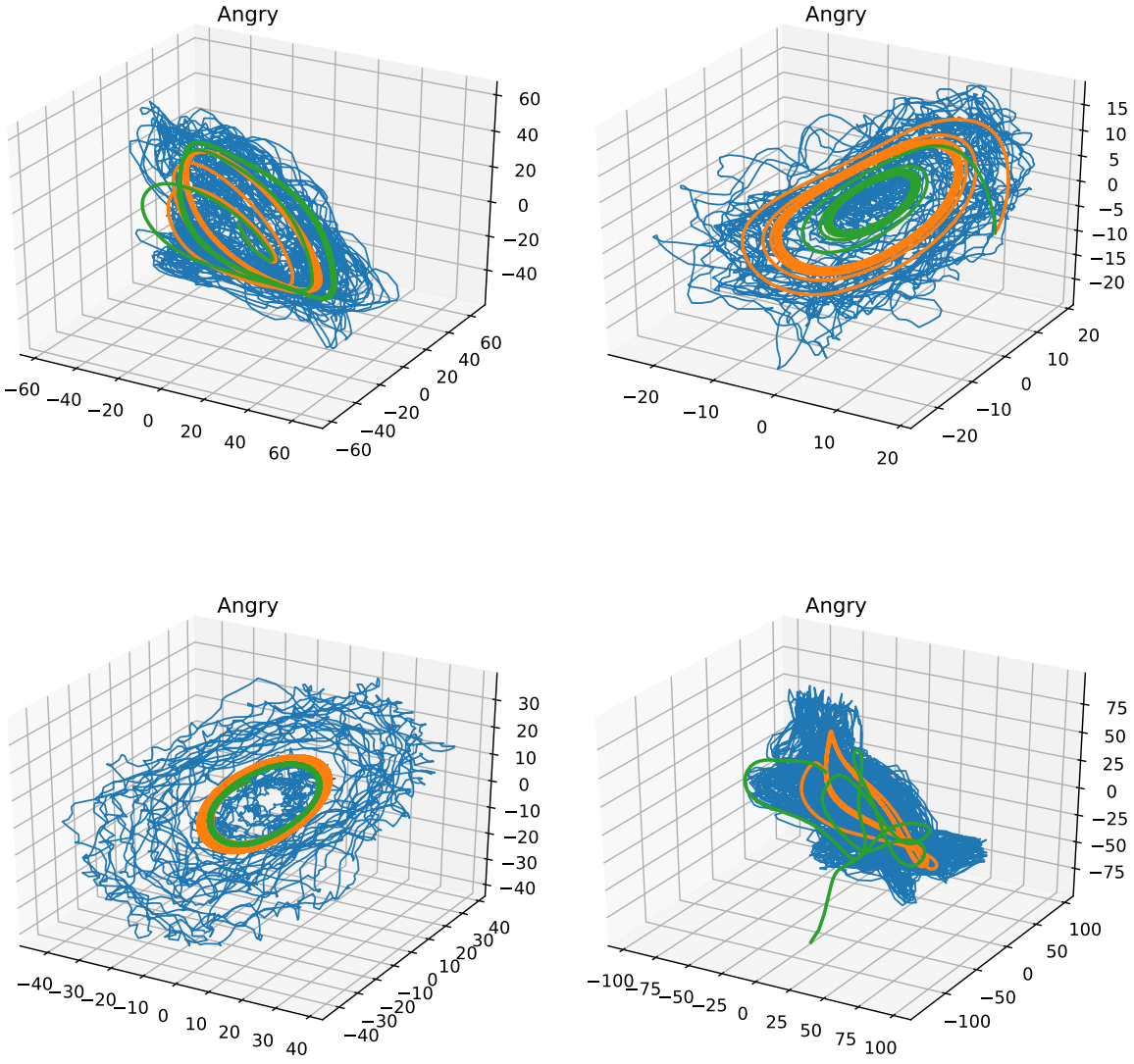

Disgust
Fearful
Fearful
Fearful
Fearful
Fearful
Fearful
Happy
Happy
Happy
Happy
Happy
Imaginary
Real

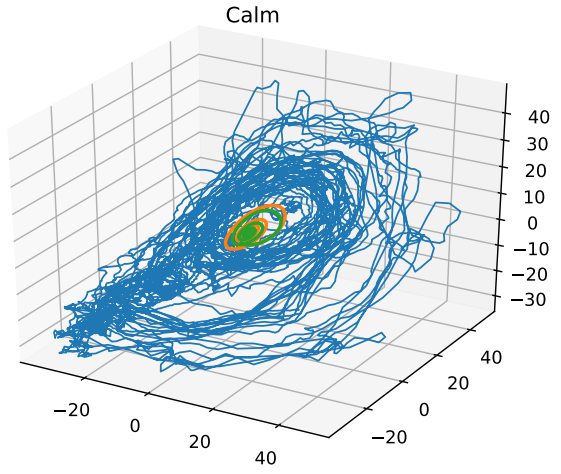

Happy
Neutral
Neutral
Neutral
Sad
Sad
Sad
Sad
Sad
Sad
Sad
Sad
Imaginary
Real

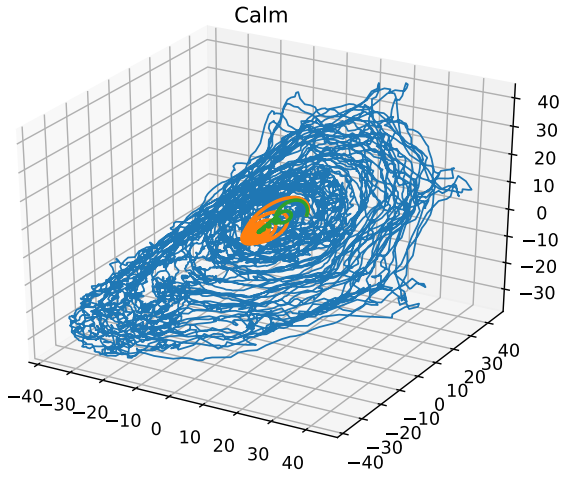

Surprised
Imaginary
Real
0.2
0.1
0.0
−0.1
−0.2
0.94
0.96
0.98
1.00
Surprised
Imaginary
Real
0.96
0.98
1.00
Surprised
Imaginary
Real
0.925
0.950
0.975
1.000
Surprised
Imaginary
Real
0.95
1.00
Surprised
Imaginary
Real
0.90
0.95
1.00
Surprised
Imaginary
Real
0.95
1.00

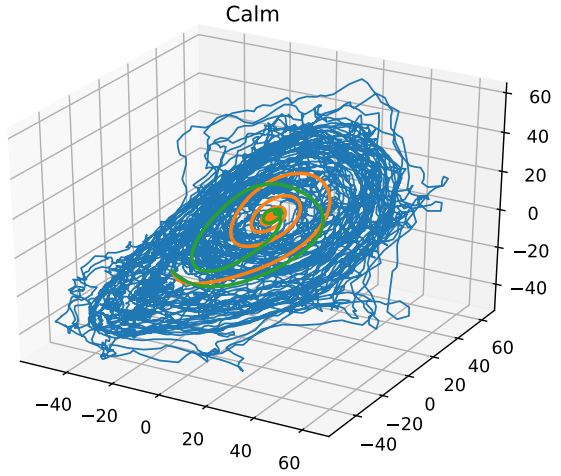
F: Comparison of nonlinear optimisation methods

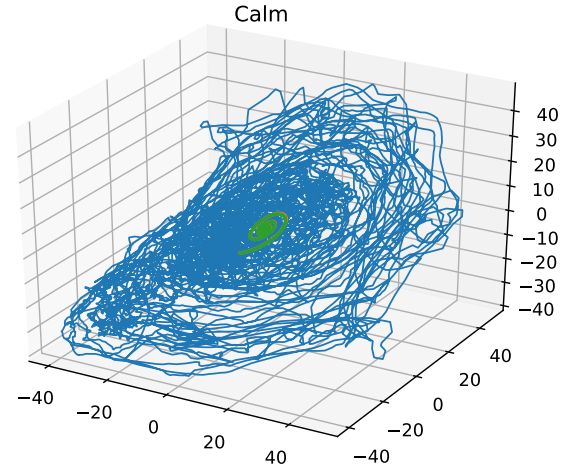
Each plot presented here shows a comparison between the locally optimised and globally optimised state evolution models, as discussed in section 2.3.2. The reconstructed trajectories are plotted (blue), as well as simulations of the locally optimised (orange) and globally optimised (green) models.

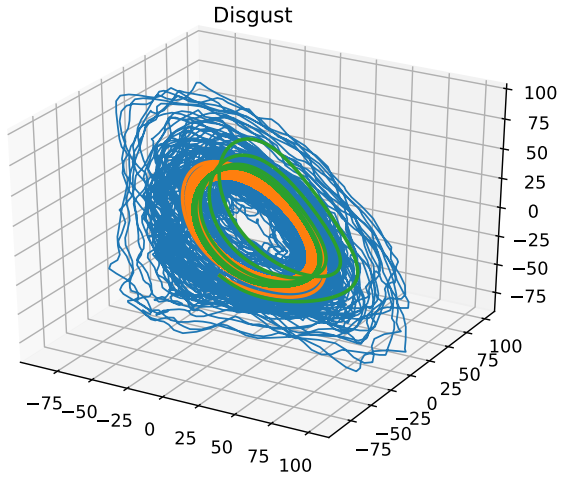


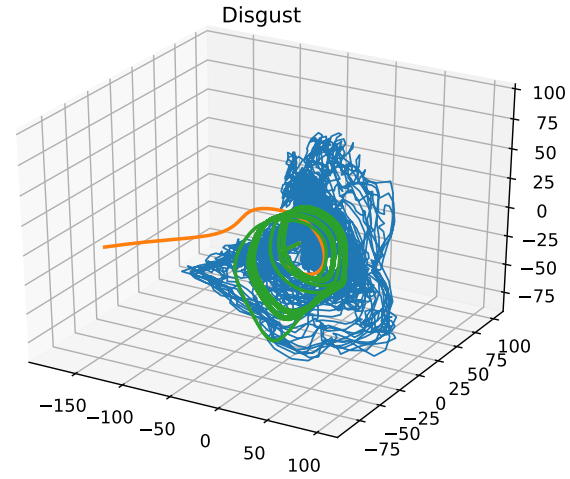

Calm

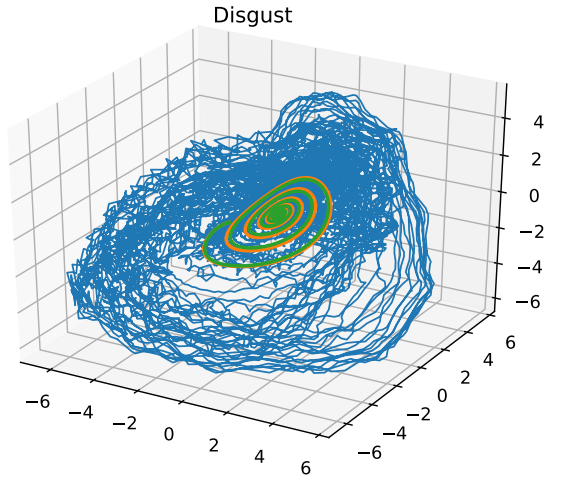

Calm

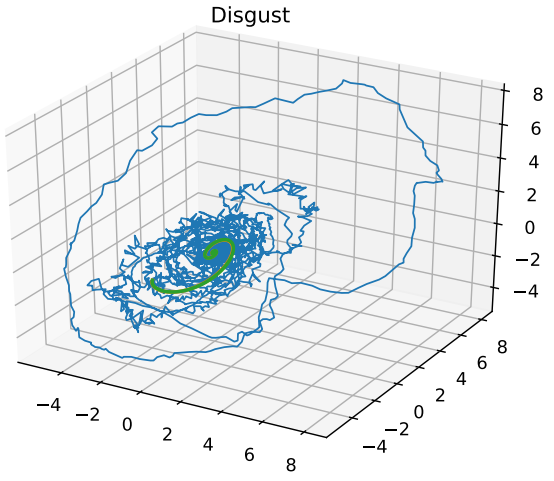

Calm

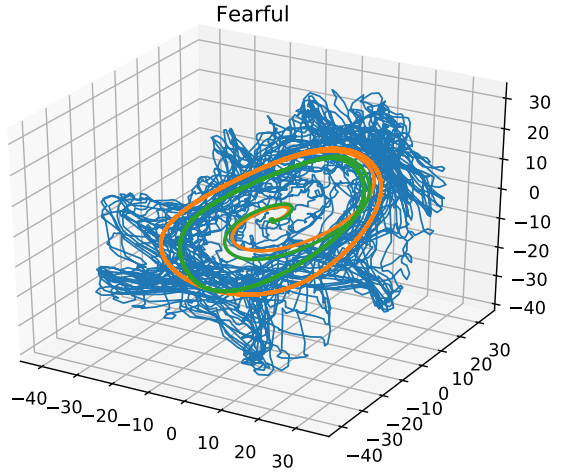

Calm

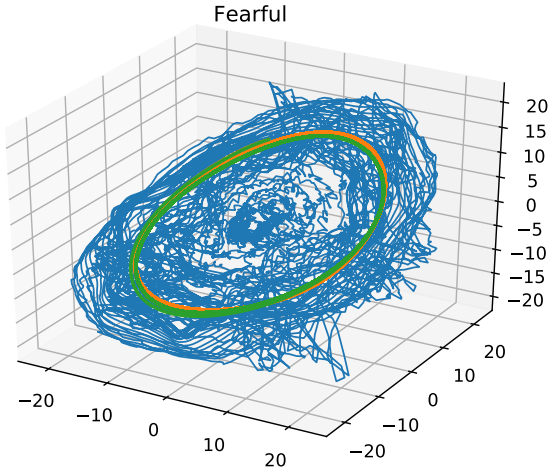

Disgust

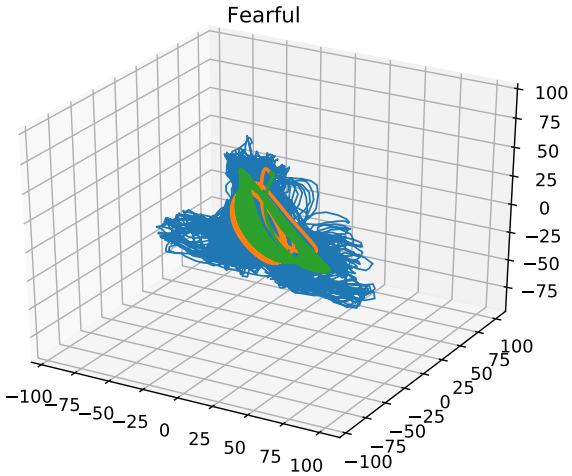

Disgust

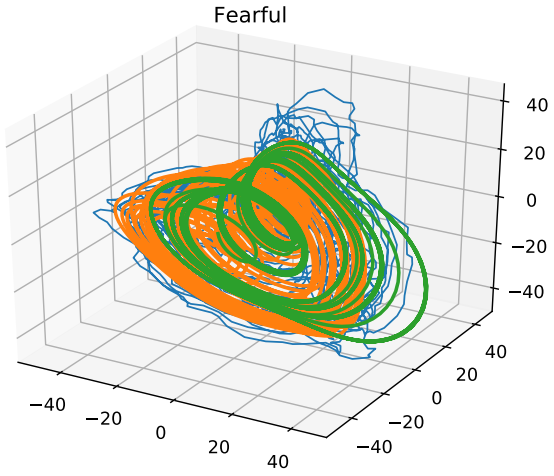

Disgust

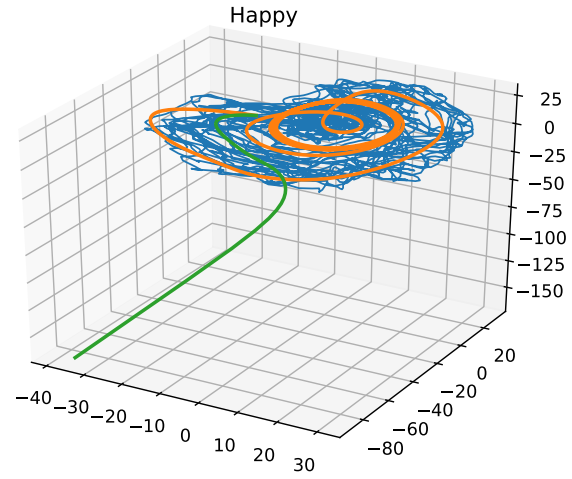

Disgust

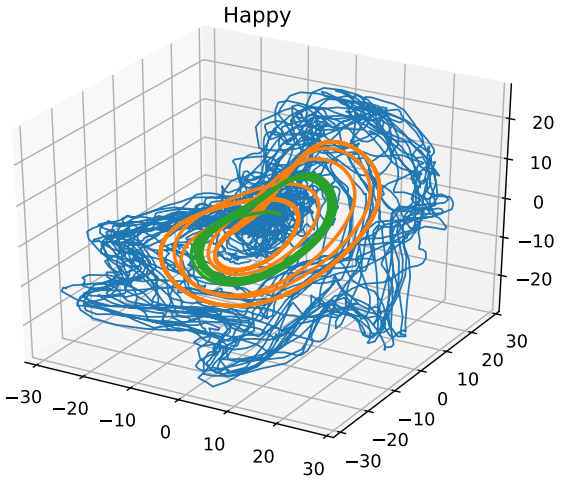

Fearful

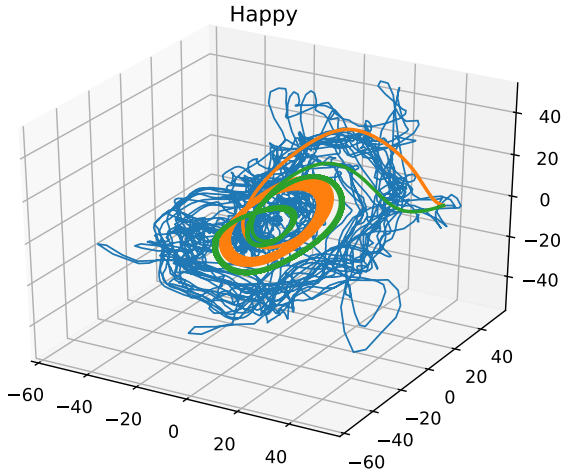

Fearful

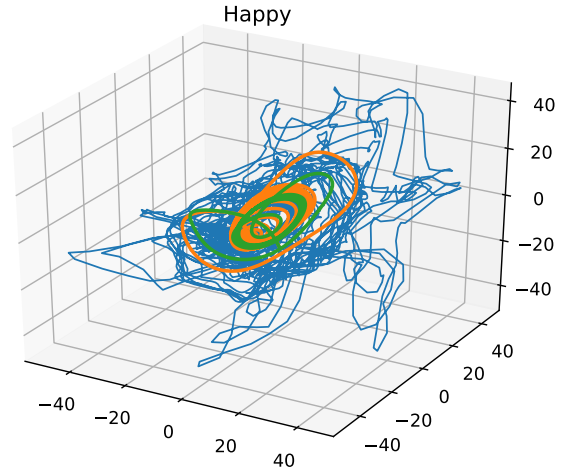

Fearful

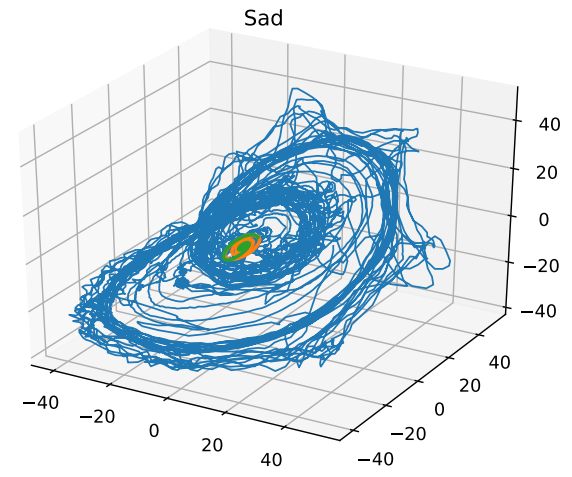

Fearful

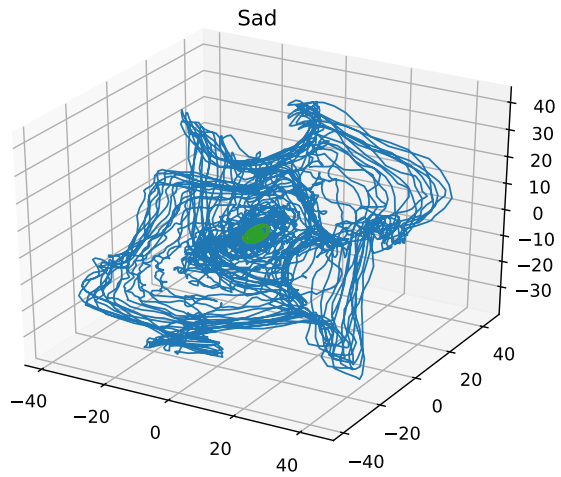

Happy
25
0
−25
−50
−75
−100
−125
−150
−40 −30 −20 −10 0 10 20 30
20 0 −20 −40 −60 −80

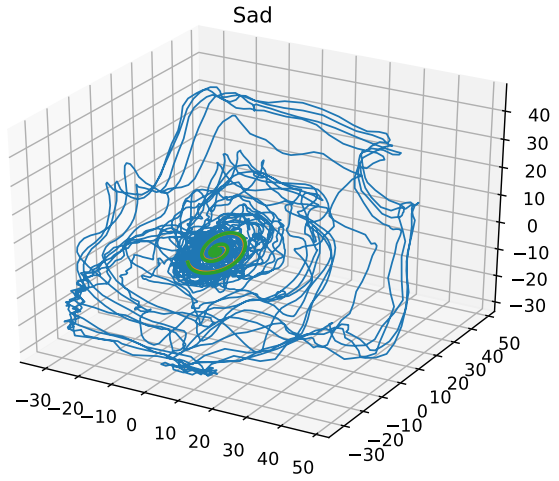

Happy
20
10
0
−10
−20
−30 −20 −10 0 10 20 30
30 20 10 0 −10 −20 −30

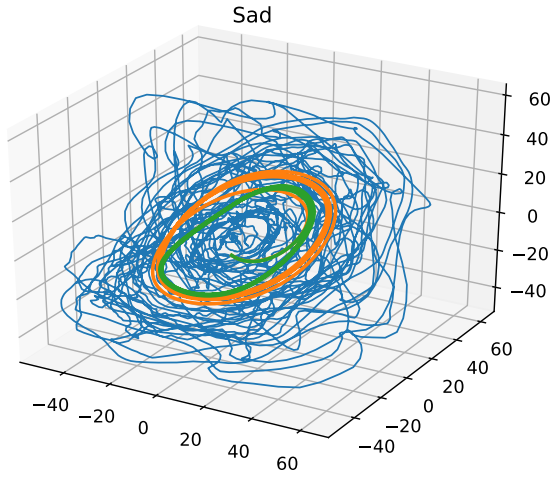

Happy
40
20
0
−20
−40
−60 −40 −20 0 20 40
40 20 0 −20 −40 −60

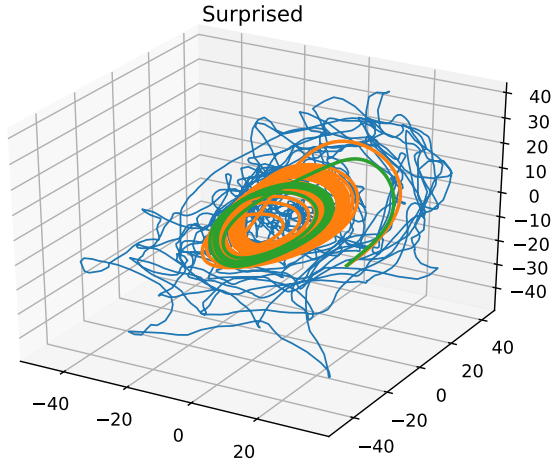

Happy
40
20
0
−20
−40
−40 −20 0 20 40
40 20 0 −20 −40

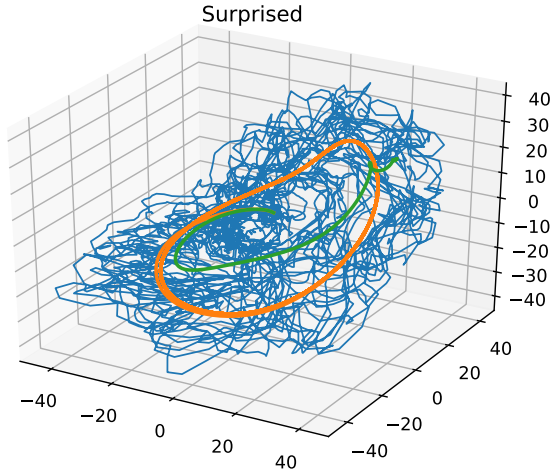

Sad
40
20
0
−20
−40
−40 −20 0 20 40
40 20 0 −20 −40

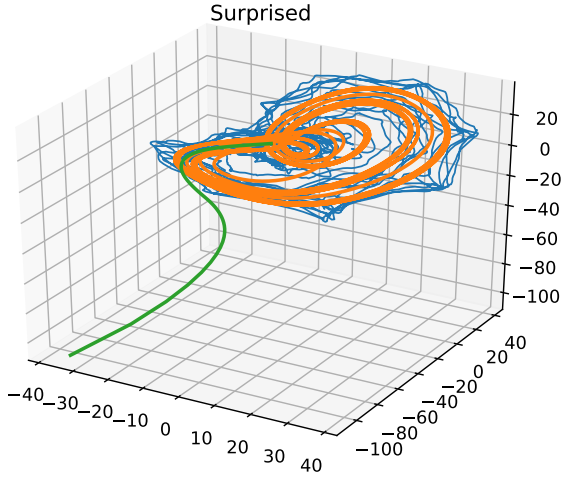

Sad
40
30
20
10
0
−10
−20
−30
−40 −20 0 20 40
40 20 0 −20 −40

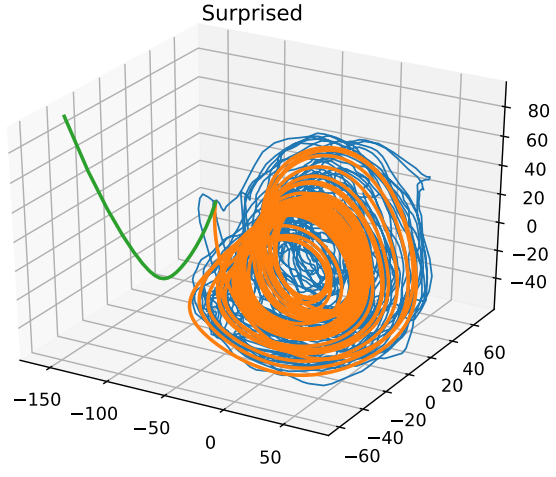

Sad


Sad


Surprised


Surprised


Surprised


Surprised