Mapping Acoustics to Kinematics in Speech

by

Rohan Bali

A thesis submitted in conformity with the requirements for the degree of Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto

© Copyright 2016 by Rohan Bali
Abstract

Mapping Acoustics to Kinematics in Speech

Rohan Bali
Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto
2016

An accurate mapping from speech acoustics to speech articulator movements has many practical applications, as well as theoretical implications of speech planning and perception science.

This work can be divided into two parts. In the first part, we show that a simple codebook can be used to map acoustics to speech articulator movements in natural, conversational speech. In the second part, we incorporate cost optimization principles that have been shown to be relevant in motor control tasks into the codebook approach. These cost optimizations are defined as minimization of integral of magnitude velocity, acceleration and jerk of the speech articulators, and are implemented using a dynamic programming technique.

Results show that incorporating cost minimization of speech articulator movements can significantly improve mapping acoustics to speech articulator movements. This suggests underlying physiological or neural planning principles used by speech articulators during speech production.
Acknowledgements

I wish to express by sincere gratitude toward Prof. Willy Wong for giving me the opportunity to work on this research and for guiding me through my master’s research as my supervisor. Your open-mindedness and patience towards new ideas has made the last couple of years both challenging and fun.

I would like to thank my supervisory committee, Prof. Luca Scardovi, Prof. Tim Bressmann and Prof. Pascal Van Lieshout. This thesis is benefited greatly from their insight.

In particular, I want to acknowledge Prof. Pascal Van Leishout for all his support and guidance. I really appreciate your patience in introducing me to speech science, your receptiveness to my ideas, and your diligence in teaching me how to frame my ideas and arguments.

I would like to thank my colleagues at UofT, both at the Sensory Communication Lab, and at the Oral Dynamics Lab. A special thanks to Aravind Namasivayam for his help in collecting data for my research.

I am lucky to have some great friends in Toronto, and I want to take this opportunity to thank them all. In particular I want to acknowledge Ellen Hayes, Shailin Gosalia and Venkatesh Nandakumar for their company, support and encouragement. I also want to acknowledge my friends back home in Al Ain and my friends from UofI.

Finally, my gratitude to my brother and to my parents is inexpressible. Thank you for encouraging me, thank you for supporting me, and thank you for your constant and unconditional love.
Contents

1 Introduction .............................................. 1

1.1 Background ............................................ 1

1.2 Previous Work .......................................... 4

1.3 Current Work ........................................... 7

1.4 Thesis Outline ......................................... 9

2 Methods for Data Collection and Analysis ............ 11

2.1 Participants ............................................ 12

2.2 Stimuli .................................................. 12

2.3 Procedure for Data Collection ......................... 13

2.4 Electromagnetic Articulography (EMA) ................ 14

2.4.1 Description of AG501 EMA ......................... 14

2.4.2 Operating Principles ............................... 15

2.4.3 Operating Procedure ............................... 17
V

2.4.4 Performance .................................................. 19

2.5 Data Processing .................................................. 19

2.5.1 Pre-Processing .................................................. 20

2.5.2 Silence Removal .............................................. 21

2.5.3 Kinematic Processing ........................................ 23

2.5.4 Acoustic Processing .......................................... 24

2.5.5 Preparation of Training and Test Dataset .................. 28

2.6 Evaluation of Mapping Techniques ............................. 29

2.6.1 Correlations .................................................. 29

2.6.2 Root-Mean-Square Deviation ................................. 30

3 Acoustic to Kinematic Mapping - Part A : Traditional Approach With Vector Quantization 32

3.1 Description .................................................... 32

3.2 Methods ....................................................... 33

3.2.1 Construction of the Codebook .............................. 33

3.2.2 Mapping Acoustics to Kinematics ......................... 34

3.2.3 Model Evaluation .......................................... 36

3.2.4 Similarity of Acoustics Within in a Cluster ............... 36
3.2.5 Similarity of Kinematics Within a Cluster ........................................... 38

3.3 Results ........................................................................................................... 40

3.3.1 RMS Errors .............................................................................................. 40

3.3.2 Correlation ............................................................................................... 43

3.4 Comparisons ................................................................................................. 46

3.4.1 Number of VQ codes ................................................................................ 46

3.4.2 Comparison to Original Method by Hogden et al [21] .............................. 49

3.4.3 Comparison to Linear Coding ................................................................... 52

3.5 Source of Prediction Errors ........................................................................... 57

3.6 Discussion ...................................................................................................... 62

4 Acoustics To Kinematic Mapping - Part B : Optimized Vector Quantization Approach ........................................................................................................... 66

4.1 Description ..................................................................................................... 66

4.2 Methods ......................................................................................................... 69

4.2.1 Processing Dataset ................................................................................... 70

4.2.2 Construction of the Codebook ................................................................... 71

4.2.3 Mapping Acoustics To Kinematics .......................................................... 72

4.2.4 Computational Considerations .................................................................. 85
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.5 Model Evaluation</td>
<td>86</td>
</tr>
<tr>
<td>4.3 Results</td>
<td>87</td>
</tr>
<tr>
<td>4.3.1 Minimum Cost Path With Separate Training and Test Datasets</td>
<td>87</td>
</tr>
<tr>
<td>4.3.2 Minimum Cost Path With Test Dataset Included in the Training</td>
<td>91</td>
</tr>
<tr>
<td>4.4 Comparisons</td>
<td>96</td>
</tr>
<tr>
<td>4.4.1 Minimum Kinematic Cost Path Compared to a Random Path</td>
<td>96</td>
</tr>
<tr>
<td>4.4.2 Cost of the Real Kinematic Path Compared to a Random Path</td>
<td>98</td>
</tr>
<tr>
<td>4.4.3 Comparison between Minimum Distance, Minimum Acceleration and Minimum Jerk</td>
<td>99</td>
</tr>
<tr>
<td>4.5 Sources Of Error</td>
<td>101</td>
</tr>
<tr>
<td>4.6 Discussion</td>
<td>102</td>
</tr>
<tr>
<td>5 Discussion and Future Work</td>
<td>104</td>
</tr>
<tr>
<td>Appendices</td>
<td>113</td>
</tr>
<tr>
<td>A Correlation of Energy in Speech Frequencies and Articulator movements</td>
<td>114</td>
</tr>
<tr>
<td>B Correlation Statistics of Articulatory Predictions Using Method in Chapter 3</td>
<td>117</td>
</tr>
<tr>
<td>C Shortcoming of Using Correlation and RMSD for Measuring Prediction</td>
<td></td>
</tr>
</tbody>
</table>
Contents

Performance 125

D Source of Error for Chapter 3 128

D.1 Time Lag 128

D.2 Non Stationary vocal tract transfer function 129

D.3 Sub-Optimal representation of acoustics 130

D.4 Simultaneous Coding 131

E Djikstra’s graph search algorithm 133

F Computational Considerations 137

G Kinematic Costs comparison 140

H Kinematic Cost Histogram - Comparison with real kinematic path 144

I Kinematic Cost Histogram - Comparison with closest to real kinematic path 147

J Completing the Kinematic Set 149
Chapter 1

Introduction

1.1 Background

Speech production is a complex task which involves the coordinated movement of different muscles in a task specific manner under restrictions of the vocal tract modulation properties, aerodynamics and bio-mechanics of oral structures [16]. Expiratory air coming from the lungs passes through the glottis and is then modulated by the vocal tract. Movements of speech articulators change the shape of the vocal tract which in turn changes the modulation properties (called the vocal tract filter function) of the vocal tract. Speech is therefore a direct consequence of the coordination between the expired air and the movements of the glottal folds and speech articulators. Speech articulators like the lips and the jaw have been adapted for speech from their original oral motor functions of chewing and swallowing [28]. Perhaps due to the fact that speech articulators have not evolved for the sole purpose of speech, the positions of individual speech articulators have complex consequences for modulation frequencies. Due to the fact that modulation frequencies are affected by the positions of multiple speech articulators, the speech production system has to maintain coordination in the movements of these articulators in order to produce the desired acoustic result. In addition to the limitations on articulator movements placed on them due to their influence on the vocal tract filter function, the speech articulator movements are also constrained by the anatomy and the bio-mechanics of the speech production system. These restrictions are seen in the low dimensionality of tongue movements and their looping pattern when producing certain
speech sounds \cite{16}. Speech articulators movements are also constrained by the aero-
dynamic conditions in the vocal tract, particularly for the planning and production of
devoicing in oral stops and sequencing of voiced/voiceless features in consonant clusters
\cite{22}. Hence, the movement of speech articulators obey various control principles to meet
the required restrictions in order to produce speech. The relationship between movement
of speech articulators and speech acoustics is therefore not straightforward.

The mapping of the movement of speech articulators to speech acoustics is called forward
mapping. Speech acoustics are a direct consequence of the properties of the vocal tract.
Although the complex relationship between the acoustics and kinematics makes forward
mapping difficult to quantify analytically, there is little debate that speech acoustics
should be, in theory, completely and uniquely determined given a vocal tract shape.
Forward mapping is often studied using either bio-mechanical models of the vocal tract
\cite{51} \cite{31} \cite{16} or using empirical speech data \cite{21}. The mapping of speech acoustics to
movements of speech articulators is called reverse mapping (also called inverse mapping
or articulator inversion). Speech planning and production are often studied from the
perspective of speech acoustics as a consequence of speech articulator movements, and
thus the form of the reverse mapping solution is less intuitive. The current work deals
with the form of the reverse mapping solution, its limitations, and its implications for
theory building in speech science.

We can gain insight into the reverse mapping relationship by studying the forward map-
ing bio-physical speech production models. Work on these models has shown that an
infinite combination of vocal tract shapes can generate acoustics with identical formant
values for a given vocal tract model \cite{4}. Hence, multiple model-based configurations of
speech articulator positions are able to produce the same speech acoustic. An empirical
investigation \cite{40} supports these model-based assumptions by observing multiple artic-
ulatory configurations that produced similar vocal tract modulation properties. This
indicates that a single valued mapping process cannot predict speech articulator posi-
tions from speech acoustics. The multivalued acoustic to kinematic solution raises some
very important questions, with perhaps the most significant one being, how do speakers
select between different kinematic configurations to produce a specific acoustic result?

The study of speech articulator movements reveals patterns in which speech articulators
often move together as being controlled in a single unit, a phenomenon referred to as
articulatory gestures in specific theoretical frameworks \cite{45}. Speakers show consistency
in the use of these gestures to produce speech phonemes. Thus, the selection of certain kinematic positions to produce specific acoustics follows some speech planning principles. We currently lack a proper detailed understanding of the origins of these gestures used to produce speech sounds. The theory of Articulatory Phonology (AP) attempts to describe the concept of gestures as having a representation at the lexical level, and in doing so treats gestures as basic units of speech phonology [6]. In the context of AP, the recovery of articulatory gestures from acoustics could also help to explain the speech perception process. Other models, like the Directions Into Velocities of Articulators (DIVA) model considers acoustics as the basis of speech production and perception. Articulatory movements in this model are seen as a direct consequence of aiming for acoustic goals instead of gestural goals. This means that articulatory inversion principles would be critical to understanding how the DIVA model can generate a particular movement pattern given a desired acoustic outcome.

The development of a computational technique that can go from speech acoustics to speech kinematics can help us gain insight into speech planning and production processes, and perhaps even the speech perception process. Since the acoustic to articulatory mapping is multivalued, an accurate solution to articulatory inversion cannot rely on the value of the acoustic frame alone. Cues from the context in which the acoustic signal is produced and the cost of using certain kinematic configurations are some of the features which may aid in reverse mapping. They form part of an accurate solution to articulatory inversion. An accurate articulatory inversion solutions may help us study the underlying kinematic units in speech, if a speaker indeed plans speech in a kinematic reference frame. They can also help us gain insight into the neural basis of speech production and speech perception.

The answers to the questions of underlying strategies in speech production are not only of theoretical interest. Understanding how we can map an acoustic signal to the underlying kinematic patterns can help improve our ability to drive facial animations using acoustic speech signals. Current implementations of talking face animations require the transmission of both the speech audio channel as well as the video channel with the facial animations. If the talking face animation video can be derived directly from acoustic input alone, this would allow the transmission of talking head speech animations at much lower bit rates. Using these principles would also aid in automatic speech recognition (ASR) technologies where a video channel can be used to complement information from the audio channel. Acoustic to kinematic mapping can be particularly useful in ASR
applications in a noisy environment where traditional ASR implementations fail [11]. The understanding of the relationship between speech acoustics and speech articulator movements may also have clinical applications in the diagnosis and treatment of speaking disorders. An example of this would be patients suffering from facial paralysis, who may benefit from bio-feedback showing them how their face intended to move based on their speech output [35].

1.2 Previous Work

Reverse mapping or Articulatory Inversion, is the recovery of speech articulatory positions from speech acoustics. Articulatory inversion has been a long-standing complex problem. Extensive research has been done in the last several decades on reverse mapping, but not only do previous works differ in their philosophical approach, they also differ in their formulation of the problem itself. This is because there is flexibility in how the acoustic and articulatory spaces can be defined, making this acoustic to articulatory mapping problem ill-formed. Phonetic gestures, location of maximum constriction in the vocal tract, area function of sections of the vocal tract as well as the positions of speech articulators are a few different representations used to define the articulatory space. The acoustic space can be defined by the time series speech representation, the vocal tract filter function, LPC coefficients, cepstral coefficients and mel-frequency cepstral coefficients, among other representations.

Many frameworks have been used to study the acoustic to articulatory position mapping. For example, biophysical models can be used to generate speech acoustics from vocal tract models. They include a lossless tube model that models the vocal tract as a number of concatenated tubes, models that parameterize the vocal tract by the location and degree of constriction of the vocal tract [51] and the Maeda model [31] which parameterizes the vocal tract by the positions of speech articulators in addition to parameters that describe the shape and size of the vocal tract. These models use physics of wave propagation in the vocal tract model to find the relationship between the vocal tract shape and the resultant acoustics. An issue with using biophysical models is that articulatory inversion results can differ drastically based on the underlying assumptions of that model. Hogden et al. [21] reviews previous works using biophysical models and discuss that articulatory inversion solutions presented by these models depends significantly on the modeling of
energy loss in these models, and the dimensionality of the articulatory and acoustic space used in the models. However, most models of the vocal tract differ in both these two features. Hence, a technique for reverse mapping that works for one model, may not work on another model, and may not work for empirical speech data. Another approach is to represent articulatory space in terms of linguistic units. An example of this is the model proposed in [7] which defines the articulatory space in terms of articulatory gestures. An issue with these higher level representations of articulatory space is that a quantitative definition of articulatory gestures is often limited to abstract task specifications related to control parameters like stiffness, damping and (theoretical) resting positions. Finally, a number of studies use the positions of speech articulators in space to represent the articulatory dimensions [21] and map them to representations of speech acoustics. This representation suffers from the issue that it is not easy to define the vocal tract shape in its entirety by only capturing the positions of speech articulators.

Several computational techniques have been proposed to solve the problem of mapping articulatory positions from acoustics (e.g., multiple linear regression models, vector quantization codebooks, neural network models, Gaussian Mixture Models and Hidden Markov Models). These techniques in their original form treat the articulatory inversion problem as having a unique solution, thus ignoring the fact that for a given acoustic signal, different articulatory configurations could be the original source (a phenomenon often referred to as motor equivalence). Examples of such techniques include Linear Regression methods like that proposed by Yehia et al. [59] where a linear model maps speech acoustic to articulatory positions. Atal et al. [4] presents a computer sorting technique where a code-book is constructed by linearly dividing the acoustic space to map an acoustic value to possible articulatory positions based on data sampled from a model of a vocal tract. Hogden et al. [21] showed that a similar code-book technique with as few as 128 codes created by vector quantization can be used to map an acoustic value to a specific articulatory value.

Modifications to the above-mentioned techniques have since been proposed to allow these techniques to treat the acoustic to articulatory mapping as one-to-many, and to arrive at multiple possible solutions to the articulatory inversion problem. For linear regression models, this can be done by dividing the articulatory space into distinct regions and constructing an acoustic to articulatory linear regression model for each of these regions [2]. A similar modification can be used for neural network models where a separate Neural Network is trained on different articulatory regions [41]. For the vector quantization
codebook, one can address the problem of multiple solutions to articulatory inversion by not averaging the articulatory position values corresponding to an acoustic cluster, and treating all these articulatory positions as possible solutions. These modifications now allow us to solve the articulatory inversion problems in two independent steps. The first step is applying techniques mentioned above to arrive at multiple possible articulatory positions as solutions to the articulatory inversion problem. The second step is finding a strategy to select the correct (or most likely) solution out of all the possible solutions. We are able to solve steps one and two separately due to the independent nature of their solutions. For example, a solution proposed to the second step (i.e. a selection strategy for paths) using an ensemble Neural Network model as the first step, will also work if the Neural Network model is replaced by a Vector Quantization model (with the one-to-many modification).

A common suggestion in selecting a solution to the articulatory inversion problem is to impose a smoothness constraint on the viable solutions. The motivation for this is that speech articulators have been observed to show a tendency for movements which utilize low energy [29]. Speech articulators obey certain constraints on how fast they can change direction over time, how fast they can move and how fast they can accelerate. Thus, of all the possible ways in articulatory space that one could produce a sound, one could select an articulatory path that maximizes some measure of smoothness. Work by Neufeld & Van Lieshout [36] show that the movement of the tongue relative to the palate indeed follows a path that minimizes jerk. This idea of minimizing a cost associated with kinematic movements has also been suggested in other works of human motor control. It has been shown that the coordination of hand movements in reaching tasks obeys principles of minimization of jerk [12]. The smoothness constraint in speech articulation was introduced by [4] where solutions to the articulatory inversion were limited to smoothly changing vocal tract area functions. Since then, a number of techniques have incorporated a smoothness criteria to narrow down to an exact solution for articulatory inversion. Sorokin and colleagues [49] proposed a curve fitting method that prevents the inverse solution to deviate too far away from the neutral position of the speech articulators. Toda and colleagues [55] used a Gaussian Mixture model for articulatory inversion and incorporated the statistical trajectories of the articulator movements. Richmond [43] used a mixture density network model for estimating maximum likelihood trajectories with constraints on the articulators position and movement. Others, like Schroeter and Sondi [47] presented a dynamic programming (DP) method based on the Dijkstra's graph search algorithm that could find the optimal path (with represented frames of articulatory
movements) of minimum cost (where cost was defined as a sum of articulatory movement costs between subsequent frames) among many such paths. This DP technique has been used in subsequent work to implement articulatory inversion solutions with articulatory constraints. Finally, Richards et al. [42] used an articulatory codebook as well as a neural network model to arrive at a set of possible solutions to articulatory inversion, and used the DP technique with articulatory and acoustic costs to arrive at a unique solution. Taken together, these works show improvement in articulatory inversion predictions by imposing minimum movement constraints.

A different approach to finding the smoothest kinematic path is proposed by Ghosh and Narayanan [17]. Their technique uses the fact that solving for minimum distance, acceleration or jerk paths, is the same as solving for the path of minimum power when passed through certain high pass filters. While this method greatly simplifies the problem of finding the path that satisfies many definitions of smoothness, it relies on an assumption that the kinematic to acoustic relationship is linear for small kinematic movements. However, this assumption of linearity may not always hold true as there are certain kinematic positions where the acoustic-kinematic relationship is sensitive to small changes. This concept is shown in the quantal theory of speech [52] which suggests that the kinematic to acoustic relationship is non-linear, and that certain kinematic configurations are unstable, i.e. small changes to kinematic positions may cause significant acoustic changes.

1.3 Current Work

The vector quantization method proposed by Hogden [21] shows promising results for articulatory inversion. Although Hogden’s work ignored the one-to-many nature of the acoustic to articulatory mapping, they were able to show that articulatory positions could be predicted with high accuracy by using the average articulatory configuration used to produce that acoustic output. Hogden used a dataset with utterances containing two vowels spoken in a /g/ context with vowel pairs selected from a set of ten vowels. The dataset they used contained 90 utterances, which is a very restricted dataset. In our work, we will show that the vector quantization codebook method proposed by Hogden [21] can also be used for predicting articulatory positions from acoustics for a dataset containing naturally spoken phonetically diverse sentences. We also propose an explanation why,
Hogden’s method gives good results even though it ignores the one-to-many nature of the acoustic to kinematic mapping.

The second path of our work deals with articulatory inversion when the acoustic to articulatory mapping is treated as one-to-many. Hogden’s vector quantization method clustered the acoustics using vector quantization, and averaged the articulatory configurations corresponding to specific acoustic values in a cluster. In contrast, our approach in Chapter 4 will treat all these articulatory configurations as viable solutions, and explore principles leading to the correct articulatory configuration from the set of possible articulatory configurations.

One suggestion to select the correct articulatory configuration is to introduce a cost associated with these configurations, and to pick configurations with the lowest cost. Many cost constraints related to the smoothness of kinematic movements have been proposed in previous works, as described above. These include costs related to the range of movement (minimum movement constraints) and the amount of acceleration or jerk of motion. The path of minimum cost under these constraints can be found by considering articulatory configurations as nodes of graph, and finding the path of minimum cost. A technique to find paths minimizing the cost related to kinematic movements is presented by Schroeter and Sondhi [47]. Work on empirical data using this technique [42, 26] has shown to improve articulatory inversion predictions. As mentioned above, work on other human motor control tasks like reaching tasks of the arm have shown that the arm follows a path of minimum jerk [12]. There is also evidence to show that the movements of speech articulators, in particular the tongue, also follow a path of minimum jerk between target positions [36]. However, so far no studies on empirical data have examined the performance of minimum kinematic costs constraints on the articulatory inversion problem.

The current work examines the articulatory inversion solutions using minimum movement (minimum integral of magnitude of velocity), minimum integral of magnitude of acceleration and minimum integral of magnitude of jerk constraints. We do this by modifying the graph search algorithm proposed by Schroeter and Sondhi [47]. Applying these constraints to predictions from the same dataset allows us to make a comparison between the prediction performances of each of these constraints. Results presented in this thesis shows evidence that the articulatory path chosen by a speaker is optimized for each of the three constraints mentioned, and in particular for acceleration and jerk. We
discuss implications of this kinematic cost optimization shown during speech on speech science. We also discuss limitation of this methods that arise due to incompleteness of the dataset. Hence, we suggest techniques future works may wish to explore in order to complete the dataset, and to ensure the robustness of this method for articulatory inversion.

Hence, this thesis contains two studies. The first study (in Chapter 3) analyzes the performance of Hogden’s vector quantization codebook approach [21] on a dataset of naturally spoken and unrestricted speech. This was done to show that a vector quantization clustering could be used for articulatory inversion on a large and unrestricted dataset. The second study (in Chapter 4) modifies the vector quantization codebook approach and incorporates different minimum kinematic cost constraints. We compare the improvement in kinematic predictions offered by incorporating these kinematic constraints by comparing the performance of different constraints to each other, and by comparing them to the one-to-one codebook method shown in Chapter 3.

1.4 Thesis Outline

Following this general introduction, Chapter 2 describes the methodology of the studies shown in the current work. These studies use empirical speech acoustic and kinematic data collected using an AG501 Electromagnetic articulography (EMA) machine. The principles of working of the EMA machine and the setup used to collect data from it are discussed in this chapter. We also describe the participants of this study and the speech stimuli. Data from the EMA machine is processed before it is suitable to use for our studies. Methods for acoustic and kinematic data processing, as well as silence removal are discussed in detail. Two studies are presented in this thesis, the first showing the performance of Hogden’s work on a bigger, less restricted dataset, and the second showing the prediction performance of a Vector Quantization method when using strategies of path selection with kinematic cost optimization. These methods will be evaluated based on their prediction performance. The method for obtaining predictions and the preparation of the test dataset are different for each of the studies, and hence these elements are discussed separately in chapters 3 and 4. However, the metrics of evaluating prediction accuracy are the same for both the studies and will be discussed in chapter 2.
Chapter 3 presents a study where we evaluate the performance of Hogden’s [21] vector quantization codebook (VQ) method for reverse mapping on a larger and less restricted dataset. This chapter contains methods for preparation of the VQ codebook, methods for prediction evaluation, the predictions results found using this VQ codebook, an evaluation of the properties of the codebook, and a comparison to prediction results presented by the original Hogden study.

Chapter 4 presents a study where we evaluate reverse mapping taking into account the one-to-many nature of the acoustic to kinematic relationship. This method builds upon the VQ method presented by Hogden to construct a codebook detailing the association between acoustic values and a set of possible kinematic values that could produce those acoustic values. In order to determine the optimal solution from this set of possible reverse mapping solutions, we introduce kinematic costs for kinematic configurations and define a reverse mapping solution that minimizes this kinematic cost. The kinematic costs we evaluate are integral magnitude velocity (i.e. total distance), integral magnitude acceleration and integral magnitude jerk. This chapter describes methods to define a one-to-many vector quantization codebook, how to determine a reverse mapping solution that is optimized for kinematic costs, the method for prediction evaluation, prediction results found using each of the cost solutions, and a comparison to the predictions reported in chapter 3.

Chapter 5 discusses the results presented in chapter 3 and chapter 4 and their implications for theory building and more practical implementations.
In studying the relationship between speech acoustics and speech articulator kinematics, it is necessary to get simultaneous and accurate measurements from each of these domains. In addition to acoustics of speech which is recorded by means of a microphone, a variety of systems can be used to track the motion of articulators. Some studies have used an ultra-violet (UV) light source based video recording system to track the position of markers placed on the face. However, this method has the limitation that it cannot be used to capture the movement of speech articulators hidden inside the mouth, e.g., tongue position. 3D MRI imaging techniques can also be used to capture the articulator movements during speech, but a major limitation in using this technique is the cost associated with running MRI sessions. Ultrasound techniques for capturing the shape and movement of the tongue are increasing in popularity for their ability to measuring detailed tongue shapes in real time in a non-invasive manner. However, while this technique can capture tongue data with high dimensionality, it faces challenges when capturing the movement of other speech articulators like the jaw. Finally, some studies have used X-Ray to extract vocal tract shapes during speech. However, this technique is limited due to its hazardous nature.

For this study, speech articulator kinematic data were collected using a 3D electromagnetic articulograph (EMA, AG501). This system is based on the original three transmitter EMA machine described by Perkell in 1992. Similar articulograph systems have
been used in previous studies studying speech [59, 21]. The AG501 machine works on
the principle of electromagnetic induction and has the ability to measure the position of
16 points in three spatial and two angular (pitch and yaw) dimensions [60]. It has the
ability to capture the movements of speech articulators, including the ones inside the
mouth, with high spatial and temporal accuracy.

The acoustic data are windowed and liftered (a technique for removing pitch contributions
discussed in detail in 2.5.4) in order to transform them into a form that is more closely
related to the shape of the vocal tract. This process is explained in more detail in section
2.5.4. The position data of each articulator is averaged for the duration of the acoustic
window to produce a pseudo-static representation of their movement within that window
of time. Positions of articulators are corrected for the movement of the head or the jaw.
This process is explained in section 2.5.3.

2.1 Participants

A total of three speakers (1 male, 2 female) participated in this study. Participant A was
a 45 year old male and spoke Chinese as his first language but spoke English fluently with
no discernible accent. Participant B and C were 25 and 22 year old females respectively
and spoke English as their first language. None of the participants self-reported any
speech, language, or hearing problems and all gave consent before participating in this
study. Participants were compensated $20 for their participation (with the exception of
participant A who was the author’s graduate supervisor).

2.2 Stimuli

The dataset used for this experiment consisted of phonetically balanced sentences from
the ATR British English Database [3]. This database contains 200 sentences, each with
between 10-20 words. Each participant was presented with the sentence list a few hours in
advance and asked to read the sentences silently so that they could familiarize themselves
with the words. Since the goal of the study was to examine phonetically diverse speech
with few restrictions, the participants were asked to speak these sentences in a natural
2.3 Procedure for Data Collection

The 200 sentences were broken down into 20 sets, each containing 10 consecutive sentences. Thus, the first set contained sentence 1-10, the second set contained sentence 11-20, and so on. Each of these sets was presented one at a time to the participant on a computer monitor placed approximately 1 meter from their face. The font of the sentences was adjusted to the participant’s level of comfort.

Once the participant finished reading all sentences in a set, recording was turned off and the next set was presented on the computer screen before restarting recording. This allowed for a pause of roughly 30 seconds between sentence sets. If the participant made a mistake in speaking a sentence, they were instructed to finish reading the sentence with the mistake, and repeat it once again immediately afterwards. A note was made about the location of these misspoken sentences so that they could be removed in post-processing. Once the participant had finished speaking the 20 sets, they were instructed to speak the 20 sets again starting from the first set (i.e. sentences 1-10), for a total of two complete recordings of the dataset. The entire recording process took 60-90 minutes per participant.

The movement of speech articulators was tracked in the front-back (x axis) and up-down (z axis) directions. The angle of elevation of the tongue tip was also captured for one participant (subject C). Additionally, movements of the head were also captured in order to calculate measures of speech articulator movement free from head motion influences.

An AG501 3D electromagnetic articulograph (EMA) machine was used to record the participant’s acoustics and movement of speech articulators as they spoke sentences from the dataset. A detailed description of the AG501 is provided in the next section.
2.4 Electromagnetic Articulography (EMA)

2.4.1 Description of AG501 EMA

We used the AG501 3D Electromagnetic Articulography (EMA) machine made by Carstens Medizinelektronik GmbH to simultaneously record acoustics and movement of speech articulators. This machine has the ability to measure the movement of up to 16 coils (which are glued on to the surface of speech articulators) with a sampling rate of up to 1250 Hz (although we use a lower rate of 250 Hz) and an error of less than 0.5mm in the three spatial dimensions [60]. The EMA machine also records acoustics with an inbuilt sound card sampling at 48 kHz. Figure 2.1 shows the AG501 EMA machine.

![AG501 3D electromagnetic articulograph](image)

Figure 2.1: AG501 3D electromagnetic articulograph
2.4.2 Operating Principles

The EMA machine tracks the position and orientation of electromagnetic coil which are glued on to relevant anatomical positions of the participant. Transmitter coils create a changing electromagnetic field around the participant’s head. This changing electromagnetic field induces a voltage in receiver coils (which are conducting wires wound in a loop) in accordance with Faraday’s and Lenz’s law as shown in equation (2.1). In this equation, \( v \) is the voltage of the induced signal, \( N \) is the number of turns on the coil and \( Flux \) is the magnetic flux through the coil’s area.

\[
v = -N \frac{d}{dt} * Flux
\]  

(2.1)

Work by [46] has shown that the voltage induced is inversely proportional to the cube of the distance between the transmitter and the receiver. Each transmitter induces a voltage at a different frequency, and so the contribution to the induced voltage from a particular transmitter can be measured separately. By knowing the distance of the coil from three transmitters, the position of the receiver coil can be triangulated. This concept is illustrated in figure 2.2. A hypothetical sphere is drawn centered on a transmitter coil with radius equal to the distance of the transmitter to the receiver coil. All points on the surface of such a sphere represent possible locations of the receiver coils. If the distance from three transmitters is known, then the two points of intersection of three such spheres represent possible locations of the receiver coils.
Figure 2.2: Triangulation of the position of the receiver coil given distance from three transmitters. Spheres are centered at the transmitter with radius equal to the calculated distance to a receiver coil. Two spheres intersect at a circle (shown in blue) and three spheres intersect at two points (show in yellow). These two yellow points represent possible locations of the receiver coil.

However, the position of the receiver coils cannot be calculated from the induced voltages from three transmitters alone as the orientations of the coils also affects this induced voltage. The receiver coil has six degrees of freedom, three spatial (in x, y and z dimensions) and three angular (pitch, yaw and roll). Of these, pitch (θ) and yaw (φ) reduce the induced voltage by a factor of \( \cos(\theta) \) and \( \cos(\phi) \) respectively. Thus, there are five relevant independent unknown variables (spatial dimensions x, y and z, and orientation angles for pitch and yaw).

The AG501 machine we used in this study uses six transmitters placed in a configuration described in Andreas Zierdt in [61]. The six known variables (induced voltages due to the six transmitters) are sufficient to calculate the five unknown variables (three positions and two orientation) of the receiver coil. Because the field equations are non-linear, the solutions for the five unknowns are computed numerically using a modified version of the Newton algorithm [61].
2.4.3 Operating Procedure

Before the receiver coils are glued on the speaker’s speech articulators, the AG501 machine has to be calibrated. During calibration the machine determines the field model and the relationship between the induced voltage and electromagnetic field for each sensor. This relationship is sensitive to coil properties among other factors and thus as a precaution, the machine is re-calibrated before every session.

Sixteen receiver coils are glued on the speech articulators of the participant. Of the coils that are attached, the ones of interest are the tongue tip (TT), tongue dorsum (TD), tongue back (TB), lower lip (LL), upper lip (UL) and Jaw (JW). The TT coil is attached 1 cm from the tip of the tongue, the TF coil is attached 4 cm from the tip of the tongue and the TB coil is attached 5.5-6 cm from the tip of the tongue.

Additionally, coils on the nose, behind the left ear (LE) and behind the right ear (RE) are used to correct for the movement of the head. Unused coils are glued to the participant’s cheeks or their forehead.
A receiver coil captures pitch and yaw but does not capture roll along its axis as this does not change the induced voltage in the receiver coil. Hence, in order to capture the angle of elevation of the tongue, the axis of the coil should be placed parallel to the midsagittal plane.

Coils are glued on the speech articulators of the speaker in the desired orientation. The participant is asked to sit under the transmitter configuration of the EMA machine and the receiver coils are connected to the EMA machine by leads. In order to correct the speech articulator measurements for reference frames, we record a trial reading. This trial reading is taken with the speaker’s head at rest, the transverse plane parallel to the floor (accomplished using a 3D bubble level), and with no movements from any of the articulators. This position is considered a baseline position and the position of the head
in subsequent recordings is corrected to this reference position so the other coil positions are effectively decoupled from head motion.

2.4.4 Performance

The manufactures of the AG501 machine, Carstens Medizinelektronik GmbH, claim a spatial accuracy of 0.3mm.

Lau [27] analyzed the performance of the AG501 system by measuring the accuracy of measurements made by the AG501 machine along the x-y plane though the origin (z=0 as defined by the manufacturer). For static coil positions, the machine showed no error when the coil was placed in the center of the field, and a maximum error of 0.43mm when the coil was placed 100mm back and 100mm to the left of center. The AG501 system also shows an average dynamic RMS error of 0.058mm across different position of a receiver coil. Studies by [27] and [50] show that the AG501 has a consistent error trend across different spatial locations.

2.5 Data Processing

The AG501 EMA system records acoustics with a sampling rate of 48 kHz and kinematics with a sampling rate of 250 Hz. The acoustic and kinematic data are processed in three steps. The data is first pre-processed by down sampling the acoustic data, and the kinematic data is band-pass filtered and adjusted for the head-based reference frame. Silent regions in this pre-processed data are found and removed from the dataset. Finally, the data is windowed in 30 ms windows and processed once more where the kinematic data in a window is averaged, and the vocal tract filter function for the acoustic data in a window is determined. We discuss these steps in detail next.
2.5.1 Pre-Processing

Acoustic data captured at 48 kHz by the EMA machine is down-sampled to 16 kHz, as the 0-8 kHz frequency range captures the range of most relevant speech sounds \[13\]. This down sampling is done using algorithms 8.2 and 8.3 in \[38\].

The AG501 EMA machine samples movement data at 250Hz. For each kinematic sensor we have three values representing the location of the kinematic sensor coil in space, and two angular values representing pitch and yaw of the coil about the axis of its loop.

We shift the reference frame of movement data for some articulators to account for contributions of other articulators. For example, the lower lip rests on the jaw. Therefore any movement of the jaw is added to the movement of the sensor coil attached to the lower lip. Similarly, the positions of all sensors mounted on the face are affected by the movement of the person’s head, and hence need to be decoupled from the latter. The Head-Decoupling is done by moving the specific articulator coordinates from the AG501’s reference frame to the Head’s reference frame, and similarly Jaw-Decoupling is done by moving the specific articulator’s coordinates from the AG501’s reference frame to the Jaw’s reference frame.

Several different methods have been proposed to perform the decoupling the movements of the head and jaw. The position estimation method (3DPE), the estimated rotation method (ERM) \[58\] and linear subtraction are some of the methods that have been used in the past. In the current study, the JOANA method proposed in \[19\] was used. This method has been shown to produce smaller errors in mapping reference frames compared to any of the other methods.

The movements of the jaw and upper lip are decoupled from the movement of the head. The movements of the lower lip and tongue are decoupled from the movement of the Jaw.

After mapping the movement data to the appropriate reference frame, the kinematic positions are low pass filtered to 20Hz. Speech related articulator movements are band-limited to 15Hz \[33\] using a five pole Butterworth filter so no significant information is lost. Figure \[2.4\] shows that 99\% of the signal energy for the movement of the tongue tip is contained below 5Hz.
2.5.2 Silence Removal

We assume that data collected during acoustic silence is unrelated to the direct relationship between acoustics and kinematics. With EMA, the positions of the articulators are recorded even when the subject is not speaking. These acoustically silent portions of recordings can correspond to the speaker pausing between words, swallowing or gaps that occur naturally in unrestricted speech. These silences may not necessarily be meaningless, however since these movements do not have an acoustic representation, we do not use them in defining the relationship between acoustics and kinematics. Unlike previous experiments like [21] where the speech stimulus is restricted, we work with natural speech and with phonetically diverse speech segments. Hence there are many instances where the dataset is acoustically silent but the speaker may still be moving his/her speech articulators. These segments in the dataset are identified and removed.

To identify silent regions we compare the short term energy in a speech segment and apply a threshold to classify whether that segment is silent or not. No specific criterion is used and a threshold is set by trial and error. Short term energy is found for 30ms
segments of speech using the formula defined in equation 2.2. We make the assumption that the majority of the samples contain speech segments and any speech segment with energy levels less than 25% of the average energy in all the segments corresponds to a silent segment. In order to aid our assumption that most samples contain speech, we first get rid of long silences in a sentence. Silent segments are therefore identified and removed in two stages. The first stage is defined at the sentence level and the second stage is defined at the level of the 30ms segments.

The envelop of the acoustics associated with a sentence was found by low-passing the absolute values of the acoustics signal at 15Hz. For this a 5th order Butterworth filter was used. Any sub segment within a sentence which is longer than 0.2 seconds and within which every sample has an absolute value less than 25% of the mean value of the acoustic envelop is identified. This sub-segment is assumed to correspond to a long pause in speech and will be referred to as a silent segment. The original sentence is broken down into smaller parts without long pauses by extracting the acoustics surrounding a silent segment. The acoustics without long pauses are then used for analysis. Figure 2.5 shows a sentence and the parts without long silences which are extracted from it.

![Figure 2.5: Time domain representation of a sentence. The red boxes are speech segments without long pauses. See text for more details.](image)

The acoustics and kinematics within extracted parts are windowed into 30ms segments. The average energy in a 30ms frame is calculated by taking the sum of squares of values of acoustic in each frame, and averaging this across the frames. If any 30ms segment contains acoustic energy less than 25% of this average energy, then that segment is deleted from the dataset (see Figure 2.6).
\[
ShortTermEnergy = \sum_{n=1}^{480} s(n)^2 \tag{2.2}
\]

Figure 2.6: Figure shows the process by which 30ms silent regions are removed. The figure above shows three 30ms windows picked randomly from the dataset. The average energy of a 30ms window from all windows in the dataset is 2.03 units. The left and middle windows have more energy than 25% of 2.03 and are therefore classified as regions with speech. The right window has less energy than the 25% threshold and is classified as a silent region and discarded from analysis.

2.5.3 Kinematic Processing

The silence removal procedure windows the kinematic data into 30ms windows. Values for each movement dimension i.e. front-back (x), left-right (y), up-down (z), pitch and yaw, for each receiver coil is averaged within a window. This reduces the amount of data to one value for each dimension for each receiver coil for every 30ms window, effectively lowering the sampling rate from 250Hz to \(1/0.03 = 33.33\)Hz. This is similar to the 30Hz sampling rate used in other studies (e.g. [21]).

Table 2.1 shows the maximum amplitudes of speech articulator movements for all three speakers.
Table 2.1: Maximum amplitude of movement of speech articulators (mm). The contribution of the jaw is removed from the lower lip and the tongue amplitudes.

<table>
<thead>
<tr>
<th></th>
<th>Speaker A</th>
<th>Speaker B</th>
<th>Speaker C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tongue Tip X</td>
<td>31.6</td>
<td>22.5</td>
<td>25.1</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>25.5</td>
<td>23.7</td>
<td>21.1</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>30.1</td>
<td>17.6</td>
<td>27.4</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>34.6</td>
<td>29.0</td>
<td>25.6</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>32.7</td>
<td>22.5</td>
<td>25.4</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>30.4</td>
<td>29.2</td>
<td>21.2</td>
</tr>
<tr>
<td>Jaw X</td>
<td>8.7</td>
<td>10.8</td>
<td>6.5</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>14.4</td>
<td>28.1</td>
<td>20.0</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>6.2</td>
<td>6.4</td>
<td>6.9</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>11.5</td>
<td>10.6</td>
<td>9.3</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>6.3</td>
<td>10.4</td>
<td>15.0</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>13.4</td>
<td>17.3</td>
<td>14.5</td>
</tr>
</tbody>
</table>

2.5.4 Acoustic Processing

The silence removal procedure windows the acoustic data into 30 ms windows. In order to link the acoustics to the kinematics, we transform the acoustics into a form more closely related to the kinematic values by calculating its vocal tract transfer function. In order to prevent high frequency artifacts when finding the vocal tract filter function, we first multiply the acoustics data in a window with a hamming window.

Based on the source filter theory of speech which is detailed later in the section 2.5.4, the vocal tract transfer function is directly related to the positions of the speech articulators. The two methods that are widely used to derive the vocal tract transfer function are Linear Predictive Coding (LPC), which model the vocal tract as an all pole filter defined by pairs of coefficients, and cepstral liftering, which finds the vocal tract function by removing the repeated elements in the speech spectrum which correspond to the fundamental frequency and higher harmonics. For the current study, we use cepstral liftering as this is the method preferred in some previous studies [21]. The procedure for cepstral liftering is explained later in this section.
Source-Filter Model of Speech Production

The source-filter model defines speech as a combination of the sound source from the vocal chords which is modulated by a linear acoustic filter of the vocal tract. The vocal chords are responsible for the periodic structure of speech and associated energy distribution in the speech signals. This periodic speech signal is associated with the fundamental frequency and the higher harmonics. The vocal tract filtering is associated with the vocal tract transfer function, which defines the resonance properties of the vocal tract. The resonance of the vocal tract is thus a direct consequence of the shape of the vocal tract. The shape of the vocal tract in turn is largely influenced by the actions of the articulators. Figure 2.7 shows the locations of the glottal source and the vocal tract components which determine the vocal tract transfer function. The figure 2.8 shows the acoustic outputs from the glottal source and the transfer function for a particular configuration of the vocal tract.

Figure 2.7: Figure showing the source and filter components compromising the source filter model
Chapter 2. Methods for Data Collection and Analysis

Figure 2.8: The left panel shows the acoustics (in the frequency domain) generated by the glottal source. The central panel shows the vocal tract filter which is defined by the shape of the vocal tract. The panel on the right shows the final acoustic which is a consequence of a multiplication of the frequencies from the glottal source with the vocal tract transfer function.

The source and the filter are modeled as independent systems, i.e. the vocal folds action does not determine the resonant characteristics, and conversely the shape of the vocal tract (and the associated articulator positions) is not assumed to affect the source signal. This allows us to study the vocal tract shape by measuring the resonance properties of the sound, and the glottal source properties by measuring the fundamental frequency and harmonic structure of the speech signal.

When articulator positions are stationary (or quasi-stationary) during the production of an acoustic sound, then one can determine its corresponding vocal transfer function. In this study, we analyze the acoustics and kinematic in 30ms windows. The change in kinematic values for this duration is relatively small as the articulators move with a bandwidth under 15Hz (66ms) \cite{33}. We can therefore consider the kinematic values to be quasi-stationary for our analysis window. The vocal tract transfer function can be obtained through cepstral liftering.

Cepstral Liftering

In this study we use the first 20 cepstral coefficients defined in the 0-16kHz range. The cepstrum is defined as the inverse Fourier transform of the log of the Fourier transform of the signal.

\[
Cepstrum = ifft(\log(fft(Speech)))
\]  

(2.3)
The reason for taking the cepstrum is that the pitch can be isolated as a single point in the cepstral domain. In the Fourier domain, the speech signal is seen as a multiplication of the acoustic transfer function and the Fourier transform of the pitch. The Fourier transform of the pitch is a train of frequency spikes starting from the fundamental frequency (F0) and subsequent spikes separated by the value of the fundamental frequency (making up the harmonics). By taking the log of the Fourier transform, the multiplication of the transfer function and pitch is transformed into a sum. This allows the F0 to be isolated as a single point when we take the inverse Fourier transform of this value.

\[
S(\omega) = G(\omega) \ast H(\omega)
\]

\[
\log(S(\omega)) = \log(G(\omega)) + \log(H(\omega))
\]

(2.4)

Here \(G(\omega)\) is the Fourier transform of the glottal source and \(H(\omega)\) is the vocal tract transfer function.

When taking the inverse Fourier transform of the log Fourier values, the repeated peaks forming the fundamental frequency (F0) are isolated to a single point typically located at a higher cepstral value. The lower cepstral values represent the envelop of the speech spectrum reflecting the vocal tract transfer function.

A typical value for the fundamental frequency for male speakers is approximately 100 Hz and for female speakers is approximately 200 Hz [57]. A pitch of 85 Hz has a time period of 1/120 = 0.008 seconds and a pitch of 210 Hz has a time period of 1/210 = 0.004 seconds. At a sampling rate of 16 kHz (a sampling period of 0.0000625 seconds), these pitch value are mapped to 0.008/0.0000625 = 128 and 0.004/0.0000625 = 64 quefrency (unit of cepstral coefficients) values respectively. The F0 is thus unlikely to be found in the first 20 quefrency values we take for our analysis. All the higher quefrency values are set to 0. This process is called cepstral liftering. We then convert the remaining cepstrum values to the corresponding speech spectrum for our analysis. This is done by taking Fourier transform of the inverse cepstrum of the quefrency values. The inverse cepstrum transforms the cepstral value back into the time domain (see 2.9). The inverse cepstrum is defined as
\[ \text{InverseCesptrum} = \text{ifft}(\exp(\text{fft(cepstrum)})) \]  \hspace{1cm} (2.5)

We use the magnitude of the frequency response of these inverse cepstral values for our analysis.

![Frequency Characteristics](image1.png)

![Frequency Response](image2.png)

Figure 2.9: Figure on the left shows the frequency characteristics of a 30ms segment of sound. The figure on the right shows the frequency response of the same segment after lifting (i.e. after removing contribution from the glottal source).

### 2.5.5 Preparation of Training and Test Dataset

We split each dataset (i.e., data from one speaker) into a training set and a test set. The training and test dataset are used to evaluate the performance of the mapping models described in chapters 3 and 4. These mapping models are trained on the training set, and acoustic to kinematic predictions are found and evaluated on the test set.

We use a 10-fold cross validation technique to evaluate the predictive models. For this method, the dataset is partitioned into 10 sub samples containing equal number of sentences. As explained in section 2.3, we record each of the 200 sentences in the stimuli set twice for each speaker. Each sub sample contains 20 of the 200 sentences, as well as the repetition of these sentences from the repeated recording. This results in 40 sentences in each sub-sample. Of the 10 sub-samples, one sub-sample is retained as the test set, and the remaining 9 sub-samples are used as the training set. This cross-validation process is repeated 10 times, which each of the 10 sub-samples used exactly once as the test set. This results in 10 test sets (each with 20 sentences repeated twice) and 10 corresponding training sets (each with 180 sentences repeated twice). A mapping model constructed
using a particular training set is evaluated using its corresponding test set.

The training data contains about 60000 acoustic frames. These acoustic frames represent vocal tract transfer functions (VTFF) in 30ms frames of data. The exact amount of data in a training set differs between datasets due to the different rates of speech by participants. The amount of data in a training set also differs between the 10 training sets constructed from a dataset due to different lengths of sentences.

2.6 Evaluation of Mapping Techniques

This work presents two techniques explained in Chapter 3 and Chapter 4 to map acoustics to articulatory positions. These techniques are evaluated by assessing the accuracy of acoustic to kinematic predictions. The details of finding these predictions are different for both the techniques and are explained separately in Chapters 3 and 4. However, the metrics to evaluate the accuracy of predictions using these techniques are the same, and are discussed in this section.

We use two metrics to evaluate prediction performance, namely, how well the predictions are correlated to the real values, and the root mean squared difference (RMSD) between the predicted and the real values. Both the correlation and the RMS error are measures of closeness between two variables, but where correlation gives us a measure of the closeness of the trends in movement direction in real and predicted data, RMSD gives a measure of how different the magnitude of these values are from one another. Together they help us better understand the quality of the predictions.

2.6.1 Correlations

Correlation is a measure of dependence between variables. By measuring the correlation between the real kinematic positions and the predicted kinematic positions, we can determine if the predicted kinematics change in the same manner as the real kinematics.

There are a few different techniques to measure correlations. We use the MATLAB
function `corr` to measure correlations, which gives us the Pearson’s Product Moment correlation value. Pearson’s correlation values indicate the linear relationship between variables and the values range between -1 and 1 with 1 meaning maximum positive linear correlation, -1 meaning total negative linear correlation and 0 meaning no correlation. Pearson’s correlation is calculated as

$$\rho(X, Y) = \frac{cov(X, Y)}{(\sigma_X \sigma_Y)}$$

(2.6)

It is important to note that a Pearson’s correlation of 1 does not necessarily mean perfect predictions. Since Pearson’s correlation is a measure of linear dependence, any set of random variables $X$ and $Y$ where $Y = X \times C$ where C is a constant will give a Pearson’s correlation of 1, although $X$ and $Y$ are not the same. A more detailed analysis of the interpretation and shortcomings of correlation is provided in section C of the appendix.

### 2.6.2 Root-Mean-Square Deviation

The root-mean-square deviation (also known as the root-mean-square error) is a measure of the magnitude of the difference between two variables. RMSD is calculated as

$$RMSD = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (x_t - y_t)^2 / n}$$

(2.7)

A higher RMSD value indicates poorer predictions and a RMSE value of 0 indicates perfect prediction.

Unlike correlation which indicated how closely two variables vary with each other, RMSD indicates the magnitude of their difference. Hence RMSD successfully reflects the difference in the absolute values of variables in situations where the correlation can only show a linear relationship in the way these variables change over time.

RMSD does not take into account the degree of change in the variable values, and hence it is often meaningless to compare the RMSD of variables that differ in scale. In Figure
2.10 Variable A has a maximum amplitude of 10 units and is predicted by variable A1. Variable B has a maximum amplitude of 1 unit and is predicted by variable B1. Although the RMSD of prediction of A (1.86) is higher than that of B (0.88), the graph suggests that variable A is probably better predicted.

This problem can be overcome by normalizing the RMSD by dividing the RMSD by the amplitude of the original kinematic to produce the normalized root-mean-square deviation (NRMSD). The NRMSD of the variables in 2.10 is 0.1869 for variable A and 0.8904 for variable B. These NRMSD values more accurately reflect the true prediction performance for these variables.

A more detailed analysis of the interpretation and shortcomings of correlation is provided in section C of the appendix.
Chapter 3

Acoustic to Kinematic Mapping - Part A : Traditional Approach With Vector Quantization

3.1 Description

It was originally shown by Hogden et al [21] that a vector quantization codebook method could be used to recover articulator positions from acoustics for a restricted dataset containing only vowels, vowel-to-vowel transitions, /g/ closures and transitions into and out of /g/ closures. In this chapter, we show that the codebook vector quantization method proposed by Hogden [21] can be used to map acoustic to articulator positions for a larger dataset containing naturally spoken phonetically diverse sentences. The codebook presented here uses a similar vector quantization (VQ) clustering method to find clusters in the acoustic domain. The centers of these acoustic clusters were mapped to the average value of the articulatory positions used to produce those acoustics, which resulted in a one-to-one acoustic to kinematic codebook. We show that the performance of our codebook trained on naturally spoken phonetically diverse speech is comparable to the performance of the codebook on restricted speech data presented by Hogden et al. [21].
3.2 Methods

The procedure for constructing the codebook and finding predictions from the codebook are similar to the methods presented by Hogden et al. [21]. Acoustic to kinematic codebooks were constructed on the training set using the Vector Quantization (VQ) method. The Vector Quantization method finds clusters in the acoustics values in training dataset, explained in detail in section 3.2.1. Kinematics predictions were based on acoustics in the test set and these predictions were compared to real kinematic values. Hogden’s codebook [21] used 128, 256 and 512 codes. We evaluated predictions from codebooks using the same number of codes.

3.2.1 Construction of the Codebook

The codebook maps acoustic reference vectors to kinematic reference vectors. A vector quantization method was used to perform clustering on the acoustic data (i.e. the vocal tract filter functions, from here on referred to as VTFFs) in the training dataset. The centers of the acoustic clusters were the acoustic reference vectors. The average values of all kinematic values associated with acoustic clusters were the kinematic reference vectors. This leads a one-to-one acoustic to kinematic codebook with as many codes as there are clusters.

As explained in section 2.5.5, 10 training sets were constructed on each dataset for 10-fold cross validation. These training sets contained about 60000 datapoints, with each datapoint containing an acoustic vector (represented as VTFFs in 240 points for frequencies from 0-8000Hz in a 30ms acoustic window) and a kinematic vector (containing position information of speech articulators). Three codebooks (with 512, 256 and 128 codes) were constructed on each training dataset. This process resulted in 30 codebooks constructed per participant, and a total of 90 codebooks for all three participants.

The k-means function in Matlab was used to perform the vector quantization [1]. This k-means function uses Lloyd’s algorithm [30] to minimize the sum of the squared Euclidean distance between the acoustic values and their corresponding centers.

The clustering problem is NP-hard [25], i.e. the solution to clustering cannot be found in
polynomial time. This means that the number of steps to perform clustering scales faster than any polynomial in n (where n is the number of values being clustered). Hence finding optimal clusters (clusters where all values are closer to the center of that cluster than the center of any other cluster) is computationally expensive. Since this makes finding a globally optimal solution difficult; we settled for a suboptimal clustering solution. A suboptimal solution is reached when the cluster centers change less than a pre-defined threshold value over subsequent iterations of Lloyd’s algorithm. The time taken by Lloyd’s algorithm and the quality of the solution are affected by the initialization of the cluster centers. We used the k-means++ algorithm [3] to initialize the values of the centroids. The k-means++ algorithm initializes the cluster centers based on the statistical distribution of the data, and has been shown to improve the quality and time taken to reach an acceptable suboptimal clustering solution [3].

3.2.2 Mapping Acoustics to Kinematics

Kinematic prediction for an acoustic value was determined by the kinematic reference vector corresponding to the acoustic reference vector closest to the given acoustic value. Predictions were performed on acoustics from the test dataset, and the resultant predicted kinematics were compared to the real kinematics.

Figure 3.1 shows a schematic for the construction of the codebook, and how it was used to determine the acoustic to kinematic predictions.
Chapter 3. Acoustics To Kinematics - Part A

Figure 3.1: Kinematic prediction schematics

The acoustic reference vector selected was the one closest (in terms of minimal Euclidean distance) to the given acoustic. Figure 3.2 shows the process of associating an acoustic value to an acoustic and kinematic reference value using an abstract codebook example.

Figure 3.2: A sample codebook with six codes. Acoustics are in one dimension and kinematics are represented by abstract $K_n$ value. Acoustic value 6.6 is mapped to code with acoustic center 5.9 and thus the predicted corresponding kinematic value is K4.
3.2.3 Model Evaluation

The performance of the codebook was evaluated on the test set by comparing the predicted kinematic positions to known positions, by using correlation, root mean squared distance (RMSD), and normalized root mean squared distance (NRMSD) measures. A detailed description of the RMSD, NRMSD and correlation methods is provided in section 2.6.

The RMSD, NRMSD and correlation values were compared to their corresponding values from a null dataset found using a random codebook. These random codebooks were created by constructing a codebook on a dataset where acoustic and kinematic values are unrelated. Section 2.5.5 explained the construction of the training dataset which contains roughly 60000 data points, with each data point containing an acoustic vector (VTFF) and a kinematic position vector. In order to construct the null dataset, we modified this training dataset by associating the acoustics in one datapoint with the kinematic vector in another datapoint. As a result, the acoustics and kinematics in the null dataset are misaligned. We constructed 10 such random codebooks, and the predictions from these codebooks form the null predictions.

3.2.4 Similarity of Acoustics Within a Cluster

The codebook for this study was constructed by performing vector quantization on the speech acoustics. Hence the acoustics within a code should be roughly similar to one another, making the average difference between acoustics within a code less than the average difference for a randomly selected set of acoustic samples from the dataset. For a 512 code codebook, we can analyze the acoustic properties of each of its 512 clusters. Figure 3.3 shows the average acoustic distance in 512 codebook acoustic cluster compared to the average acoustic distance in 512 non-related datasets sets. A non-related dataset was made up of randomly chosen acoustic datapoints from the training datasets, with the same number of datapoints as an average cluster.

Figure 3.4 shows the p-values for an unpaired t-test with 1022 degrees of freedom, comparing the average acoustic distance in a cluster to the distribution of acoustic distances in the 512 non-related sets. These results are found for an unpaired t-test with 1022
degrees of freedom. We see from this figure that for frequencies less than 3000 Hz it is very unlikely that a non-related dataset could have acoustic distances as low as those found in a real acoustic cluster.

Figure 3.3: The average acoustic distance within 512 codes of the codebook compared to t non-related datasets is 0.0018. The total sumhe average acoustic distances for 512 non-related sets. Non-related sets are constructed by choosing acoustic values from the training set at random. Shown is a codebook with 512 codes for speaker A’s data. The total sum of average acoustic distances for all frequencies for of average acoustic distance for all frequencies within codebooks clusters was 0.00007.
We see from figures 3.3 and 3.4 that the average acoustic distances within a codebook cluster was significantly smaller than the average acoustic distances in the training dataset for frequencies under 3000 Hz (which roughly corresponds to the location of the third formant). The difference between the acoustic distances within a cluster and the acoustic distances in the non-related set was not significant for frequencies over 3000 Hz. This difference across frequencies is likely related to the fact that the k-means algorithm we used for clustering is more sensitive for frequencies with high energies, which are typically found below 3000 Hz (first two formants).

3.2.5 Similarity of Kinematics Within a Cluster

Since the codebook in this study was made by clustering acoustics, there was no explicit restriction on the variability of kinematic values corresponding to the acoustics in a cluster. However, results by Hogden et al. [21] showed that for his study the average kinematic value in a cluster was a good representation of all kinematics in that cluster. From this we can infer that the kinematic values in a cluster may also be similar for our
study; thus the average distance between kinematic values in a cluster should be lower than the average distance calculated for kinematic values not belonging to a cluster. Table 3.1 shows the average distance between kinematics in 512 clusters (of a 512 code codebook) is less than then the average distance calculated for 512 non-related datasets. A non-related dataset was made up of randomly chosen kinematic datapoints from the training datasets, and has the same size as an average clusters size.

<table>
<thead>
<tr>
<th></th>
<th>Average Distance Between Kinematics in Non-related Dataset</th>
<th>Average Distance Between Kinematics Within a Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tongue Tip X</td>
<td>2.58</td>
<td>3.55</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>2.1</td>
<td>2.65</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>2.32</td>
<td>3.39</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>2.92</td>
<td>4.13</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>2.45</td>
<td>3.6</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>2.57</td>
<td>3.77</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.63</td>
<td>0.89</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>1.02</td>
<td>1.48</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.39</td>
<td>0.45</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>0.81</td>
<td>0.89</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.50</td>
<td>0.64</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>0.87</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 3.1: Average distance between kinematics mapped to a code vs the average distance between kinematics in non-related datasets. Data are extracted from speaker A, and clusters are based on a 512 code codebook.

We see from table 3.1 that the kinematic distance in the cluster was significantly smaller than the kinematic distance in the non-related dataset. An unpaired t-test with 1022 degrees of freedom reveals that this difference in kinematic distances between the two datasets was significant, with a p-value lower than 0.05 (t-value greater than 1.65) for all speech articulator positions shown in the table. The only exception to this is the upper lip in the vertical (Z) direction which shows a p-value of 0.07 (t-value 1.48). An explanation for this is that the upper lip shows little movement in the vertical direction, and this movement may have minimal effect on the produced acoustic signals.
3.3 Results

This section shows the results from predictions using codebooks with 512, 256 and 128 codes. Subsections 3.3.1 shows the RMSD and NRMSD values; and subsection 3.3.2 shows the correlation values for predictions from the codebooks for all three speakers. By comparing the predictions from the codebook with a null distribution, it can be shown that the performance of the aligned codebook was significantly better.

A comparison for the performance between these different numbers of codes is shown in section 3.4.1.

3.3.1 RMS Errors

Tables 3.2, 3.3 and 3.4 show the root-mean-square deviations (RMSD) and normalized root-mean-square deviations (NRMSD) of the predictions from the codebook for 512, 256 and 128 codes respectively for three speakers. Since we used a 10-fold cross validation technique, these tables show the mean RMSD and NRMSD values found across the 10 training-test set pairs.

<table>
<thead>
<tr>
<th></th>
<th>RMSD</th>
<th></th>
<th></th>
<th>NRMSD</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Speaker A</td>
<td>Speaker B</td>
<td>Speaker C</td>
<td>Speaker A</td>
<td>Speaker B</td>
<td>Speaker C</td>
</tr>
<tr>
<td>Tongue Tip X</td>
<td>3.16</td>
<td>2.70</td>
<td>2.26</td>
<td>0.10</td>
<td>0.12</td>
<td>0.09</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>2.55</td>
<td>2.14</td>
<td>2.33</td>
<td>0.10</td>
<td>0.09</td>
<td>0.11</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>2.71</td>
<td>1.94</td>
<td>2.73</td>
<td>0.09</td>
<td>0.11</td>
<td>0.10</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>3.81</td>
<td>3.19</td>
<td>3.0</td>
<td>0.11</td>
<td>0.11</td>
<td>0.12</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>2.88</td>
<td>2.70</td>
<td>2.29</td>
<td>0.09</td>
<td>0.12</td>
<td>0.09</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>3.04</td>
<td>2.92</td>
<td>2.76</td>
<td>0.10</td>
<td>0.10</td>
<td>0.13</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.70</td>
<td>0.97</td>
<td>0.46</td>
<td>0.08</td>
<td>0.09</td>
<td>0.07</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>1.15</td>
<td>2.25</td>
<td>1.4</td>
<td>0.08</td>
<td>0.08</td>
<td>0.07</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.48</td>
<td>0.51</td>
<td>0.69</td>
<td>0.08</td>
<td>0.08</td>
<td>0.10</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>1.03</td>
<td>1.17</td>
<td>0.93</td>
<td>0.09</td>
<td>0.11</td>
<td>0.10</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.63</td>
<td>1.15</td>
<td>1.80</td>
<td>0.10</td>
<td>0.11</td>
<td>0.12</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>0.94</td>
<td>2.08</td>
<td>1.60</td>
<td>0.07</td>
<td>0.12</td>
<td>0.11</td>
</tr>
</tbody>
</table>

Table 3.2: RMSD and NRMSD in mm using a codebook constructed with 512 codes for jaw, tongue, upper lip and lower lip data
Table 3.3: RMSD and NRMSD in mm using a codebook constructed with 256 codes for jaw, tongue, upper lip and lower lip data

<table>
<thead>
<tr>
<th></th>
<th>Speaker A</th>
<th>Speaker B</th>
<th>Speaker C</th>
<th>Speaker A</th>
<th>Speaker B</th>
<th>Speaker C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tongue Tip X</td>
<td>3.19</td>
<td>2.72</td>
<td>2.29</td>
<td>0.10</td>
<td>0.12</td>
<td>0.11</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>2.57</td>
<td>2.16</td>
<td>2.36</td>
<td>0.10</td>
<td>0.09</td>
<td>0.11</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>2.73</td>
<td>1.96</td>
<td>2.77</td>
<td>0.09</td>
<td>0.11</td>
<td>0.10</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>3.85</td>
<td>3.24</td>
<td>3.03</td>
<td>0.11</td>
<td>0.11</td>
<td>0.12</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>2.90</td>
<td>2.72</td>
<td>2.33</td>
<td>0.09</td>
<td>0.12</td>
<td>0.09</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>3.06</td>
<td>2.99</td>
<td>2.79</td>
<td>0.10</td>
<td>0.10</td>
<td>0.13</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.71</td>
<td>0.98</td>
<td>0.47</td>
<td>0.08</td>
<td>0.09</td>
<td>0.07</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>1.16</td>
<td>2.28</td>
<td>1.44</td>
<td>0.08</td>
<td>0.09</td>
<td>0.07</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.50</td>
<td>0.51</td>
<td>0.69</td>
<td>0.08</td>
<td>0.08</td>
<td>0.10</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>1.03</td>
<td>1.18</td>
<td>0.94</td>
<td>0.09</td>
<td>0.11</td>
<td>0.10</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.63</td>
<td>1.16</td>
<td>1.82</td>
<td>0.10</td>
<td>0.11</td>
<td>0.12</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>0.95</td>
<td>2.11</td>
<td>1.62</td>
<td>0.07</td>
<td>0.12</td>
<td>0.11</td>
</tr>
</tbody>
</table>

Table 3.4: RMSD and NRMSD using a codebook constructed with 128 codes for jaw, tongue, upper lip and lower lip data

<table>
<thead>
<tr>
<th></th>
<th>Speaker A</th>
<th>Speaker B</th>
<th>Speaker C</th>
<th>Speaker A</th>
<th>Speaker B</th>
<th>Speaker C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tongue Tip X</td>
<td>3.21</td>
<td>2.74</td>
<td>2.31</td>
<td>0.10</td>
<td>0.12</td>
<td>0.09</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>2.60</td>
<td>2.17</td>
<td>2.38</td>
<td>0.10</td>
<td>0.09</td>
<td>0.11</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>2.75</td>
<td>1.97</td>
<td>2.80</td>
<td>0.09</td>
<td>0.11</td>
<td>0.10</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>3.87</td>
<td>3.29</td>
<td>3.06</td>
<td>0.11</td>
<td>0.11</td>
<td>0.12</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>2.91</td>
<td>2.76</td>
<td>2.35</td>
<td>0.09</td>
<td>0.12</td>
<td>0.10</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>3.09</td>
<td>3.04</td>
<td>2.82</td>
<td>0.10</td>
<td>0.10</td>
<td>0.13</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.72</td>
<td>1.00</td>
<td>0.47</td>
<td>0.09</td>
<td>0.09</td>
<td>0.07</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>1.18</td>
<td>2.34</td>
<td>1.44</td>
<td>0.08</td>
<td>0.09</td>
<td>0.07</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.50</td>
<td>0.52</td>
<td>0.70</td>
<td>0.08</td>
<td>0.08</td>
<td>0.10</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>1.05</td>
<td>1.19</td>
<td>0.94</td>
<td>0.09</td>
<td>0.11</td>
<td>0.10</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.64</td>
<td>1.16</td>
<td>1.83</td>
<td>0.10</td>
<td>0.11</td>
<td>0.12</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>0.96</td>
<td>2.14</td>
<td>1.62</td>
<td>0.11</td>
<td>0.12</td>
<td>0.11</td>
</tr>
</tbody>
</table>

These results show that all kinematics predicted from the codebook showed about a 10% RMS error relative to their respective amplitudes (NRMSD). Results from participant A (male) generally showed a higher RMS deviation compared to results from participants B and C (females). However, since the normalized RMSD values of all the speakers are similar (with these values for speaker A often lower than those for speakers B and C), higher RMS values for participant A were likely due to higher amplitudes of motion of speaker A’s speech articulators. Table 2.1 in section 2.5.3 shows the amplitude of
movement of speech articulators for each speaker. Of all the speech articulators, the positions of the jaw seem to be predicted most accurately.

We compared the RMSD values using predictions from the codebook to the RMSD values from predictions using randomly selected codes from the codebook (i.e., the null distribution). Table 3.5 shows the RMSD values using the codebook and using the null distribution for a 512 code codebook on speaker A’s data. Since the 10-fold cross validation procedure gave us 10 sets of predictions, data presented here show the mean and standard deviation of the predictions from codebook mapping and from the null distribution. An unpaired t-test with 18 degrees of freedom (as there are predictions from 10 aligned codebooks, and 10 random codebooks) reveals that the difference between the predictions from the VQ codebook and the null codebook is significant with a p-value lower than 0.05 (t-value greater than 1.73) for all speech articulators.

We compared the RMSD values using predictions from the codebook to the RMSD values from predictions using the random codebooks (found by random association of acoustics and kinematics as described in 3.2.3). Predictions from the random codebook form the null distribution. The table 3.5 shows the RMSD values using the codebook and using the null distribution for a 512 code codebook on speaker A’s data. Since the 10 fold cross validation procedure gave us 10 sets of predictions, data presented here show the mean and variance of the predictions from codebook mapping and from the null distribution. A paired t-test with 9 degrees of freedom reveals that the difference between the predictions from the VQ codebook and the null codebook is significant with a p-value lower than 0.05 (t-value greater than 1.73) for all speech articulators. In this t-test, the predictions for each articulators are compared between the VQ codebook and the random codebook for each of the 10 training/test datasets, thus bringing the degrees of freedom to 9.
### Table 3.5: Mean and standard deviation for NRMSD for predictions using 512 code codebook for speaker A, compared to the mean and standard deviation of NRMSD of predictions from a null distribution

<table>
<thead>
<tr>
<th></th>
<th>VQ Codebook Mapping</th>
<th>Null Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>SD</td>
</tr>
<tr>
<td>Tongue Tip X</td>
<td>0.10</td>
<td>0.005</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>0.10</td>
<td>0.001</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>0.09</td>
<td>0.002</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>0.11</td>
<td>0.003</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>0.09</td>
<td>0.002</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>0.10</td>
<td>0.005</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.09</td>
<td>0.006</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>0.08</td>
<td>0.007</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.08</td>
<td>0.002</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>0.09</td>
<td>0.001</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.10</td>
<td>0.007</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>0.11</td>
<td>0.004</td>
</tr>
</tbody>
</table>

3.3.2 Correlation

Tables 3.6, 3.7 and 3.8 show the correlations between the predictions from the codebook for 512, 256 and 128 codes respectively for the three speakers. Since we used an 10-fold cross validation technique, the data shown in the table indicate the mean correlation from all the trials.
Table 3.6: Correlation between real kinematics values and predicted values using a code-book constructed with 512 codes for jaw, tongue, upper lip and lower lip data

<table>
<thead>
<tr>
<th></th>
<th>Speaker A</th>
<th>Speaker B</th>
<th>Speaker C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tongue Tip X</td>
<td>0.66</td>
<td>0.42</td>
<td>0.50</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>0.61</td>
<td>0.50</td>
<td>0.38</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>0.72</td>
<td>0.36</td>
<td>0.55</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>0.65</td>
<td>0.51</td>
<td>0.41</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>0.72</td>
<td>0.40</td>
<td>0.57</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>0.68</td>
<td>0.62</td>
<td>0.40</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.74</td>
<td>0.71</td>
<td>0.71</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>0.75</td>
<td>0.74</td>
<td>0.73</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.59</td>
<td>0.47</td>
<td>0.43</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>0.55</td>
<td>0.40</td>
<td>0.26</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.64</td>
<td>0.29</td>
<td>0.41</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>0.58</td>
<td>0.49</td>
<td>0.33</td>
</tr>
</tbody>
</table>

Table 3.7: Correlation between real kinematics values and predicted values using a code-book constructed with 256 codes for jaw, tongue, upper lip and lower lip data

<table>
<thead>
<tr>
<th></th>
<th>Speaker A</th>
<th>Speaker B</th>
<th>Speaker C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tongue Tip X</td>
<td>0.64</td>
<td>0.40</td>
<td>0.48</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>0.60</td>
<td>0.48</td>
<td>0.35</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>0.70</td>
<td>0.33</td>
<td>0.53</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>0.63</td>
<td>0.49</td>
<td>0.39</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>0.70</td>
<td>0.39</td>
<td>0.55</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>0.67</td>
<td>0.60</td>
<td>0.38</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.72</td>
<td>0.70</td>
<td>0.69</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>0.74</td>
<td>0.73</td>
<td>0.71</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.58</td>
<td>0.45</td>
<td>0.41</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>0.54</td>
<td>0.40</td>
<td>0.24</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.63</td>
<td>0.27</td>
<td>0.38</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>0.57</td>
<td>0.47</td>
<td>0.31</td>
</tr>
</tbody>
</table>
Table 3.8: Correlation between real kinematics values and predicted values using a codebook constructed with 128 codes for jaw, tongue, upper lip and lower lip data

Table 3.9 compares these correlation results to correlation values based on the null codebook by showing the mean and standard deviation of correlations from the 10-fold cross validation analysis. An unpaired t-test with 18 degrees of freedom show that the prediction from the codebook is significantly better than prediction form the null codebook, with a p-value of under 0.05 (t-value greater than 1.73) for all speech articulators.
Table 3.9: Mean and Standard deviation for correlation for predictions using 512 code codebook for the three speakers, compared to the mean and standard deviation of correlation of predictions from a NULL distribution

<table>
<thead>
<tr>
<th></th>
<th>VQ Codebook Mapping</th>
<th>Null Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>SD</td>
</tr>
<tr>
<td>Tongue Tip X</td>
<td>0.66</td>
<td>0.01</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>0.61</td>
<td>0.02</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>0.72</td>
<td>0.03</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>0.65</td>
<td>0.01</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>0.72</td>
<td>0.02</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>0.68</td>
<td>0.02</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.74</td>
<td>0.01</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>0.75</td>
<td>0.01</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.59</td>
<td>0.03</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>0.55</td>
<td>0.03</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.64</td>
<td>0.02</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>0.58</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Detailed graphs showing the distributions of correlations for predictions from different trials are shown in section B of the appendix.

3.4 Comparisons

3.4.1 Number of VQ codes

The figure 3.3 shows RMSD values for predicted versus real articulator kinematics for speaker A with different number of codes.
Figure 3.5: RMSD for predictions of speaker A’s articulators for codebooks with different number of codes

The figure 3.6 shows the correlation values between real and predicted kinematic data for speaker A with different number of codes.
Figure 3.6: Correlation for predictions of speaker A’s articulators for codebooks with different number of codes

The predictions show higher accuracy (i.e., higher correlation and lower RMSD) with increasing number of codes.

Table 3.10 shows the p-values (from a paired t-test) for RMSD of predictions using different number of codes. These p-values show that there was a statistically significant improvement in prediction performance between 128, 256 and 512 codes. This is because by increasing the number of codes, we can distinguish between more acoustic VTFFs, and hence we can distinguish between more sounds. As expected, the improvement in performance by using 512 codes instead of 128 codes was the most significant. However, while the improvement from using 512 codes instead of 256 codes was always significant, this was not always the case for using 256 codes instead of 128 codes, as shown for upper lip data. This may be because increasing to 256 codes from 128 codes results in adding 128 new codes, whereas increasing 512 codes from 256 codes results in adding twice that
many new codes.

<table>
<thead>
<tr>
<th></th>
<th>Codebook Comparison: RMSD p-values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>128 Codes vs 256 Codes</td>
</tr>
<tr>
<td>Tongue Tip X</td>
<td>0.002/3.83</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>0.05/1.83</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>0.01/2.82</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>0.005/3.25</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>0.02/2.4</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>0.003/3.57</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.22/0.81</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>0.18/0.96</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.11/1.32</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>0.09/1.45</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.005/3.25</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>0.05/1.83</td>
</tr>
</tbody>
</table>

Table 3.10: P-values comparing prediction performances using different number of codes. The degrees of freedom for the t-test are 9. Data presented as p-value/t-value

### 3.4.2 Comparison to Original Method by Hogden et al [21]

Table 3.11 compares RMSD values from this study to those reported by Hogden et al. [21] in their study using a more restricted dataset.
### Table 3.11

<table>
<thead>
<tr>
<th>Articulator</th>
<th>Current Codebook</th>
<th>Hogden et al Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tongue Tip X</td>
<td>3.19</td>
<td>2.08</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>2.57</td>
<td>2.16</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>2.73</td>
<td>2.08</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>3.85</td>
<td>2.16</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>2.90</td>
<td>2.25</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>3.06</td>
<td>1.93</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.71</td>
<td>0.95</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>1.16</td>
<td>1.75</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.50</td>
<td>0.66</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>1.03</td>
<td>0.54</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.63</td>
<td>2</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>0.95</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 3.11: Table showing RMSD values of predicted articulator positions reported by Hogden et al. [21] compared to the RMSD values found in our study. Codebooks are based on 256 codes and results from participant A are presented.

Table 3.12 compares the correlations between real and predicted kinematics reported by Hogden et al. [21] in their codebook study using a more restricted dataset with our correlation results.
<table>
<thead>
<tr>
<th>Articulator</th>
<th>Current Codebook</th>
<th>Hogden et al Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tongue Tip X</td>
<td>0.64</td>
<td>0.93</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>0.60</td>
<td>0.88</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>0.70</td>
<td>0.93</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>0.63</td>
<td>0.90</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>0.70</td>
<td>0.93</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>0.67</td>
<td>0.92</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.72</td>
<td>0.54</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>0.74</td>
<td>0.78</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.58</td>
<td>0.66</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>0.54</td>
<td>0.76</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.63</td>
<td>0.62</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>0.57</td>
<td>0.80</td>
</tr>
</tbody>
</table>

Table 3.12: Table showing correlations of predicted articulator positions reported by Hogden et al. [21] compared to the correlation results from our experiment. Codebooks are based on 256 codes and results are from participant A.

A direct comparison of these results from the two studies has to be done with some caution. The dataset for codebook used by Hogden et al. did not correct articulators for the position of the jaw. Hence the positions of the tongue and lower lip were in different reference frames across the two studies. NRMSD values for prediction performance were not reported by Hogden and colleagues. In the absence of these normalized RMSD values, it is difficult to make a comparison between the RMSD of predictions on datasets from different speakers. The correlation values shown by Hogden et al. have a higher correlation when predicting the position of the tongue, upper and lower lip, and lower correlation when predicting the position of the jaw. The tongue, upper and lower lip movements are significant for the production of consonants. The study by Hogden contained only one consonant (/g/) [21]. Furthermore, Hogden et al. [21] used the same acoustic stimuli to construct the training and test dataset. These factors may contribute to the higher correlation in the prediction of these speech articulators.
3.4.3 Comparison to Linear Coding

We see that the vector quantization codebook method used by Hogden et al. can produce accurate articulatory inversion predictions for a large and diverse dataset. It is interesting to compare the prediction performance of the codebook method to more complex machine learning methods used for articulatory inversion. In this subsection we compare the performance of the codebook method and the linear regression method for articulatory inversion using our dataset. We see that the prediction performance of these two methods is generally comparable. This indicates that the VQ codebook method which doesn’t rely on a complex mathematical modeling of data can perform articulatory inversion comparably to complex machine leaning approaches.

The vector quantization method and the linear regression method used in other studies have different approaches to accomplish the acoustic to kinematic mapping, and each method suffers from its own drawbacks. The vector quantization method takes into account the non-linear nature of the acoustic to kinematic mapping, but does this at the expense of ignoring the continuous nature of this mapping. The linear regression model takes into account the continuous nature of this mapping, but does this by approximating the non-linear relationship as a linear one. Both the vector quantization method we used in this study and the linear regression model fail to account for the phenomenon of motor equivalence. In the case where multiple configurations of speech articulators can produce the same acoustic value, both these methods make an attempt to map to a weighted average value of these configurations. This leads to degradation in performance for both of these methods.

The linear regression technique models the relationship between independent variable and multiple dependent variables as a linear equation. This linear equation is fitted by using a least squared approach. The independent variables for articulatory inversion are the acoustics parameters and the dependent variables are the speech articulator positions.

We compare the results of the prediction of articulatory positions from the acoustics found in this study to findings based on the linear regression technique presented by Craig et al. Measurements by Craig and colleagues were performed using a system where a video camera was used to track the location of markers placed on the speaker’s speech articulators. The video file from the camera was processed to determine the location and movement of these markers. Since the video camera is unable to track the
movement of speech articulators inside the mouth, the model presented in [9] did not include information about the movement of the tongue.

To implement this comparison between the two methods, we used the linear regression model as described in [9] on our dataset. This method used the root mean squared energy, 16th order linear predictor coefficients (LPC), 16th order line spectral pairs (LSP), and the first derivatives of the previous two sets. The linear regression model is represented by the equation shown in (3.1)

\[
\text{KinematicsParameters} = \text{Coeffecients} \times \text{AcousticParameters}
\] (3.1)

Kinematics parameters are shown in equation 3.2, acoustic parameters are shown in equation 3.3 and coefficients are shown in equation 3.4.
We constructed a linear regression model for the same 10 training sets that we used to construct VQ codebooks. These linear regression models were tested on their respective test sets, and their average performance (RMSD and correlation values) are reported below.

Figure 3.7 shows a comparison of the RMS errors for speaker A in this study using the codebook method (based on 512 codes) versus the linear regression method.
Figure 3.7: RMSD comparison of prediction from vector quantized codebook with 512 codes and linear regression model proposed by [9] on data from speaker A.

Figure 3.8 shows the comparison of the correlation values between predicted and real kinematic data for speaker A comparing the linear regression method used in [9] versus the codebook approach used in the current study.
Chapter 3. Acoustics To Kinematics - Part A

Once again, a direct comparison of these methods need to be done with some caution. The acoustic parameters used in this linear regression study by Craig and colleagues [9] (RMS energy, LPC, LPS and the derivatives of LPC and LSP) are different from those used in the current study (VTFF). While the LPC coefficients represent the VTFFs, the LSP coefficients and their derivatives more explicitly define the location of the formants as well as the change in energies across frequency values. The RMS energy takes into account the contribution of the glottal source, which we have ignored in our VQ mapping model by assuming the validity of the independent source filter model. It is not clear what the contributions of these additional parameters are in improving the predictions from the linear mapping model.

The linear regression model has comparable, and sometimes even better performance than the vector quantization codebook method. However, the vector quantization method
more accurately describes the non-linear relationship between acoustics and kinematics, and thus offers a higher upper-limit on prediction accuracy that may not be possible to achieve using linear regression. Chapter 4 shows a modification to the vector quantization mapping which improves the performance of this method over the linear regression method.

### 3.5 Source of Prediction Errors

The sources of prediction errors in this study can be divided into two categories; procedural errors and inherent errors due to the one-to-one codebook mapping. Procedural errors may be overcome by changing the data processing procedures. These include errors due to time lag between acoustics and kinematics, non-stationary kinematics in a window, improper removal of silence sections, suboptimal representation of acoustics values, and suboptimal clustering solutions. Inherent errors are those errors which are due to using a one-to-one codebook, and thus cannot be overcome unless this mapping technique is fundamentally changed.

We have analyzed various potential sources of procedural errors and explored possible solutions to them. However, we found that none of our potential solutions significantly improved prediction performance. Below, we list these sources of procedural errors and the solutions we have tried for them. A detailed description of our attempts to account for or address these errors is described in the Appendix section [D].

- **Time Lag** - Previous studies [21] [39] have shown that delaying the acoustics with respect to kinematic by about 15 ms may improve prediction performance. Adding this delay showed no improvement of predictions for our codebook (see appendix section [D.1]).

- **Non-stationary Kinematics** - The kinematics were not stationary in the 30ms frames we used for data analysis, with the tongue tip moving as much as 2.4mm during this interval (see appendix section [D.2]). We attempted to solve for this issue by creating a codebook using 10ms frames instead of 30ms frames. However this codebook showed no significant improvement in prediction performance.
• **Improper removal of silence sections** - Inclusion of silences in windows may lead to inaccurate VTFF calculations. Our silence removal procedure (see section 2.5.2) does not remove some short duration silences. Hence we may benefit from using a more robust silence removal techniques like [44].

• **Sub-Optimal representation of acoustics** - The VTFFs we use to represent the acoustic values had more energy (and more variability in energy) in the lower frequencies corresponding to the first formant, than in frequencies corresponding to higher formants. Since the VQ clustering procedure relies on Euclidean distance, it is more sensitive to the acoustic information related to the first formant. It is possible that we see better codebook predictions results if the VTFFs have a more equal weighting of energies in different formants. We attempted to solve this by creating a codebook on normalized (z-scored) VTFFs; however this showed no improvement on prediction performance.

In addition to the procedural errors described above, there were other procedural errors that we did not attempt to solve. These procedural errors are listed below.

• **Suboptimal clustering solution** - Up to 12% of the acoustics in the training set may be clustered into the wrong code. This is because we terminated the k-means clustering algorithm at a suboptimal solution. Since this issue is fundamental to the k-means algorithm (see 3.2.1), it can only be overcome by allowing the k-means algorithm to run for a substantially longer time to reach an optimal solution.

• **Insufficient data** - For a few test-training set pairs, there were acoustic data in the test set for which there was no similar acoustic representation in the training set. This may be due to insufficient amount of data, or insufficient variability in the dataset.

Inherent errors describe those errors which are fundamentally linked to using a one-to-one codebook technique for finding acoustic to kinematic predictions. Below we identify and describe two of these errors.

1. **Error due to quantization**
The inherent issue with the vector quantized codebook is that only as many articulatory configurations as number of codes can be recovered from the codebook. Even if the codebook technique captures the acoustic to kinematic relationship completely, one cannot expect perfect performance from the codebook as more than 512 articulatory positions can be used by the speaker during speech.

As suggested by Hogden et al [21], the best possible performance of the quantization technique can be found if clustering the acoustic domain also clusters the kinematic domain. This ‘optimal codebook’ can be simulated by clustering kinematics in the training set (instead of acoustics) and mapping kinematics in the test set to their closest kinematic cluster centers. Hence this optimal codebook ignored acoustics and directly clustered kinematics. This new codebook and mapping simulates a situation for the original codebook in which the acoustic to kinematic mapping is perfectly captured by the codebook i.e. if the distance between the acoustic values A and B is smaller than the distance between acoustic values A and C, then their corresponding kinematic vectors $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$ also show the same behavior.

Table 3.13 shows the performance of this revised codebook made by quantization of kinematics and comparing it to the original codebook based on quantizing acoustics. The findings show that even in the case where the codebook captures the acoustic-kinematic relationship perfectly, the inherent nature of the codebook will still lead to errors in prediction.
### Table 3.13: Comparison of predictions of an ideal codebook (made by quantization kinematics) with the original codebook. Table shows the presence of error even in the ideal case. Codebooks constructed with 512 codes on speaker A’s data.

<table>
<thead>
<tr>
<th></th>
<th>Original Codebook</th>
<th>Codebook by Quantizing Kinematics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tongue Tip X</td>
<td>3.16</td>
<td>0.96</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>2.55</td>
<td>1.2</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>2.71</td>
<td>0.5</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>3.81</td>
<td>0.78</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>2.88</td>
<td>1.15</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>3.04</td>
<td>0.73</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.70</td>
<td>0.95</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>1.15</td>
<td>0.45</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.48</td>
<td>0.78</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>1.03</td>
<td>0.46</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.63</td>
<td>0.74</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>0.94</td>
<td>0.49</td>
</tr>
</tbody>
</table>

2. **Articulatory variability within a cluster due to motor equivalence**

While the kinematics within a code generally showed lower variability than the kinematics in the entire dataset, often kinematics within a code can have a big difference between them.

Figure 3.9 shows two kinematics picked from the same code for a codebook constructed for speaker A with 512 codes. These two kinematic profiles show significant difference between them, and one reason for this difference could be found in the fact that speech production shows motor equivalence (different kinematic patterns can generate the same/similar acoustic patterns).
Figure 3.9: Kinematic configuration A and kinematic configuration B are both mapped to the same acoustic code.

The difference between these two kinematic configurations is significant. Figure 3.10 compares the distance between these two kinematics configurations, and the average distance between a training set from speaker A’s data. We see that the distance between the two kinematic configurations is more than the average distance between any two kinematic configurations in the training set for most speech articulators.
The averaged kinematic reference vector is often not a good representation of the kinematics corresponding to the acoustics within the cluster due to this articulatory variability. An analysis of the codebook (on speaker A’s data) also shows that for 98% of the acoustics in the training set, their corresponding kinematic values would be better predicted by a code different from the code the given acoustic signal was mapped to.

### 3.6 Discussion

The current study shows that the vector quantization (VQ) method presented by Hogden et al. [21] can be used to map acoustic to articulator positions for a large dataset of naturally spoken English sentences. Predictions made using this VQ codebook in the current study are significantly better than predictions from the null dataset. As discussed in section 3.4.2 there are many issues in making a direct comparison between the prediction performance reported in this study and in the study by Hogden and colleagues [21]. It is not appropriate to compare RMSD values in the absence of the information
about the amplitude of movement of speech articulations for the two speakers. Data for
the tongue and the lower lip are in different reference frames across the studies, making
a comparison of correlations reported by them difficult. Keeping these differences in
mind, we see from the RMSD comparisons in section 3.3.1 that the error in predictions
from the current codebook on natural speech has slightly higher RMSD for articulator
movement in the front-back (X) direction when compared to RMSD values reported by
Hogden et al. Correlation comparison in section 3.3.2 show that the correlation for the
tongue and the lips are lower, but the correlation of the jaw movement is greater for the
current codebook, compared to the codebook by Hogden and colleagues [21].

A drop in performance in our codebook when compared to the work by Hogden et al is
to be expected as we used a significantly bigger and more diverse dataset with the same
number of codes. In our current study, we clustered approximately 60000 frames of 30ms
acoustics, which represents about 30 minutes of data. In comparison, the Hogden et al.
codebook clustered less than 3 minutes of acoustic data. The Hogden et al codebook
clustered acoustic samples of two vowels (selected from a set of 10 vowels) spoken in
a /g/ context. The training dataset in our work contained 180 phonetically balanced
English sentences, with each sentence repeated twice. This represented a more diverse
acoustic dataset when compared to the training dataset used by Hogden and colleagues.
By restricting the speech stimuli, the latter study has less of a chance having to deal with
motor equivalence during speech. In the absence of motor equivalence, the articulatory
positions used to produce the acoustics within a cluster will be more similar than in the
case when there is motor equivalence. Hence, averaging kinematic position values, as was
done in both these studies, may be a more effective strategy for the restricted dataset of
Hogden et al. [21].

Apart from the differences in the dataset, there are other factors that may contribute to
the difference in performance between the two studies. Some of these differences can be
attributed to the difference in measuring equipment. For the current study, we use the
AG501 system which has been shown to have higher accuracy than the three transmitter
EMA system used by Hogden et al. [24]. In addition, the three transmitter system fails
to accurately reflect articulator positions once the participant’s head moves away from
the mid-line of the EMA machine. The effect of these errors on the codebook is not clear.
Another potential source of difference is that the kinematic dataset used by Hogden and
colleagues does not map certain articulatory positions to different reference frames. In
the current study we mapped the jaw and upper lip positions to that of the participant’s
head’s reference frame, and the lower lip and tongue to that of the jaw. Finally, it should be noted that in Hogden et al study, the training set and the test set were made up of different repetitions of the same acoustic stimuli. In our study, we constructed the test set from sentences which are different from sentences used for training the codebook. This makes the prediction problem more difficult in our study. Despite these differences, we see that the performance of the codebook by Hogden et al. [21] and the performance of the codebook in the current study are comparable.

When compared to linear regression technique presented in section 3.4.3, we see that the current VQ codebook shows poorer prediction performance (for several articulators). It is not clear if this improvement is due to difference in techniques, or due to additional acoustic parameters used by the linear regression technique in describing the acoustic dimensions. Unlike the one-to-one codebook technique which restricts predictions to one out of a set number of possible kinematic predictions (equal to the number of codes; 512, 256 or 128 in this study), the linear regression model is able to map acoustics to an infinite set of kinematic values. This restriction on possible solutions for the VQ codebook creates an inherent source of error for the one-to-one codebook (see section 3.5 inherent source of error) that does not exist in the linear regression model. The linear regression model attempts to model the acoustic to kinematic relationship (which is inherently non-linear [59] and not a functions mapping [40]) as a linear relationship. Hence the linear regression model has an upper limit on accurate production performance, which is lower than the potential upper limit offered by the VQ codebook technique. Chapter 4 shows that by increasing the set of possible predicted solutions, the VQ technique can outperform the linear regression technique.

Chapter 4 presents a new acoustic to kinematic mapping technique which takes the inherent issues with the mapping technique described in this chapter into account. As discussed in section 3.5, the one-to-one codebook method used here suffers from some inherent flaws. This technique restricts possible prediction solutions to as many configurations as there are codes in the codebook. Since only kinematic reference vectors are used as prediction solutions, we rely on these few (512, 256 or 128) reference vectors to accurately predict all the kinematics in the test set. Furthermore, this technique ignores the presence of motor equivalence in the dataset. In the case that there are similar acoustics in the training set which may correspond to different kinematic configurations, we use the average of these kinematic configurations to predict the acoustic.
A more accurate acoustic to kinematic mapping technique is presented in chapter 4 and considers all kinematic values in a cluster as possible solutions, instead of just their average values (i.e. the kinematic reference vectors). Since each cluster contains roughly 100 values (dataset size divided by number of codes), this increases the number of possible solutions by a 100 times. In cases where some of these 100 kinematic configurations are very different from each other (i.e. in cases of motor equivalence), a selection strategy is needed which takes all 100 kinematic configurations into account to accurately predict kinematic values despite motor equivalence in the dataset. Chapter 4 discusses how such strategies can be implemented in order to enable us to choose one correct prediction solution from these 100 possible solutions.
Chapter 4

Acoustics To Kinematic Mapping - Part B : Optimized Vector Quantization Approach

4.1 Description

In this chapter, we present a codebook mapping strategy which improves upon the mapping strategy from the previous chapter by taking into account that the solution to the articulatory inversion problem is non-unique. This means that multiple articulatory positions can be used to produce the same acoustic. This property of the speech production is called Motor Equivalence or Articulatory Compensation. Work by Atal et al. [4] showed that an infinite combination of vocal tract shapes could generate acoustics with identical formant positions. A study by Qin and Carreira-Perpin [40] shows support for the observation that the acoustic to articulatory mapping is not unique during speech. We observe a similar phenomenon in our dataset. Figure 3.9 in section 3.5 shows that very different kinematic configurations may be mapped to the same acoustic cluster. In order to take into account these multiple solutions for articulatory inversion, the technique presented in this chapter modifies the vector quantization method from the previous chapter to increase the solution space to a set of possible kinematic patterns instead of a single averaged solution.
We created a one-to-many vector quantization codebook which reflects the one-to-many acoustic to kinematic relationship. The acoustic data were clustered similar to the technique presented in the previous chapter by running a vector quantization clustering technique to find clusters and cluster centers (acoustic reference vectors) in the acoustics. However, whereas in the previous chapter we averaged the kinematics corresponding to the acoustics in a cluster, in the current study we preserve all these kinematic values to form a kinematic reference vector set. This results in a one-to-many codebook where each acoustic reference vector maps to a kinematic reference vector set. All kinematic configurations in this kinematic reference vector set are treated as possible solutions to articulatory inversion.

We used a kinematic cost optimization method to pick a specific correct kinematic solution for articulatory inversion from a kinematic reference vector set. We used a systematic strategy for selecting a kinematic value because phonetic studies have shown consistency in the kinematic configurations a speaker uses to produce a sound. These consistencies manifest themselves as specific articulatory gestures which are defined as units of coordinated movements of speech articulators [7]. These gestures are not only used by a speaker over multiple utterances of the same sound, but are also used consistently for the same phoneme regardless of speaker. The theory of articulatory phonology uses the concept of gestures for defining lexical units [6]. Hence, according to this theory, gestures are the basic units of speech planning, and speech acoustics are emergent properties of the speech production system. On the other hand, The Directions Into Velocities of Articulators (DIVA) [56] model considers acoustics as the basis of speech production and perception and suggests a cost optimization mechanism is used to plan articulatory motions. Regardless of the theoretical basis of articulatory gestures, there is evidence that articulatory gestures are used in speech production. This suggests principles of organization of articulatory patterns used during speech, and we attempt to uncover these principles using kinematically based cost optimization techniques.

We evaluated the performance of the codebook by optimizing three kinematic cost functions, minimizing movement calculated as the minimization of the summation of the magnitude of velocity, minimizing acceleration calculated as the minimization of the summation of magnitude of acceleration and minimizing jerk calculated as minimizing the summation of magnitude of jerk. These cost optimization strategies are based upon other studies in human motor control where movements have been shown to obey principles of minimization of some kinematic cost. Perhaps the most influential work in this
area is done by Flash and Hogan [12] who showed that arm trajectories in reaching tasks follow a path of minimization of jerk (which is defined as the rate of change of acceleration). Work by Neufeld and Van Lieshout [36] indicates that the tongue motion during speech production also shows a similar type of jerk minimization. This principle of minimizing kinematic cost shown for articulators has inspired work on articulatory inversion to use similar principles to inform the articulatory inversion solution, as detailed next.

Many studies have incorporated a minimum cost path principle for finding the correct solution for articulatory inversion. However, most work on this topic is restricted to a theoretical analysis of the minimum cost solution without presenting empirical results. For example, Atal et al. [4] presents a theoretical framework where speech articulators move as little as possible while producing speech acoustics. Similarly, Tourville and Guenther [56] present a theoretical framework where the articulatory inversion solution is biased towards a certain comfortable articulator position. Studies that do present empirical data are restricted to optimizing for only minimum distance paths. Work by Sorokin, Leonov and Trushkin [49] and work by Richards et al. [42] have shown that imposing the minimum movement constraint on real speech data does indeed improve articulatory inversion predictions. The few studies that take into account minimum acceleration and jerk solutions, like the work by Ghosh and Narayanan [17], may have incorrectly assumed linearity of the kinematic to acoustic relationship for small movements of articulators. These studies with empirical data show that while optimizing for kinematic costs improves the articulatory inversion predictions, this prediction is not perfect. It is not clear why this is the case, but so far little analysis has been done on the kinematic cost properties of real kinematic patterns used by a speaker.

The current study presents articulatory inversion solutions using a one to many acoustic to kinematic codebook based on real speech data. We derive solutions from the optimization of kinematic costs, defined as functions that minimize kinematic parameters (distance, acceleration and jerk). Specifically, we present an analysis of the properties of the kinematic patterns used by a speaker in the context of kinematic cost, and discuss the reasons for the improvement in performance using functions to minimize these costs, as well as the reasons why these solutions do not show perfect prediction results.
4.2 Methods

The procedure for constructing the one-to-many codebook in this study was built upon the one-to-one codebook from the previous chapter. Codebooks were constructed from the training set using a vector quantization method. Kinematics predictions were derived from acoustics in the test set and these predictions are compared to real kinematic values.

Similar to the previous chapter, we used a 10-fold cross validation technique in order to evaluate prediction performance. This evaluation technique lead to the construction of 10 training-test set pairs as described in section 2.5.5. However, in addition to these 10 sets, we constructed 10 additional sets for which a test dataset was included in its corresponding training dataset. These new training-test dataset pairs allowed us to study the performance of the mapping strategies when the correct solution to articulatory inversion was guaranteed to exist in the training set.

We derived predictions using three selection strategies. These were minimizing movement calculated as the minimization of the summation of the magnitude of velocity (= distance traveled), minimizing acceleration calculated as the minimization of the summation of magnitude of acceleration and minimizing jerk calculated as minimizing the summation of magnitude of jerk. For simplicity, these kinematic cost functions are henceforth referred to as minimum movement, minimum acceleration and minimum jerk respectively. Since finding the minimum kinematic cost solutions required significant computational time, these mapping strategies were only evaluated on speaker A’s data.

In order to compare results from this study with results from the study in the previous chapter, we performed predictions using a 512 code codebook. However, due to limitation of the 512 codebook in calculating the minimum jerk prediction (explained in section 4.2.4); we also performed predictions using a 2048 code codebook.

Hence, for each of the 20 training-test dataset pairs, two codebooks were constructed (with 512 and 2048 codes). For each codebook, three sets of predictions were made using the three kinematic cost minimization selection strategies.
4.2.1 Processing Dataset

As explained in section 2.5.5, the dataset from the speaker was broken into a training datasets and a test datasets. The training datasets contained acoustic and kinematic values used to construct the one-to-many acoustic to kinematic codebook. The test dataset contained the acoustic values which we used to perform predictions from the codebook, and kinematic values which we used to evaluate the accuracy of these predictions. In this study, we constructed additional codebooks made on training sets which were different from those used in the previous chapter. Additionally, while the test dataset was the same for this chapter and the previous chapter, we defined regions in the test dataset (called continuous regions) which we used to evaluate the codebook performance. These modified training datasets, and continuous regions in the test dataset are explained below.

As in the case of the previous chapter, a 10-fold cross validation method was used (explained in 2.5.5) to construct 10 training and test set pairs. These training and test sets were constructed based on sets of sentences which are mutually exclusive. However, in addition to the 10 training sets constructed for 10-fold cross validation, we constructed additional training sets based on each of the 10 original training sets. The new training sets were constructed by appending data from the test set into the training set. This led to an additional set of 10 new training set and test set pairs where the test set was contained within the training set.

The new training-test set pairs allow us to study the performance of the codebook when articulatory inversion solution is guaranteed to exist in the training set (and consequently in the codebook). A source of error in the one-to-one codebook in the previous chapter was that there existed kinematic configurations in the test set with no similar kinematic configuration in the training set (shown in section 3.5). Hence, it was difficult to know if the error in performance of the codebook was due to this absence of the correct solution from the codebook, or due to some inherent flaw in the codebook mapping strategy. By adding the test set to the training set, we could study the codebook performance when the correct solution to articulatory inversion was contained in the codebook.

Articulatory inversion prediction was performed on acoustics in the test set using a kinematic cost minimization strategy. The acoustics over which this kinematic cost was minimized was defined as the acoustics between silent regions. Section 2.5.2 describes that the silent removal procedure in detail. Acoustics in 30ms frames were analyzed for
silences, and silent frames were removed. Hence, consecutive 30ms frames of acoustics that were not silent formed continuous acoustics. Figure 4.1 shows continuous acoustics marked on a sample acoustic section.

Figure 4.1: Figure shows an acoustic section in the time domain roughly 1 second in length. Vertical lines are separated by 30ms and acoustics between these lines represent acoustic in frames. Acoustics frames that are not silent are shaded. Continuous acoustics are marked on the figure as consecutive frames of acoustic which are not silent.

### 4.2.2 Construction of the Codebook

The codebook used in this study was a one-to-many acoustic to kinematic codebook. It mapped an acoustic reference vector to a set of kinematic reference vectors.

This codebook was based upon the one-to-one codebook from the previous chapter described in section 3.2.1 with identical procedures for finding acoustic reference vectors. As in the case of the one-to-one codebook, a k-means algorithm was used to cluster the acoustics (i.e. the vocal tract filter functions, from here on referred to as VTFFs).
center of an acoustic cluster was the acoustic reference vector for that cluster. As many clusters are created as number of codes required in the codebook. A detailed description of the k-means algorithm and configurations used to implement it are described in section 3.2.1 of the previous chapter.

Where in the previous chapter, kinematics associated with a cluster were averaged to find the kinematic reference vector, in this study we preserved these kinematic values to construct a kinematic reference vector set of possible solutions. The k-means algorithm finds clusters containing roughly the same number of data points in each cluster. Hence, by increasing the number of acoustic clusters, we decrease the number of datapoints in a cluster and as a results also decrease the number of kinematic configurations capable of producing an acoustic pattern in that cluster. With a training dataset containing roughly 60000 data points (shown in section 2.5.5), a kinematic set for a 512 code codebook contains roughly 100 possible solutions ($\approx \frac{60000}{512}$), and a kinematic set for a 2048 code codebook contains roughly 25 possible solutions ($\approx \frac{60000}{2048}$).

4.2.3 Mapping Acoustics To Kinematics

In this chapter, we present three strategies based on minimization of distance, acceleration and jerk. All of these techniques have two parts, with the first past for each technique being common. The first part of each technique involved using the vector quantized codebook to map acoustics to a kinematic reference vector set. A kinematic reference vector set was defined as a set of all the kinematic configurations from the dataset that could have produced the given acoustic. The second part involved using a strategy to pick one kinematic configuration from the predicted kinematic reference vector set.

Figure 4.2 shows the schematic of the methods used in this chapter to perform acoustic to kinematic predictions. Part A in this schematic shows the division of the test dataset into continuous acoustics, which has been discussed in 4.2.1. Part B in this schematic shows the mapping from acoustics to kinematic reference vector set. Part C shows the selection of kinematic values from the kinematic reference vector set based on kinematic cost optimization. Part B and Part C are explained below in subsections Acoustics to Kinematic Reference Set and Selecting Kinematic Values from the Kinematic Set respectively.
Chapter 4. Acoustics To Kinematics - Part B

Figure 4.2: Schematic showing acoustic to kinematic predictions from the one-to-many codebook used in this chapter.

**Acoustics to Kinematic Reference Set**

Mapping is performed to predicted kinematic values corresponding to acoustic values in the test set. In this study, the test dataset was broken into regions of continuous acoustics as explained in section 4.2.1. These continuous acoustic regions were roughly 1 second, or 30 frames (of 30ms), of data corresponding to an uninterrupted acoustic segment spoken by the speaker.
The technique to map acoustics to kinematic reference set is similar to the mapping technique presented in chapter 3. However, instead of mapping an acoustic value to an average kinematic reference value, we now mapped an acoustic value to a kinematic reference set.

An acoustic (VTFF) is mapped to a code from the codebook by finding the acoustic reference vector closest to the given VTFF. The kinematic reference vector set for this code comprised the predicted possible kinematic values for the given acoustic reference. Figure 4.3 shows the process of associating an acoustic value to an acoustic and kinematic reference set using a codebook defined by acoustics (represented as a one dimensional value) and kinematics (represented as abstract values $K_n$).
Figure 4.3: An abstract codebook with six codes. Acoustics are in one dimension and kinematics are represented by abstract $K_n$ values. Acoustic value 6.6 is mapped to code with acoustic center 5.9 and thus the predicted kinematic reference vector contains configuration values K10, K11 and K12.

Figure 4.4 shows the schematic to find a kinematic reference set for each frame of acoustic data in the continuous acoustic region from the test dataset. This figure depicts in greater detail the Part B section of figure 4.2.
**Chapter 4. Acoustics To Kinematics - Part B**

76

Figure 4.4: Schematic showing the process of mapping acoustic frames in a continuous acoustic segment to kinematic reference vectors. This figure depicts in greater detail the Part B section of figure 4.2.

**Selecting Kinematic Values from the Kinematic Set**

The articulatory inversion solution was found for continuous acoustics, which are a set of consecutive acoustic frames without silences (as defined in section 4.2.1). Each acoustic frame was mapped to a kinematic reference vector set using the method described in the sub-section above. A possible articulatory inversion solution for the continuous acoustic was found by selecting one kinematic reference value from each reference vector. This possible articulatory inversion solution is called a kinematic path. Figure 4.5 shows an example kinematic path. The 'Acoustics to kinematic reference vector sets' mapping marked on this figure is described earlier in sub section Acoustics to Kinematic Reference Set.
In this chapter we find the articulatory inversion solution as the kinematic path that optimizes a kinematic cost function. These kinematic cost functions are defined as total distance traveled by speech articulators, or as the summation of magnitude of acceleration or jerk of the movement of the speech articulators.

A trivial solution to find the kinematic path of minimum cost is to find all possible kinematic paths, and to select the one with the lowest cost (i.e., using an exhaustive search technique). However, this is computationally challenging making as the number of possible solutions is very large (explained in detail in section 4.2.4). This makes the exhaustive search technique practically unfeasible. Hence, we used a modified Dijkstra’s shortest path in a graph solution to find the path of minimum cost [10]. A similar solution was presented by Schroeter and Sondhi [47] for finding the kinematic path of
minimum kinematic distance for articulatory inversion. The minimum acceleration and jerk solutions were implemented as modifications to the method used by Schroeter and Sondhi [47] to find the optimal distance solution.

Dijkstra’s algorithm is a graph search algorithm, and hence in order to use it in our application, we needed to frame our articulatory inversion problem as a graph. The graph must have the following properties: a) A traversal of the graph from a beginning node to an end node must contain a possible solution to articulatory inversion (i.e., it must be a kinematic path as described in figure 4.5), b) The cost between two nodes in the graph should be found using only those two nodes. Below we show how such graphs can be constructed in order to find the minimum distance, minimum acceleration and minimum jerk solutions.

1. Minimum Distance Solution

We used a modified Dijkstra’s shortest path approach to find the path of minimum movement from all the paths in the codebook. A detailed explanation of this technique can be found in the appendix section [E]. The construction of a graph to which we apply this algorithm is shown in figure 4.6 and described below.

In order to find the kinematic configurations for n consecutive frame of acoustics (belonging to a continuous acoustic), we found the codes from the codebook corresponding to each of these n acoustics. All the values in the kinematic reference vector for a code are represented as nodes. In figure 4.6, these values are represented as \( K_i^j \) where \( i \) is the frame number, and \( j \) is the \( j^{th} \) kinematic reference vector in the kinematic reference vector set. Each node is connected to each other node in the subsequent frame, and to no other node. Additionally, the connections are directed, i.e., they can only be traversed one way (node on the left, to node on the right). Due to the structure of the graph, a traversal of a graph from a node in the first frame, to a node in the last frame, contains exactly one kinematic reference vector value from each frame. Hence this traversal contains a kinematic path. All traversals of this graph represent all possible kinematic paths.

Dijkstra’s algorithm allows us to efficiently find which kinematic path shows minimum movement. Movement in a path was calculated as the sum of movements between each subsequent node in a kinematic path. Movement between two nodes was calculated as the sum of the magnitude of change in the positions of all speech
articulators between the two nodes. This is expressed in the equation (4.1)

\[
Movement\_Between\_Nodes = \sum |(K_{i+1}^k - K_i^j)|
\]  

(4.1)

Figure 4.6: Figure shows the structure of the graph used to run a modified Dijkstra’s algorithm on the kinematic data to find the path of minimum cost. Each node represents a kinematic configuration that can be used to produce the acoustic associated with that frame. This graph shows each frame as having an equal number of nodes, but this is not necessarily the case. Additionally, arrows are drawn as a subset of possible connections to keep the graph simple.

The time complexity of the minimum distance solution is given in section 4.2.4

2. Minimum Acceleration

Acceleration is calculated as the difference in velocity of kinematic from one frame to the next. Each \( K_i^j \) kinematic reference vector values represents a static position
of kinematics, and hence acceleration can be calculated by the equation 4.2

\[
\text{Acceleration Between Nodes} = \sum |(K_{r_{i+2}} - K_{r_{i+1}}) - (K_{s_{i+1}} - K_{s_{i}})|
\]  

(4.2)

Hence, we require three \(K^j_i\) kinematic reference vector values to calculate acceleration. In figure 4.6 we notice that each node only contains one kinematic reference vector, and hence we cannot calculate the acceleration between two nodes. Dijkstra’s algorithm requires that we are able to calculate cost between two nodes using only those two nodes. In this section, we propose modifications to the graph in figure 4.6 which enable us to calculate the acceleration cost. This modified graph is shown in figure 4.7 and a detailed description of these modifications is given below.

In order to find the kinematic configurations for \(n\) consecutive frame of acoustics, we found the codes from the codebook corresponding to each of these \(n\) acoustics. These codes were used to find a kinematic reference vector set for each frame. A node was constructed containing two values, with the first value chosen from the kinematic reference vector set of the current frame’s code, and the second value chosen from the kinematic reference vector set of the next frame’s code. Nodes were constructed for a frame such that all values from the current frame’s kinematic reference vector and all values from the subsequent frame’s kinematic reference vector were chosen. If there are \(m\) values in a kinematic reference vector (values in a kinematic reference vector may change between codes, which for simplicity is assumed to be constant in this example), there were a total of \(m^2\) nodes in a frame. Each node in a frame was connected to all nodes in a subsequent frame where the second value in the current node was the same as the first value in the node in the next frame. The connections were directed, i.e. they can only be traversed one way (node on the left, to node on the right). A traversal of this graph that started from any node in the first frame, and ended at any node in the last frame (kinematic path) contains exactly one node from each frame. By considering both values in the first node of a kinematic path, and the second values of all the other nodes; we notice that the kinematic path contains one possible solution for articulatory inversion. Furthermore, all kinematic paths represent all possible solutions for articulatory inversion.

The Dijkstra’s minimum cost path solution finds the path that minimizes total acceleration. Acceleration in a path is calculated as the sum of acceleration values
between each subsequent node in a kinematic path.

\[
\text{Acceleration\_In\_Path} = \sum \text{Acceleration\_Between\_Nodes} \quad (4.3)
\]

A kinematic path in the graph contained \( n-1 \) nodes, and the summation in the above equation is found over all subsequent nodes in this kinematic path.

Acceleration between two nodes was calculated as the sum of the magnitude of acceleration in the positions of all speech articulators between the two nodes. From the figure 4.7, we see that a node in the \( i^{th} \) frame can be represented as \( K_{i}^{j}, K_{i+1}^{k} \). Here \( j \) and \( k \) indicates that the node contains the \( j^{th} \) and \( k^{th} \) reference vectors from the \( i^{th} \) and \( i + 1^{th} \) reference vector set. The acceleration between nodes \( K_{i}^{s}, K_{i+1}^{j} \) and nodes \( K_{i+1}^{j}, K_{i+2}^{r} \) is found as

\[
\text{Acceleration\_Between\_Nodes} = \sum | (K_{i+2}^{r} - K_{i+1}^{j}) - (K_{i+1}^{j} - K_{i}^{s}) | \quad (4.4)
\]

This equation can be re-expressed as

\[
\text{Acceleration\_Between\_Nodes} = \sum | K_{i+2}^{r} - 2 \cdot K_{i+1}^{j} + K_{i}^{s} | \quad (4.5)
\]

Since \( K \) is a multidimensional quantity with as many dimensions as speech articulators measured, the summation in equation 4.6 sums over the magnitude of acceleration over all the dimensions.
Figure 4.7: Figure shows the structure of graph used to run a modified Dijkstra’s algorithm on the kinematic data to find the path of minimum acceleration. This graph shows each frame as having an equal number of nodes, but this is not necessarily the case. Additionally, arrows are drawn as a subset of possible connections to keep the graph simple.

3. Minimum Jerk
The previous subsection *Minimum Acceleration* described the modifications to the graph used to find minimum distance path (shown in figure 4.6) to construct the graph used to find minimum acceleration path (shown in figure 4.7). In this section, we propose similar modifications to the graph used to calculate the minimum acceleration path, in order to construct a graph that is able to calculate the minimum jerk path. Both the rational and steps for these modifications are very similar to the those for the modification to the minimum distance path graph presented in *Minimum Acceleration* subsection above. Hence we do not repeat these in this subsection. This modified graph is shown in figure 4.8.

The Dijkstra’s minimum cost path solution finds the path that minimizes total jerk. Jerk in a path is calculated as the sum of jerk values between each subsequent node in a kinematic path.

\[
Jerk\_In\_Path = \sum Jerk\_Between\_Nodes
\]  
(4.6)

A kinematic path in the graph contains n-2 nodes, and the summation in the above equation is found over all subsequent nodes in this kinematic path.

Jerk between two nodes is calculated as the sum of the magnitude of jerk in the positions of all speech articulators between the two nodes. From the figure, we see that a node in the \(i^{th}\) frame can be represented as \(K_{i}^{j}, K_{i+1}^{k}, K_{i+2}^{l}\). Here, \(j, k\) and \(l\) indicate that the node contains the \(j^{th}\), \(k^{th}\) and \(l^{th}\) reference vectors from the \(i^{th}, i+1^{th}\) and \(i+2^{th}\) reference vector set. The acceleration between nodes \(K_{i}^{r}, K_{i+1}^{s}, K_{i+2}^{t}\) and nodes \(K_{i+3}^{u}, K_{i+4}^{v}\) is found as

\[
Jerk\_Between\_Nodes = \sum | (K_{i+3}^{u} - K_{i+2}^{t}) - (K_{i+2}^{t} - K_{i+1}^{s}) | - (K_{i+2}^{t} - K_{i+1}^{s}) - (K_{i+1}^{s} - K_{i}^{r}) | \\
(4.7)
\]

This equation can be re-expressed as

\[
Jerk\_Between\_Nodes = \sum | K_{i+3}^{u} - 3 * K_{i+2}^{t} + 3 * K_{i+1}^{s} - K_{i}^{r} | \\
(4.8)
\]

Since \(K\) is a multidimensional quantity with as many dimensions as speech articulators measured, the summation in equation 4.8 sums over the magnitude of jerk over all the dimensions.
Figure 4.8: Figure shows the structure of graph used to run a modified Dijkstra’s algorithm on the kinematic data to find the path of minimum jerk. This graph shows each frame as having an equal number of nodes, but this is not necessarily the case. Additionally, arrows are drawn as a subset of possible connections to keep the graph simple.
4.2.4 Computational Considerations

This section describes the time complexity of the *minimum distance*, *acceleration and jerk* solutions. We compare the time complexities of these solutions found using Dijkstra’s algorithm and using an exhaustive search technique. Finally, we analyze the effect of the number of codes in a codebook on the computations taken to find the optimal kinematic cost solutions. A detailed analysis of the calculations involved in finding the time complexities stated in this section can be found in the appendix section F.

We calculate the time complexities for the minimum kinematic cost solution for a graph where \( n \) is the number of frames in the continuous acoustic, and \( m \) is the number of nodes in the kinematic vector set for a given frame.

The *minimum distance* path found using Dijkstra’s algorithm for a continuous acoustic has a time complexity given by \((n - 1) \times m^2\). The *minimum distance* path found using an exhaustive search for a continuous acoustic has a time complexity given by \( m^{n-1} \). We see that the Dijkstra solution is faster than the exhaustive search solution by a factor of \( m^{n-3}/(n - 1) \).

The *minimum acceleration* path found using Dijkstra’s algorithm for a continuous acoustic has a time complexity given by \((n - 2) \times m^4\). The *minimum acceleration* path found using an exhaustive search for a continuous acoustic has a time complexity given by \( m^{2(n-2)} \).

The *minimum jerk* path found using Dijkstra’s algorithm for a continuous acoustic has a time complexity given by \((n - 3) \times m^6\). The *minimum jerk* path found using an exhaustive search for a continuous acoustic has a time complexity given by \( m^{3(n-3)} \).

While the Dijkstra solution reduced the number of computations required to calculate all minimum cost paths, calculating the minimum jerk solutions (which requires \((n - 3) \times m^6\) computations) still represented a significant computational challenge. The number of computations required can be decreased by decreasing the value of \( m \) (i.e. number of nodes per frame). For a training dataset with \( V \) data points, the number of values in the cluster (which is the same as the number of values in the kinematic reference vector set) is on average given by \( m_N = V/N \), where \( N \) is the number of codes. Hence increasing the number of codes brings down the complexity of the minimum jerk path calculation.
For this reason, we used 2048 codes codebook instead of 512 code codebook to find the minimum jerk solution. The complexity of the minimum jerk solution using 2048 codes is smaller than the same calculations using a 512 code codebook by a factor of 4096 ($= ((n - 3) \ast (V/512)^6)/((n - 3) \ast (V/2048)^6) = 2048^6/512^6$).

### 4.2.5 Model Evaluation

The performance of the mapping and selection strategies was evaluated by comparing the predicted kinematic positions to known positions using correlation and RMSD measures. Mapping was evaluated under two conditions; the first with separate training and test datasets, as used in the previous chapter, and the second with the test set included in the training dataset. The first condition was evaluated using a 10-fold cross validation method as explained in section 2.5.5. The second condition was also evaluated using a 10-fold cross validation method by including the test dataset in the training dataset for each test-training dataset pair used in the first condition. This second paradigm was used to evaluate the performance of selection strategies when the solution to articulatory inversion is guaranteed to exist in the training dataset.

The performance of the mapping strategies was compared to the performance of the one-to-one codebook mapping from the previous chapter. Additionally, we compared these mapping strategies to two type of null sets. The first type of null set is constructed as described in the previous chapter in section 3.2.3. This null set was made from predictions from a codebook where the acoustics and kinematics were misaligned. The second type of null set was constructed by choosing random paths from the codebook as the predicted path. The performance from a large number of these random paths (100000 paths) was averaged to find the average prediction performance of choosing a random path from the codebook.

One-to-many codebooks were constructed using 512 codes and 2048 codes. We used 512 codes in order to compare the performance of this codebook with the performance of the codebook in the previous chapter. We used 2048 codes as the solutions to the minimum jerk path is computationally difficult to calculate for a 512 code codebook, as detailed in section 4.2.4.
4.3 Results

4.3.1 Minimum Cost Path With Separate Training and Test Datasets

This section shows the performance (in terms of correlation and RMSD values) of the minimum kinematic cost path when the test dataset was not included in the training dataset (i.e. the first test paradigm in section 4.2.5). These minimum cost paths were found using a 512 code codebook as well as a 2048 code codebook. However, due to computational challenges explained in section 4.2.4, the minimum jerk path was not calculated for the 512 code codebook. The predictions from these minimum cost paths were compared to the predictions from the one-to-one codebook and the null distributions from the previous chapter. Additionally, we compared the predictions to the average predictions from a random path (as described in section 4.2.5). A statistical comparison between the minimum cost path and the random paths is presented in section 4.4.1.

Correlation

Table 4.1 shows how well the predicted positions of speech articulators are correlated with their real positions for a 512 code codebook. Predicted paths shown minimize the summation of the absolute value of the articulator velocity (minimum distance path) and absolute value of the articulator acceleration. Predictions were performed on speaker A’s dataset using a 512 codes one-to-many codebook. The path that minimizes the summation of magnitude of jerk was not included in this analysis on the 512 code codebook because of the complexity of this calculation, as explained in section 4.2.5. Also shown on the table is the average performance of a random path from the codebook (as explained in section 4.2.5).
Table 4.1: Correlation of minimum kinematic cost paths with the real path. Results from 512 code codebook based on data from speaker A are shown.

The correlation results for the 512 code codebook show that the kinematic paths were predicted with slightly poorer accuracy when compared to the prediction performance of the one-to-one vector quantization codebook of chapter 3. The predictions however were significantly better when compared to the null distribution used in Chapter 3. The prediction correlation from chapter 3 as well as the null distribution for predictions can be found in section 3.3.2. The predictions from the minimum cost paths were better than the average predictions from a random path.

Table 4.2 shows how well the predicted positions of speech articulators were correlated with their real positions for a 2048 code codebook. Predicted paths shown minimize the summation of the absolute value of the articulator velocity (minimum distance path), acceleration and jerk. Predictions were performed on speaker A’s dataset using a 2048 codes one-to-many codebook. The performance of the mapping strategies were compared to the performance of the one-to-one codebook recalculated using 2048 codes, and to the average performance of a random path (as explained in section 4.2.5). Both the performance of the one-to-one codebook, as well as the random path using a 2048 code
codebook is shown in the table 4.2.

<table>
<thead>
<tr>
<th></th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>One to One Codebook</td>
</tr>
<tr>
<td>Tongue Tip X</td>
<td>0.67</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>0.59</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>0.72</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>0.65</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>0.71</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>0.67</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.72</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>0.73</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.61</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>0.57</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.66</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>0.60</td>
</tr>
</tbody>
</table>

Table 4.2: Correlation of minimum kinematic cost paths with the real path. Results from 2048 code codebook based on data from speaker A are shown.

Once again, predictions from paths of minimum cost were slightly poorer than predictions from the one-to-one codebook. However, these predictions were better than the average predictions from a random path.

**RMSD**

Table 4.3 shows the root mean squared deviation (RMSD) of the predicted kinematic path from the real path for a 512 code codebook. Predicted paths shown minimize the summation of the absolute value of the articulator velocity (minimum distance path) and absolute value of the articulator acceleration. Predictions were performed on speaker A’s dataset. The performance of the mapping strategies was compared to the performance
of the one-to-one codebook (see table 3.2 in previous chapter) and to the average performance of a random path (as explained in section 4.2.5).

\[
\begin{array}{|c|c|c|c|}
\hline
\text{RMSD} & \text{Minimum Summation} & \text{Minimum Summation} & \text{Random Path} \\
& \text{Magnitude} & \text{Magnitude} & \\
& \text{Velocity Path} & \text{Acceleration Path} & \\
\hline
\text{Tongue Tip X} & 3.14 & 3.27 & 4.8 \\
\text{Tongue Tip Z} & 2.57 & 2.79 & 3.8 \\
\text{Tongue Dorsum X} & 2.70 & 2.82 & 4.23 \\
\text{Tongue Dorsum Z} & 3.49 & 4.03 & 5.2 \\
\text{Tongue Back X} & 2.85 & 2.98 & 4.46 \\
\text{Tongue Back Z} & 2.87 & 3.18 & 4.48 \\
\text{Jaw X} & 0.83 & 0.83 & 1.12 \\
\text{Jaw Z} & 1.35 & 1.35 & 1.85 \\
\text{Upper Lip X} & 0.58 & 0.6 & 0.74 \\
\text{Upper Lip Z} & 1.21 & 1.26 & 1.54 \\
\text{Lower Lip X} & 0.70 & 0.74 & 0.95 \\
\text{Lower Lip Z} & 1.05 & 1.12 & 1.48 \\
\hline
\end{array}
\]

Table 4.3: RMSD of minimum kinematic cost paths from the real path. Results from 512 code codebook based on data from speaker A are shown

The RMSD values indicate that the minimum kinematic cost paths solutions were slightly less accurate compared to the RMSD of the mean path of the one-to-one codebook of the previous chapter shown in section 3.3.1. The minimum cost path predictions were better than the predictions from the null set (shown in section 3.3.1) or the average predictions from a random path.

Table 4.4 shows the RMSD of the predicted positions of speech articulators from the real positions used by the speaker for a 2048 code codebook. Predicted paths shown minimize the summation of the absolute value of the articulator velocity (minimum distance path), acceleration and jerk. Predictions were performed on speaker As dataset. The performance of the mapping strategies were compared to the performance of the one-to-one codebook recalculated using 2048 codes, and to the average performance of a random path (as explained in section 4.2.5). Both the performance of the one-to-one
codebook, as well as the random path using a 2048 code codebook are shown in the table 4.4.

<table>
<thead>
<tr>
<th></th>
<th>RMSD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>One to One Codebook</td>
</tr>
<tr>
<td>Tongue Tip X</td>
<td>3.07</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>2.55</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>2.69</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>3.63</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>2.88</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>2.93</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.73</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>1.20</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.48</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>1.11</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.65</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>0.96</td>
</tr>
</tbody>
</table>

Table 4.4: RMSD of minimum kinematic cost paths from the real path. Results from 2048 code codebook made on data from speaker A is shown.

The RMSD from the minimum kinematic cost paths were slightly higher than that from the one-to-one codebook. The minimum cost path predictions were better than the average predictions from a random path.

### 4.3.2 Minimum Cost Path With Test Dataset Included in the Training Dataset

This section shows the performance (in terms of correlation and RMSD values) of the minimum kinematic cost path when the test dataset was included in the training dataset.
(i.e. the second test paradigm in section 4.2.5). These minimum cost paths were found using a 512 code codebook as well as a 2048 code codebook. However, due to computational challenges explained in section 4.2.4, the minimum jerk path was not calculated for the 512 code codebook. The predictions from these minimum cost paths were compared to the predictions from the one-to-one codebook and the null distributions from the previous chapter. Additionally, we compared the predictions to the average predictions from a random path (as described in section 4.2.5). Since the predictions from the one-to-one codebook as well as the average predictions from a random path may change when we include the test dataset in the training dataset, these prediction values were recalculated in this section. Predictions were performed on speaker As dataset.

Correlation

Table 4.5 shows the correlation between the predicted positions of speech articulators with their real positions included in the training set for a 512 code codebook. Predicted paths shown minimize the summation of the absolute value of the articulator velocity (minimum distance path) and absolute value of the articulator acceleration.
Table 4.5: Correlation of minimum kinematic cost paths with the real path. The real path taken by the speaker is included in the set of possible solutions. Results from 512 code codebook based on data from speaker A are shown.

<table>
<thead>
<tr>
<th>Articulator</th>
<th>One to One Codebook</th>
<th>Minimum Summation Magnitude Velocity Path</th>
<th>Minimum Summation Magnitude Acceleration Path</th>
<th>Random Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tongue Tip X</td>
<td>0.67</td>
<td>0.97</td>
<td>0.99</td>
<td>0.39</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>0.61</td>
<td>0.98</td>
<td>1</td>
<td>0.36</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>0.73</td>
<td>0.97</td>
<td>1</td>
<td>0.49</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>0.66</td>
<td>0.98</td>
<td>1</td>
<td>0.41</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>0.73</td>
<td>0.97</td>
<td>0.99</td>
<td>0.48</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>0.69</td>
<td>0.99</td>
<td>1</td>
<td>0.46</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.74</td>
<td>0.97</td>
<td>0.99</td>
<td>0.45</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>0.75</td>
<td>0.97</td>
<td>0.99</td>
<td>0.49</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.60</td>
<td>0.96</td>
<td>0.99</td>
<td>0.30</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>0.56</td>
<td>0.97</td>
<td>0.99</td>
<td>0.20</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.65</td>
<td>0.97</td>
<td>0.99</td>
<td>0.40</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>0.59</td>
<td>0.97</td>
<td>0.99</td>
<td>0.26</td>
</tr>
</tbody>
</table>

The correlation results for the 512 code codebook shows that the kinematic path was predicted with near perfect accuracy. Both the path that minimizes distance and the path that minimizes the summation of magnitude of acceleration are very close to the real kinematic path taken by the speaker.

Table 4.6 shows how well the predicted positions of speech articulators were correlated with the real positions for a 2048 code codebook. Predicted paths shown minimize the summation of the absolute value of the articulator velocity (minimum distance path), acceleration and jerk. These predictions were compared to predictions from a one-to-one codebook also shown in the table and the average predictions from a random path.
Chapter 4. Acoustics To Kinematics - Part B

We see that, in addition to the minimum distance and summation of magnitude of acceleration paths, the path that minimizes the summation of magnitude of jerk was also very closely correlated with the real kinematic path. In contrast, the path predicted by the one-to-one codebook showed less accurate prediction performance.

RMSD

Table 4.6 shows the RMSD of the predicted kinematic path from the real path. Predicted paths shown minimize the summation of the absolute value of the articulator velocity (minimum distance path) and absolute value of the articulator acceleration. Predictions were again performed on speaker A’s dataset using a 512 codes one-to-many codebook.
### Table 4.7: RMSD of minimum kinematic cost paths from the real path. Results from 512 code codebook based on data from speaker A are shown

<table>
<thead>
<tr>
<th>Articulator</th>
<th>Minimum summation Magnitude Velocity Path</th>
<th>Minimum summation Magnitude Acceleration Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tongue Tip X</td>
<td>0.93</td>
<td>0.05</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>0.63</td>
<td>0.02</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>0.86</td>
<td>0.02</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>0.78</td>
<td>0.15</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>0.95</td>
<td>0.03</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>0.57</td>
<td>0.13</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.24</td>
<td>0.00</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>0.37</td>
<td>0.01</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.14</td>
<td>0.00</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>0.24</td>
<td>0.01</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.20</td>
<td>0.01</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>0.24</td>
<td>0.01</td>
</tr>
</tbody>
</table>

The kinematics from the paths that minimize kinematic cost show very low RMSD values for the comparison between predicted and real kinematic path that the speaker used.

Table 4.8 shows the RMSD of the predicted positions of speech articulators from the real positions used by the speaker using a 2048 code codebook. Predicted paths shown minimize the summation of the absolute value of the articulator velocity (minimum distance path), acceleration and jerk. These predictions were compared to predictions from a one-to-one codebook also shown in the table and the predictions from a random path solution.
### Table 4.8: RMSD of minimum kinematic cost paths from the real path. Results from 2048 code codebook based on data from speaker A are shown.

<table>
<thead>
<tr>
<th></th>
<th>One to one codebook</th>
<th>Minimum Summation Magnitude</th>
<th>Minimum Summation Magnitude</th>
<th>Minimum Summation Magnitude</th>
<th>Random Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tongue Tip X</td>
<td>2.92</td>
<td>0.24</td>
<td>0.13</td>
<td>0.13</td>
<td>4.55</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>2.43</td>
<td>0.27</td>
<td>0.17</td>
<td>0.13</td>
<td>3.68</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>2.56</td>
<td>0.28</td>
<td>0.16</td>
<td>0.16</td>
<td>4.00</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>3.45</td>
<td>0.31</td>
<td>0.25</td>
<td>0.24</td>
<td>5.12</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>2.74</td>
<td>0.33</td>
<td>0.18</td>
<td>0.18</td>
<td>4.3</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>2.78</td>
<td>0.29</td>
<td>0.22</td>
<td>0.22</td>
<td>4.3</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.69</td>
<td>0.16</td>
<td>0.09</td>
<td>0.09</td>
<td>1.11</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>1.14</td>
<td>0.20</td>
<td>0.16</td>
<td>0.15</td>
<td>1.84</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.46</td>
<td>0.07</td>
<td>0.04</td>
<td>0.04</td>
<td>0.65</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>1.06</td>
<td>0.10</td>
<td>0.06</td>
<td>0.06</td>
<td>1.47</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.61</td>
<td>0.07</td>
<td>0.04</td>
<td>0.05</td>
<td>0.89</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>0.91</td>
<td>0.09</td>
<td>0.05</td>
<td>0.06</td>
<td>1.45</td>
</tr>
</tbody>
</table>

In addition to the minimum distance and acceleration paths, we see that the path that minimizes jerk also showed very low RMSD from the real path.

# 4.4 Comparisons

## 4.4.1 Minimum Kinematic Cost Path Compared to a Random Path

We see from section 4.3.1 that the minimum kinematic cost paths consistently show better average prediction performance than the prediction performance from a random path (defined in section 4.2.5). In order to statistically compare these performances, we
determined the probability that a random path could perform better than the minimum summation magnitude velocity path (i.e., minimum distance path). We calculated this probability for multiple continuous acoustics, and table 4.9 shows the average of these probabilities. Data is presented for a 2048 code book for which the test dataset was not included in the training dataset and 10000 random paths were found for each continuous acoustic.

<table>
<thead>
<tr>
<th></th>
<th>Average Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tongue Tip X</td>
<td>0.05</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>0.07</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>0.03</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>0.04</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>0.03</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>0.04</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.09</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>0.09</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.16</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>0.12</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.13</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>0.10</td>
</tr>
</tbody>
</table>

Table 4.9: Average probability of that a random kinematic path found for a continuous acoustic shows equal to or better prediction (RMSD) than the minimum distance path.

We see from this table there is a small probability that a randomly selected kinematic path shows better kinematic prediction that the minimum distance path for most speech articulators. However, the difference between predictions using randomly selected path and the minimum distance path is smaller for the movements of the lip, in particular the movement of the upper lip in the X (front-back) direction. This seems to be consistent with our finding earlier in section 3.2.5 where we observed that the upper lip positions in the X direction are not clustered as well as the positions of other articulators. This may be because the movement of the upper lip is small in the X direction and the effects of these movements are not reflected in the acoustic output. Hence a codebook may not capture the front-back movements of the upper lip as well as it does the movement of all other speech articulators.
4.4.2 Cost of the Real Kinematic Path Compared to a Random Path

This chapter presents a codebook mapping technique where we select the path of minimum kinematic cost as the predicted path. This technique is based on the hypothesis that the real kinematics chosen by the speaker (i.e., the real kinematic path) is the path that minimizes some kinematic cost. Table 4.10 shows the probability that a randomly selected kinematic path has a kinematic cost (calculated as the summation magnitude velocity for an articulator) lower than the kinematic cost of the real kinematic path. Data is presented for a 2048 code codebook for which the test dataset was not included in the training dataset and 10000 random paths were found for each continuous acoustic.

<table>
<thead>
<tr>
<th>Articulator</th>
<th>Average Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tongue Tip X</td>
<td>0.01</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>0.01</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>0.01</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>0.01</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>0.01</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>0.01</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.01</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>0.01</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.02</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>0.02</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.01</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Table 4.10: Average probability that a randomly selected path shows equal to or less movement for a particular articulator than the real kinematic path

This table shows that more than 98% of randomly selected paths show more movement of speech articulators than the movement of speech articulators in the real path chosen by the speaker.
4.4.3 Comparison between Minimum Distance, Minimum Acceleration and Minimum Jerk

We see from the findings presented here that when the test dataset was not included in the training dataset (i.e. test paradigm one described in section 4.2.5), the minimum distance path showed better predictions (on average) than the minimum acceleration path; which in turn showed better prediction (on average) than the minimum jerk path. On the other hand, we see that when the test dataset was included in the training dataset (i.e., test paradigm two described in section 4.2.5), the minimum distance path showed poorer predictions than the minimum acceleration and minimum jerk paths. In this subsection we determine if these average differences in prediction are statistically significant.

Table 4.11 shows whether differences in predictions between mapping strategies for the first test paradigm are statistically significant. Shown in the table are p-values calculated from paired t-tests of prediction performance (RMSD) for continuous acoustics using different selection strategies. 2048 code codebooks were used to perform these comparisons, and predictions for continuous acoustics (of which there are about 200 per test set) from 10 test sets (in the 10 fold cross validation analysis) are averaged and then compared. Thus the paired t-test has 9 degrees of freedom.
Table 4.11: P-values and corresponding t-values, showing the probability that two selection strategies show similar prediction performance.

We see from this table that the predictions from using the minimum distance strategy were significantly different from the ones using either the minimum acceleration strategy or the minimum jerk strategy. The exception to this are the predictions of the upper and lower lip positions where these differences in most cases were found not to be significant.

The differences in predictions using the minimum acceleration and minimum jerk paths were significant for the positions of the tongue, except for the position for vertical positions of the tongue tip and tongue back. These strategies do not show a significant difference in the prediction of all other speech articulators.
For the second test paradigm, i.e., when the test dataset is included in the training dataset, we find that the predictions from the minimum distance path were significantly different (with a p-value less than 1% for all speech articulators) from the predictions based on the minimum acceleration and minimum jerk paths. However, the difference between the minimum acceleration and minimum jerk paths was not significant for any speech articulator. These p-value results are found using a paired t-test with 2098 degrees of freedom.

The implications of the significant improvement in the first test paradigm by using the minimum distance path, and the significantly poorer performance of this path in the second test paradigm are discussed in chapter 5.

4.5 Sources Of Error

The one-to-many codebook addresses the inherent sources of error that the one-to-one codebook in the previous chapter suffers from, which is that an acoustic may be produced by different kinematic configurations which are not well represented by an average of all these kinematic configurations (as discussed in section 3.5).

However, since the codebook in this chapter was based upon the same dataset and on the same vector quantization technique as the codebook in the previous chapter, it shares in principle the same procedural problems as the one-to-one codebook. These sources of error could arise due to inherent time lag between acoustics and kinematics, non-stationary vocal tract functions, improper silence removal, suboptimal representation of acoustics, error due to insufficient data, error from the k-means algorithm and error due to quantization in the acoustic domain. These errors were described in section 3.5.

Of the sources of error mentioned above, one that may be particularly relevant to the current chapter is error due to insufficient data. In order for the minimum kinematic cost path to successfully predict the real kinematic path, the real kinematic path needs to be present in the training dataset. That means that there needs to be a reference in the training dataset where the acoustics were produced with exactly the same kinematic configuration as the acoustics in the test dataset. Results in this chapter show that the prediction performance dramatically increases when the real kinematic path from the
test dataset was included in the training dataset. This implies that for the other analysis presented in chapter 4, the training dataset often did not contain the kinematic path that was used by the speaker(s) for the test set.

We suggest two possible solutions to overcome the issue of insufficient data. The first one is to modify data processing to capture more datapoints from the raw data of the speaker when creating the dataset. Kinematics are captured with a frequency of 250 Hz by the EMA AG501 machine. When processing the dataset we window data in 30ms (33Hz) segments as speech articulators are theorized to be quasi-stationary within such durations. However, we notice in our analysis in section D of the appendix that the speech articulators may show significant movement in this 30ms interval. Hence, by processing kinematic data at 33Hz instead of the 250Hz available to us, we may lose critical information on certain kinematic configurations. In order to capture this data, we can either use shorter windows (e.g. 10ms windows) or use overlapping windows.

The second solution is to interpolate new points in the dataset. Atal presents a theoretical method [48] where all regions in articulatory space which map on the same acoustic value can be inferred from the dataset. Our attempt to include this method produced inconclusive results. This method and its performance is discussed in the appendix section J. A number of reasons for the poor performance of this method are also discussed in the appendix, with the primary reason being that we may not have access to sufficient data points in our dataset in order to perform this interpolation.

4.6 Discussion

Results from this study show that when the real solution for articulatory inversion existed in the dataset, the minimum cost strategies were able to select these solutions with very little error. The minimum acceleration and minimum jerk paths showed comparable performance. The minimum acceleration and jerk paths outperformed the minimum distance path. However, the minimum distance path predictions also had low RMS error (under 1mm) which is comparable to the EMA501 machine error. It should be noted that the minimum cost strategy was given no information about the presence, location or value of the real solution, i.e. the technique was not biased towards the real kinematic solution. This performance is significantly improved from the performance of the
method by Hogden and colleagues \[21\] used on our dataset in Chapter 3. Additionally, this performance is also significantly better than the performance of the linear regression method shown in section 3.4.3. The predictions of the minimum acceleration and minimum kinematic path showed RMS errors under 0.25mm which is better than prediction values reported from any other method of articulatory inversion.

When the training and test dataset were completely distinct, the findings show that the minimum cost strategies show comparable prediction performance to the one-to-one codebook from the previous chapter. However, we see in section 4.4 that the minimum distance path showed better articulatory inversion predictions than 90-95% of randomly selected paths from the codebook. Hence, even in the possible absence of the real solution in the codebook, we see that choosing the minimum kinematic cost path remains a strategy that shows good prediction performance. When the real path was not included in the solutions set, we see that the minimum distance path outperformed the minimum acceleration and jerk paths. The minimum acceleration path and minimum jerk path do not show any significant differences.

In this chapter, we presented three strategies based on minimization of distance, acceleration and jerk. Different strategies showed better performance for when the test dataset was included versus when the test dataset is not included in the training set. Chapter 5 provides a theoretical explanation for this observation. The prediction techniques presented in this chapter attempt to reflect the underlying speech production strategies used by a speaker. Chapter 5 also examines the implications of these findings, as well as findings from Chapter 3, for speech science theory.
Chapter 5

Discussion and Future Work

In this study, we explore techniques for articulatory inversion, i.e., recovery of positions of speech articulators from speech acoustics. These techniques are analyzed using a relatively large and varied speech sample together with kinematic data derived from a participant speaking in a natural manner. Our goal is not only to find an effective technique for articulatory inversion, but to also use this technique to provide a better understanding of the speech planning and production process. Hence, in addition to looking at the accuracy of articulatory inversion, we also aim to understand the key principles and assumptions behind the techniques.

Motivation for a Current Study

Broadly speaking, the two questions relevant to articulately inversion are, how do acoustic units map to kinematic configurations, and why are certain kinematic configurations chosen to produce certain acoustic units. The first (how) question can be understood as, how can we arrive at a set of kinematic configurations that can produce a given acoustic unit? The second (why) question can be understood as, why do we select a certain kinematic configuration when many other kinematic configurations can produce the same acoustic? This work differentiates itself from other works by aiming to answer the (why) question over the (how) question.
Other articulatory inversion techniques like Multiple Linear Regression, Artificial Neural Networks and Gaussian Mixture Models aim to answer both the *how* and the *why* questions using a unified approach. This is done by either relying on prior assumptions about the acoustic-kinematic relationship or by using computational techniques for articulatory inversion which may offer little insight into understanding this relationship. We believe that this may not be a correct approach as these *how* and *why* questions may be based on very different underlying principles. Since acoustics are a result of speech articulator positions and movements, the question of how acoustics map to kinematics may be better understood by analyzing the reverse question of how do kinematics map to acoustics. On the other hand, the answer to the question of why we choose certain kinematic configurations may have its basis in motor control strategies used during speaking. In our current work, we answer the *how* and *why* questions separately. We believe that the question of how acoustics map to kinematic configurations is better understood using forward mapping models [31] [31] [16] [21], and hence we answer this question only insofar as it helps us answer the *why* question. In order to find which kinematic configurations an acoustic value may map to, we take a utilitarian approach and find the kinematic configurations that have been shown to produce similar acoustics. This technique, presented in Chapter 3, presents an easy to understand, computationally efficient, and low-overhead technique to perform articulatory inversion based on empirical data. Hence, Chapter 3 presents a solution for the *how* questions, and Chapter 4 build upon the methods in Chapter 3 to answer the *why* question.

**Results from Chapter 3 and Their Implications**

Chapter 3 shows that the knowledge of how a short segment of acoustic was produced in the past helps us to predict how that acoustic would be produced in the future. Furthermore it shows that a simple clustering of the acoustic space into a set number of discrete codes may be sufficient to capture the acoustic variance in naturally spoken speech. We show this by analyzing the performance of a one-to-one acoustic to kinematic vector quantization codebook as proposed by Hogden et al. [21] on naturally spoken speech. This codebook maps short segments of acoustic data (30ms) to the average configuration of kinematics that has been found to produce similar acoustic patterns in previous samples. Results from Chapter 3 show that the vector quantization clustering method presented by Hogden et al. [21] can be used to perform articulatory inversion for a naturally spoken phonemically diverse dataset. Prediction results shown in this chapter
are comparable to prediction results shown in the past using this same technique on a more restricted dataset \cite{21}, as well as the results shown by other articulatory inversion techniques like linear regression.

We can draw three main conclusions from the fact that the clustering codebook method shows strong results for articulatory inversion for both restricted and more varied datasets. The first is that knowledge of how similar acoustics were produced in previous samples can help inform our predictions of how an acoustic signal may be produced in the future. This shows that even if the acoustic-kinematic relationship is complex, it is still predictable and follows certain regularities in the mapping between the two dimensions. The second conclusion we can draw from the clustered codebook method is that the average kinematic configuration used to produce similar acoustics can be used as a solution for articulatory inversion. Analyzing this one-to-one codebook reveals that on occasions, very different kinematic configurations may produce the same acoustic output (as shown in section \cite{3.5}). This indicates the presence of motor equivalence in speech. Given motor equivalence, the different kinematic configurations that can produce very similar acoustic results may not be well represented by taking an average of these kinematic configurations. Despite these shortcomings related to the presence of motor equivalence, using the average the kinematic configuration shows good articulatory inversion prediction performance. This raises the question of why an averaged kinematic configuration is, although not perfect, still a very good approximation of the actual kinematic patterns used by a given speaker. Surely, it has to reflect some basic property of the acoustic-kinematic relationship. We speculate that the good prediction results found using the averaged kinematic configuration may be a result of some statistical optimization that is represented by an averaged composite. Despite the fact that the average kinematic mapping used here may represent a suboptimal prediction solution, the Hogden method of clustering acoustics into a discrete number of configurations produces good prediction results when compared to other articulatory inversion methods. This brings us the third conclusion that we draw from Chapter 3, which is that a set number of clusters (128, 256 or 512) may be sufficient to capture the variability in acoustics a speaker may produce. This information is beneficial as it provides a computationally efficient method to find acoustic patterns in the training set (past productions) that are similar to an acoustic pattern in the test set (the given acoustic segment).
A New Mapping Technique That Accounts For Motor Equivalence

While the average kinematic configuration shows relatively good prediction results, these results still show significant prediction errors. Additionally, using the average kinematic configuration as the predicted value does not give us an insight into the underlying acoustic-kinematic relationship. We believe that a better prediction result can be obtained if actual kinematic configurations used by the speaker in the training set are used instead. Since many kinematic configurations are used by the speaker in the training set to produce a given acoustic, this method creates a certain ambiguity in how to select the best kinematic configuration as the predicted value.

Many studies in the past have suggested that speech motor control follows principles of minimization of kinematic costs, and we use these principles to try to address this ambiguity. It has been shown in other works of motor control that human movements may follow a path that reduces kinematic cost. Flash and Hogan [12] showed that arm movements in reaching tasks followed a path of minimum kinematic jerk. It has been suggested that speech articulatory movements may also be influenced by kinematic cost considerations [29]. It was shown by Neufeld and Van Lieshout [36] that the tongue movements in a gesture followed a path closely related to the minimum jerk path. Since kinematic cost considerations may reflect motor control principles, in Chapter 4, we present an articulatory inversion technique that takes kinematic cost into account. Hence, Chapter 4 builds upon the articulatory inversion solution shown in Chapter 3 and incorporates kinematic cost considerations to answer the questions of why certain kinematic configurations are chosen by a speaker when many kinematic configurations are able to produce a given acoustic unit.

Results from Chapter 4 and Their Implications

Chapter 4 shows that by incorporating the knowledge of the larger acoustic unit from which the short acoustic segment was taken, we may be able accurately predict an articulatory inversion solution by taking into account kinematic cost minimization principles. The technique in Chapter 4 suggests that when many different kinematic configurations can produce the same short acoustic segment, the kinematic configuration chosen is the one that reduces kinematic cost as part of the dynamic trajectory used to produce the
larger acoustic unit. Chapter 4 uses the vector quantization codebook method proposed in Chapter 3 to find kinematic configurations that have produced similar short segments of acoustics (30ms) in previous samples. However, instead of using an average of these kinematic configurations, we preserve each kinematic configuration as a valid articulatory inversion solution. Results from this chapter show that there is strong indication that the dynamical path of speech articulators chosen by the speaker to produce a series of acoustic sounds is the path that minimizes kinematic jerk as well as kinematic acceleration.

Results from Chapter 4 show that adding kinematic cost constraints significantly improves the prediction performance of the articulatory inversion technique. This performance using minimal acceleration and jerk paths shows negligible error in prediction performance when the test dataset is included in the training dataset, i.e. when the correct solution to articulatory inversion is guaranteed to exist in the training dataset. This negligible error can likely be attributed to procedural sources of error discussed in section 3.5. Hence the minimal acceleration and jerk paths may reflect the correct solution to articulatory inversion. This may be the first such articulation inversion technique which demonstrates the ability to recover the correct articulatory inversion solution from a set of possible solutions. The predictions in Chapter 4 however suffer when the training dataset is incomplete. We see that when values from the test dataset are not include in the training dataset, the articulatory inversion performance decreases significantly, thus indicating that certain kinematic values used by the speaker in the test dataset are not contained in the training dataset. Hence, further work is needed in order to ensure its completeness.

Significance of a Minimum Kinematic Cost Path

Previous works have suggested many reasons why kinematic cost may be minimized in motor control. Lindblom [29] suggested that minimizing kinematic cost of speech articulator movements would lead to minimization of energy which would offer an evolutionary advantage. A possible counter point to an evolutionary reasoning of cost minimization offered in the same work is that speech articulator movements consist of usually small amplitudes, and hence it can be debated if cost optimization leads to a substantial saving of energy. An argument against this evolutionary perspective (albeit for hand movement motor control), is presented by Flash and Hogan [12] where they show kinematic move-
ments in a joint reference frame (which is more closely related to the musculoskeletal structures) for the arm do not show minimization of kinematic cost. It is suggested by Hogan [20] that minimization of jerk maximizes the predictability of the articulatory trajectory.

Many other works studying the acoustic-kinematic relationship in speech analyze kinematic movements between targets. The selection of these targets is important as it has implications for the reference frame of speech production. In Articulatory Phonology theory [7], speech targets are defined as vocal tract constriction goals which are met using articulatory gestures. Hence, this theory considers speech planning to take place closer to the reference frame of speech articulators. Work by Neufeld and Van Lieshout [36] also defines speech targets as related to the reference frame of speech articulators by defining them as the onset and offset of articulatory gestures. Their work shows that the tongue follows a path of minimum jerk within an articulatory gesture. This cost optimization in the kinematic cost during speech production indicates that speech planning is related to a kinematic reference frame. On the other hand, in the DIVA model [56], speech targets are defined in an auditory perceptual reference frame as defined by Miller in 1989 [32]. In this model, acoustics goals are met using an acoustic feedback mechanism along with a neural network which maps a desired change in acoustics to a kinematic movement. In the current work, we intentionally avoid an explicit theoretically informed classification of acoustics and kinematics into phonological and/or phonetic features. Hence, while in the current study, we use acoustic targets defined in 30ms frames, these targets are not explicitly related to targets as defined by the DIVA model or the Articulatory Phonology theory. The choice of targets as acoustics in 30ms frames does not necessarily reflect a speech planning goal, but should rather be understood as a tool to analyze properties of the acoustic-kinematic relationship in speech.

While the current work does not directly relate to theoretically understood speech goals, the results from this work have important implications for speech science. They show that the kinematic configurations chosen during speech minimize kinematic cost. This result of a minimal kinematic path is not compatible with the acoustic to articulatory movement mapping model included in the DIVA model. The DIVA model suggests that kinematic configurations are chosen based on acoustic feedback which informs the speech planning system of the current acoustic unit being produced, and knowledge of the desired target acoustic. This implies that speech planning is a recursive process where acoustics are steered towards a target acoustic using acoustic feedback which informs
kinematic movements. Hence, according to this model, speech kinematics are an emergent property arising by the speech production system’s effort to achieve the desired acoustic. The results from the current study suggest that speech kinematics are planned while optimizing kinematic cost. An optimization of kinematic cost would not be possible using a recursive feedback mechanism where kinematics are emergent properties. This is because in order to optimize kinematic cost, when planning the current positions of a speech articulator, the speech planning mechanism would need to take into account future positions that speech articulators will take. Hence, minimization of kinematic cost indicates that speech planning reflects kinematic principles and not acoustic ones.

The DIVA model suggests that regularities in kinematics in speech may arise due to a preference for comfortable kinematic positions, which are defined as central positions of speech articulators. In instances where multiple kinematic configurations are able to produce an acoustic, a preference for comfortable positions prevents the speech articulators from assuming unnatural positions during speech production. The DIVA model refers to this property as postural relaxation. While the DIVA model hypothesizes that postural relaxation can account for regularities in speech production, a quantitative analysis of predictions using this method on empirical data is not presented in this model. The current work shows that kinematics in speech can be recovered from speech acoustics without any explicit bias towards certain positions in the acoustic to kinematic mapping. It suggests that regularities in speech kinematics arise from motor control principles.

Motor control principles analyzed in this work may inform our understanding of the origin of speech gestures. We see from this work that speech kinematics (which are made up of speech gestures) can be accurately predicted from speech acoustics. However, it is unclear if these articulatory gestures are recovered as the minimum kinematic cost solutions because they are cost optimal ways of producing an acoustic, or they are recovered because gestures are used to produce acoustics in a cost optimal way. The first possibility would imply that speech is planned in an acoustic reference frame, and gestures are emergent properties because they are kinematically efficient ways of producing the given acoustic. However, this would raise the question of why speech gestures are consistently observed in speech given there is no explicit bias towards them in acoustic-kinematic mapping? This can be rephrased as, why are acoustic targets used in speech such that the cost optimal ways of producing these acoustic targets are gestures? The second possibility would suggest that speech is planned in a kinematic reference frame and the choice of gestures in speech; in particular the positions of articulators at the start and end
of a gesture cannot be understood using acoustic to kinematic mapping. The path taken by the speech articulators between the start and end positions of a gesture is optimized for kinematic cost (as suggested by Neufeld and Van Lieshout [36]) and hence these gestures show low kinematic cost. However, given this explanation, it is surprising that gestures were consistently recovered as cost optimal ways of producing acoustics. This is because, while the above explanation suggests that gestures show low kinematic cost, it does not state that some other kinematic path with lower kinematic cost cannot exist to produce the given acoustic between a different set of target positions. Neufeld and Van Lieshout [36] suggest that the articulators may use a minimum cost path between target positions in a gesture, but this doesn’t explain why this minimum cost path between a given set of target positions also shows lower cost than any other path the speaker can take between a different set of target positions that can produce the same acoustic. We also propose a third possibility, which is that not only is speech planned in terms of gestural goals, but the minimum cost path between target positions is a gesture is also the minimum cost path to produce the given acoustic. This means that in order to produce a set of target acoustics, the speech planning system translates these target acoustics into the corresponding gestural goals. This is done because the minimum cost path between these gestural goals is also the globally minimum cost path to produce the given target acoustics. This suggests a synergy in the acoustic and kinematic domains of speech. Kinematic gestures are chosen because the minimum cost trajectory within the start and end of a gesture is also the minimal cost trajectory to produce the acoustics corresponding to the kinematic gesture.

**Future Work**

The current study can be thought of as a technique for articulatory inversion, as well as a tool to study the relationship between acoustics and kinematics in speech. For applications in articulatory inversion, the technique in the current work suffers when the training dataset is incomplete. Hence, for accurate mapping of acoustics to kinematics, the correct articulatory inversion solution needs to exist in the training dataset. Future works can explore techniques to complete the training dataset.

As stated earlier, this work analyzed the properties of the acoustic-kinematic relationship, but does not explicitly relate to theoretically understood speech planning mechanisms. We suggest that simple modifications to the techniques in our work can be used to
answer these questions related to speech planning and motor control in speech. These modifications mainly consist of including phonological and phonetic features into our dataset. Below we analyze a few questions that future works may wish to answer using the techniques presented in this work.

Previous work by Neufeld and Van Lieshout \[36\] has suggested that the kinematic path between target positions in a gesture follows a path of minimization of jerk. Our work shows that the kinematic path over multiple gestures is optimized for kinematic jerk. Is this minimal kinematic cost path for the continuous acoustic segment in our work the same as a collection of smaller paths which have minimum jerk between target positions in gestures that make up the acoustic segment? This can be understood by comparing the minimum kinematic cost path in this work, with a kinematic cost path that optimizes cost between the start and the end of a gesture.

Gestures in speech are high level descriptions of dynamic goal oriented kinematic movements. Hence the positions of speech articulators at the start and end of a gesture are not defined in terms of their absolute positions. Does the principle of minimization of kinematic cost influence the position of speech articulators at the start and end of a gesture? By this we mean that, are the positions of speech articulators at the start and end of a gesture influenced by the acoustics and kinematics that come both before and after the given gesture. This can be answered by analyzing the changes in the articulator positions at the start and end of a gesture if the gesture was to be analyzed in isolation from the rest of the larger acoustic segment the given gesture is a part of. If there exist a kinematic path of lower kinematic cost between the acoustics corresponding to the start and end of the gesture, than the kinematic cost of the real path used by the speaker, this would imply that the kinematics positions at the start and end of the gesture are influenced by the context in which the gesture was used.

To summarize, the current study shows that speech planning utilizes motor control principle of minimum kinematic cost. It suggests that speech planning takes place in the kinematic domain, as well as that speech kinematics are planned before acoustic production. Results from this study confirm the conclusions of previous studies like Neufeld and Van Lieshout \[36\] by showing speech gestures show minimal kinematic cost. Furthermore, results from this study suggest that speech gestures may exhibit a unique symbiosis with the speech acoustics produced using them, i.e. the speech gestures may be the kinematically optimal way of producing this speech acoustic.
Appendix A

Correlation of Energy in Speech Frequencies and Articulator movements

This appendix chapter shows the correlation of the movements of different speech articulators with energies in different frequency regions in the vocal tract filter function.
Appendix A. Correlation of Energy in Speech Frequencies and Articulator movements

(c) Tongue Dorsum X

(d) Tongue Dorsum Z

(e) Tongue Back X

(f) Tongue Back Z

(g) Jaw X

(h) Jaw Z
Figure A.1: Correlation of positions of speech articulators with power in linearly divided frequency regions for the dataset from speaker A
Appendix B

Correlation Statistics of Articulatory Predictions Using Method in Chapter 3
Appendix B. Correlation Statistics of Articulatory Predictions Using Method in Chapter 3
Appendix B. Correlation Statistics of Articulatory Predictions Using Method in Chapter 3

(e)

(f)
APPENDIX B. CORRELATION STATISTICS OF ARTICULATORY PREDICTIONS USING METHOD IN CHAPTER 3

(g)

(h)
APPENDIX B. CORRELATION STATISTICS OF ARTICULATORY PREDICTIONS USING METHOD IN CHAPTER 3

(i)

(j)
APPENDIX B. CORRELATION STATISTICS OF ARTICULATORY PREDICTIONS USING METHOD IN CHAPTER 3

(k)

(l)
Appendix B. Correlation Statistics of Articulatory Predictions Using Method in Chapter 3
Appendix C

Shortcoming of Using Correlation and RMSD for Measuring Prediction Performance

C.1 shows two variables that have different values but show a Pearson correlation of 1.

Figure C.1: Figure showing two variable that have different amplitude ranges but show strong correspondence in the direction of change between samples resulting in a Person’s correlation of 1.
Appendix C. Shortcoming of Using Correlation and RMSD for Measuring Prediction Performance

Similarly, it should be noted that a Pearson’s correlation of 0 doesn’t necessary mean that the two variables are completely uncorrelated, as they may have a non-linear relationship. Figure C.2 show two variables that have a functional relationship but show a Person correlation of zero.

![Figure C.2: Figure showing two variable X and $X^2$ having a non-linear (quadratic) relationship. The Person’s correlation is 0 when X is symmetric about 0.](image)

A challenge of using RMS deviation is that the RMS value alone may not tell us about the precision of the predictions. In Figure C.3, the RMS deviation (which is same as its nRMS deviation, as A has an amplitude of 1) between variable $A$ and $A1 = A/2$ is 0.3315 whereas the RMS deviation (and nRMS deviation) between $A$ and $B$ is 0.0968. The accuracy of predictions depends on what we aim to measure. If we wish to capture the trends of change in variable $A$, then variable $A1$ is a better predictor of variable $A$ than variable $B$. However the values of rmsd and nrmsd suggest otherwise.
Appendix C. Shortcoming of Using Correlation and RMSD for Measuring Prediction Performance

Figure C.3: RMS deviation between variable $A$ and $A_1 = A/2$ is 0.3315 and the RMS deviation between $A$ and $B$ is 0.0968

This limitation can be overcome when we look at the correlation values along with the RMSD. For Figure C.3, the correlation between $A$ and $A_1$ is 1 whereas the correlation between $A$ and $B$ is 0.94.
Appendix D

Source of Error for Chapter 3

D.1 Time Lag

Figure D.1: Correlation with adding a time delay between acoustics and kinematics. Data for speaker A using a 512 code codebook
Delivering the acoustics with respect to kinematics shows slight improvement for the prediction of the movement of the upper lip in the vertical direction, and the movement of the lower lip in the horizontal and vertical direction. However, contrary to the findings in previous studies, predictions from our codebook are generally poorer with the introduction of a delay.

### D.2 Non Stationary vocal tract transfer function

By processing data in shorter windows, we may make the assumption of stationary kinematics to be more likely correct. The table D.1 shows predictions of articulatory data by using a 10ms window and compares it to the results obtained by using the 30ms window.
<table>
<thead>
<tr>
<th></th>
<th>Original Codebook</th>
<th>Codebook using 10ms windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tongue Tip X</td>
<td>2.29</td>
<td>2.41</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>2.36</td>
<td>2.37</td>
</tr>
<tr>
<td>Tongue Tip Theta</td>
<td>0.24</td>
<td>0.27</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>2.77</td>
<td>2.93</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>3.03</td>
<td>3.16</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>2.33</td>
<td>2.54</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>2.79</td>
<td>2.83</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.47</td>
<td>0.50</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>1.44</td>
<td>1.47</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.69</td>
<td>0.61</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>0.94</td>
<td>0.83</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>1.82</td>
<td>1.94</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>1.62</td>
<td>1.67</td>
</tr>
</tbody>
</table>

Table D.1: Comparison of Prediction between 10ms window and 30ms window. Dataset for speaker C and 512 code codebook.

**D.3 Sub-Optimal representation of acoustics**

We have tried an experiment where an alternate codebook was constructed using acoustic where each frequency was normalized so that all the frequencies in the vocal tract filter function has equal mean variability though the dataset. The performance of this codebook is shown in table D.2 and compared to the performance of the codebook constructed without any normalization.
Appendix D. Source of Error for Chapter 3

<table>
<thead>
<tr>
<th></th>
<th>Original Codebook</th>
<th>Codebook with normalized acoustics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tongue Tip X</td>
<td>3.4</td>
<td>3.79</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>2.79</td>
<td>2.83</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>2.94</td>
<td>3.42</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>4.05</td>
<td>4.46</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>3.14</td>
<td>3.62</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>3.2</td>
<td>3.76</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.79</td>
<td>0.93</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>1.25</td>
<td>1.52</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.54</td>
<td>0.57</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>1.16</td>
<td>1.3</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.68</td>
<td>0.68</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>1</td>
<td>1.13</td>
</tr>
</tbody>
</table>

Table D.2: RMS Error using a codebook constructed with 512 codes on speaker A’s data. The RMSD from predictions using a codebook with normalized frequency is compared to the rmsd from prediction from the original codebook.

D.4 Simultaneous Coding

A possible modification to this technique is to find simultaneous clusters in acoustics and kinematics, instead of clusters only in acoustics. This is done by first normalizing the acoustics and kinematics so that the average variability of the acoustic variables and the kinematic variables is the same. Since we deal with x dimensions of kinematics and 240 dimensions of acoustics, we now perform vector quantization in 240+x dimensions. The RMSD of kinematic predictions from a codebook constructed with simultaneous clustering of acoustics and kinematics is shows in D.3 and is compared to the RMSD from predictions using the original codebook.
### Table D.3: RMSD from predictions using a codebook with simultaneous acoustic and kinematic coding compared to the RMSD from the original codebook. Codebooks constructed using 512 code codebooks on speaker A’s data

<table>
<thead>
<tr>
<th></th>
<th>Original Codebook</th>
<th>Codebook with simultaneous acoustic and kinematic clustering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tongue Tip X</td>
<td>3.41</td>
<td>3.91</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>2.79</td>
<td>2.89</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>2.94</td>
<td>3.48</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>4.05</td>
<td>4.66</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>3.14</td>
<td>3.69</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>3.28</td>
<td>3.91</td>
</tr>
<tr>
<td>Jaw X</td>
<td>0.76</td>
<td>0.91</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>1.25</td>
<td>1.52</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>0.54</td>
<td>0.58</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>1.16</td>
<td>1.32</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>0.68</td>
<td>0.68</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>1.00</td>
<td>1.14</td>
</tr>
</tbody>
</table>
Djikstra’s graph search algorithm

Djikstra’s algorithm is used to find the shortest path between nodes in a graph. Many modifications of the Dijkstra’s algorithm have been made to suit different applications, but in its most common form, this algorithm is used to find the shortest path (or path of minimum cost) between a start node and every other node in an undirected graph.

The algorithm works as follows

1) Mark all nodes of the graph as unvisited and assign them a cost of infinity. Assign the start node a cost of 0.
2) Starting with start node, find the cost of all the nodes connected to this node. If the cost to a node added with the past cost of the current node is less than the cost already assigned to the destination node, update the cost of that node, and save the current node’s name on the destination node.

Once the cost of the nodes connected to the current node are calculated, the current node is marked as visited.
3) One by one, visit each destination node (which hasn’t been visited yet) connected to the current node. Now repeat step two treating each of these destination nodes as a current node.
4) Repeat step 2 and 3 until all nodes except the finish node has been marked as visited.

5) Backtrack from the finish node to the start node by visiting each of the saved node names. This path is the path of minimum cost.

In the current figure, the path is F-C-B-A.
Appendix F

Computational Considerations

This section states the time complexity of the minimum distance, acceleration and jerk solutions. We compare the time complexities of these solutions found using Dijkstra’s algorithm and using an exhaustive search technique. Finally, we compare the affect of the number of codes in a codebook on the computations taken to find the optimal kinematic cost solutions.

The sub-section Minimum Distance Path Time Complexity defines the time complexity for Dijkstra’s algorithm and for the exhaustive search algorithm. We then calculate and compare the time complexity for the minimum distance solution found using these methods. The relative performance of Dijkstra’s algorithm and the exhaustive search algorithm for the minimum acceleration and minimum jerk solution similar to that for the minimum distance solution. Hence the subsections 'Minimum Acceleration Path Time Complexity' and 'Minimum Acceleration Path Time Complexity' only show the time complexity of the solutions using Dijkstra’s algorithm.

F.0.1 Minimum Distance Path Time Complexity

Dijkstra’s algorithm has a time complexity defined by number of steps multiplied by the number of nodes per step squared (where steps are defined as the number of connections needed to go from a starting node to an end node in the graph) \[10\]. This is because
during Dijkstra’s algorithm, we find the cost of each node in a step to each node in the subsequent step, and this process is repeated as many times as there are steps in the graph.

The graph for the minimum distance Dijkstra solution is shown in figure 4.6. This graph has as one less step than the number of frames in a continuous acoustic (n), and has as many nodes in a step as there are kinematic values in a kinematic vector set (m). The minimum distance Dijkstra solution therefore has a time complexity given by \((n - 1) * m^2\).

The exhaustive search requires finding the cost of every path in the graph. The same graph can be used to find the exhaustive search solution as the one used to find the Dijkstra’s solution (shown in figure 4.6). Since we are required to select each combination of these m nodes in each of the n-1 steps, the time complexity is given as \(m^{n-1}\). Hence Dijkstra’s algorithm is faster than an exhaustive search by a factor of \(m^{n-3}/(n - 1)\).

With a training dataset containing roughly 60000 data points (shown in section 2.5.5), an average kinematic vector set for a 512 code codebook contains roughly 100 possible solutions (≈ 60000/512). On average a continuous acoustic contains about 30 frames. An exhaustive search contains 100^{29} possible paths that need to be searched. Dijkstra’s algorithm requires 29*10000 computations. Hence the exhaustive search solution requires about 100^{25} more computations than the Dijkstra’s search solution.

**F.0.2 Minimum Acceleration Path Time Complexity**

The graph for the minimum acceleration Dijkstra solution is shown in figure 4.7. This graph has as two less steps than the number of frames in a continuous acoustic (n), and has as many nodes in a step as the squared of the number of kinematic values in a kinematic vector set (m). The minimum distance Dijkstra solution therefore has a time complexity given by \((n - 2) * m^4\).
F.0.3 Minimum Jerk Path Time Complexity

The graph for the minimum jerk Dijkstra solution is shown in figure 4.8. This graph has as three less steps than the number of frames in a continuous acoustic (n), and has as many nodes in a step as the cubed of the number of kinematic values in a kinematic vector set (m). The minimum distance Dijkstra solution therefore has a time complexity given by \((n - 3) * m^6\).

F.0.4 Time Complexity as a Function of Number of Codes

The vector quantization algorithm clusters datapoints in the training dataset such that there are roughly the same number of datapoints in each cluster. For a training dataset with \(V\) datapoints, the number of values in the cluster (which is the same as the number of values in the kinematic reference vector set) is on average given by \(m_N = V/N\) where \(m\) is the number of values in a cluster, and \(N\) is the number of codes. Hence the number of possible kinematic solutions in a kinematic reference vector set decreases with increasing number of codebook codes.

This decrease in the number of solutions in kinematic reference vector set by increasing number of codes causes significant decrease in computation time of the Dijkstra solution. This is particularity apparent in the calculation of the minimum jerk solution. The minimum jerk path has a time complexity of \(O((n - 3) * m^6)\), where \(n\) is the number of acoustic frames and \(m\) is the number of kinematic values in a kinematic reference vector set. For a training dataset with \(V\) number of data points, \(m_{512}\) is calculated as \(V/512\) for a 512 code codebook, and the time complexity for the minimum jerk solution is hence \(O((n - 3) * V/512^6)\). This value represents a very large computational complexity. For a 2048 code codebook, \(m_{2048}\) is calculated as \(V/2048\), bringing the time complexity down to \(O((n - 3) * V/2048^6)\). This complexity is smaller that for a 512 code codebook by a factor of 4096 (= 2048^6/512^6).
Appendix G

Kinematic Costs comparison

We analyze kinematic costs of kinematic paths that correspond to acoustics between silences in speech. We compare these kinematic path costs to the kinematic costs of the real kinematic path taken by the speaker.

We find all continuous acoustics (acoustics between silences) in the test dataset described in section 4.2.1. For each continuous acoustic, we find 10 kinematic paths by random sampling.

Figure G.1 shows a scatter plot of possible kinematic path with the average movement of articulators in the path on the x-axis, and the distance of the path from the real path chosen by the speaker on the y-axis.
Appendix G. Kinematic Costs comparison

Figure G.1: Figure showing the distance of the path from the real kinematic configurations used plotted against the distance traveled by articulators across frames for that path. The distance traveled by articulators of the real kinematic path chosen by the speaker is 0.24

Similar behavior is shown when we examine the acceleration and jerk of articulators in possible paths. Figure G.2 shows the average acceleration, and G.3 shows average jerk of articulators in possible paths plotted against the distance of the paths from the real path chosen by the speaker.
Figure G.2: Figure showing the distance of the path from the real kinematic configurations used plotted against the acceleration by articulators across frames for that path. The acceleration of the real kinematic path chosen by the speaker is 0.393
Appendix G. Kinematic Costs comparison

Figure G.3: Figure showing the distance of the path from the real kinematic configurations used plotted against the jerk of articulators across frames for that path. The jerk of real kinematic path chosen by the speaker is 0.7

The figures G.1, G.2, and G.3 show that the real kinematic path may optimize for minimizing a movement cost for speech articulators.
Appendix H

Kinematic Cost Histogram - Comparison with real kinematic path

Figure H.1 shows the distribution of the kinematic cost (calculated as minimum distance) of random paths selected from the codebook for one continuous acoustic. The kinematic cost of the real kinematic path is 0.10. If we are to model the kinematic costs of random paths at a normal distribution, we get a distribution centered at 0.57 with a variance of 0.035. The probability a random path has the cost equal to or less than the real kinematic path (i.e. the p-value) is $7.8 \times 10^{-42}$. 
Figure H.1: The kinematic cost plotted against percentage of paths from the codebook with that cost. Figure plotted for a 512 code codebook for a continuous acoustic from speaker A’s data.

The figure H.1 shows the distribution of kinematic costs for only one continuous acoustic. We repeat this experiment for all continuous acoustics in the dataset of speaker A and calculate the p-value for each. This is shown in the figure H.2.
Figure H.2: The p-value of the kinematic costs (total distance) of the real kinematic path plotted. Figure plotted for a 512 code codebook for a continuous acoustic from speaker A’s data.
Appendix I

Kinematic Cost Histogram - Comparison with closest to real kinematic path

We repeat the analysis presented in [H] but for the kinematic path that is closest to the real path. The figure 1.1 shows the distribution of p-values. The p-values represent the probability of finding a path with kinematic cost less than or equal to the kinematic cost of the closest kinematic path available in the dataset. The mean p-value is 0.004, which means that on average, 0.4% of kinematics paths have lower cost than the path closest to the real path.
Figure I.1: The p-value of the kinematic costs (total distance) of the real kinematic path plotted. Figure plotted for a 512 code codebook for a continuous acoustic from speaker A’s data.
Appendix J

Completing the Kinematic Set

In the previous section, we posed the question that how do we identify all the kinematics the produce the same acoustic i.e how do we complete the kinematic set. In this section we will attempt to answer this question.

Methods presented by [4] show a mathematical way in which we can find all the kinematic values that map to a given acoustic. This technique relies on the assumption that the kinematic to acoustic mapping is linear for small perturbations in the kinematic domain.

Let’s assume kinematic \( x_0 \) and an acoustic \( y_0 \) where \( y_0 \) is produced by kinematic \( x_0 \). \( y_0 = f(x_0) \) where \( y_0 \) is some functional mapping between the kinematic and acoustic domain. We then can say that

\[
y \approx y_0 + B \ast (x - x_0)
\]  
(J.1)

or

\[
\Delta y \approx B \ast \Delta x
\]  
(J.2)

where \( x \) represents a small deviation from \( x_0 \) and \( B \) is a matrix of martial derivatives.
Appendix J. Completing the Kinematic Set

Hence we are approximating the kinematic to acoustic function as a linear one in the vicinity of a point $x_0$. The $i^{th}$ row and $j^{th}$ column of B is given as

$$b_{ij} = \frac{\partial f_i}{\partial x_j}, i = 1, 2, ..., n; j = 1, 2, ..., m \quad (J.3)$$

Here $n$ is the dimensions of $y$ i.e the acoustics and $m$ is the dimensions of $x$ i.e the kinematics.

These partial derivatives are calculated numerically for the dataset.

Since we assume that the mapping between kinematics and acoustics is linear in the vicinity of $x_0$, then

$$\partial y_i = b_{i1} \partial x_1 + b_{i2} \partial x_2 + \ldots + b_{im} \partial x_m \quad (J.4)$$

If we had a value in the dataset where the kinematics move only in the direction $\partial x_j$ from $x_0$, when the change in acoustic for this caused due to this change would give use the value of $b_{ij}$.

$$\frac{\partial y_i}{\partial x_j} = b_{ij} \quad (J.5)$$

However it is highly unlikely that such a kinematic value exists in the dataset that different from kinematic value $x_0$ only in the $x_j$ direction. However if a kinematic exists in the dataset that differs from kinematic $x_0$ in a direction $v = < v_1, v_2, v_3, ..., v_m >$ and causes an acoustic $y_v$

$$\nabla_v y_i = \nabla y_i.v \quad (J.6)$$

Which leads us to the linear equation.
If we are able to find $m$ such vectors like $v$ which are linearly independent, then for each acoustic dimension $i$ we have $m$ equations and can thus solve for the $m$ unknowns $b_{i1}, b_{i2}, b_{i3} \ldots b_{im}$.

For more articulatory dimensions than acoustic dimensions i.e $m \leq n$, then the solution for $\Delta x$ for a value of $\Delta y$ exists in $m-n$ dimensions. For a $\Delta y = 0$ a corresponding solutions to $\Delta x$ represents all the small perturbations from $x$ that cause no change in the output acoustics. These $\Delta x$ are the null space of matrix $B$. These $\Delta x$ values are defined by the equation

$$\Delta x = a_1 \ast v_{n+1} + a_2 \ast v_{n+2} + a_3 \ast v_{n+3} \ldots a_{m-n} \ast v_m$$  \hspace{1cm} (J.8)

Where $a_1, a_2, a_3 \ldots a_{m-n}$ are arbitrary constants, and $v_{n+1}, v_{n+2} \ldots v_m$ represent eigenvectors to square matrix $B^tB$ which have eigenvalues 0. The $\Delta x$ which are a linear combinations of eigenvectors of $B^tB$ with eigenvalues 0, also span the null space of matrix $B$ \[53\].

Regions in articulatory space where movement of articulators ($x$ vectors) cause no change in acoustics ($y$ vector) are called fibers \[23\]. A fiber containing $x_0$ can be found by traveling in a direction $\Delta x$ as defined in equation \[J.8\] away from $x_0$. Since the assumption of linearity of mapping between $x$ and $y$ is likely only true for small perturbations, $\Delta x$ represents a tangent to the fiber. A point $\tilde{x}$ defined as

$$\tilde{x} = x_0 + \epsilon \ast \Delta x$$  \hspace{1cm} (J.9)

where $\Delta x$ is as defined in \[J.8\] and $\epsilon$ represents a small perturbation.

Other points along the fiber can be found by re-evaluating the mapping between $x$ and $y$ around $\tilde{x}$ and finding the null space around point $\tilde{x}$. 

\[y_{v_{i}} - y_{0_{i}} = v_{1} \ast b_{i1} + v_{2} \ast b_{i2} + v_{3} \ast b_{i3} \ldots v_{m} \ast b_{im}\]  \hspace{1cm} (J.7)
This process of finding fibers in the articulatory domain can be repeated for each of the x’s (articulatory configurations) in dataset that map to the same y (vocal tract transfer function). Hence for each y value, we have multiple articulatory fibers, with each point of the fiber representing a possible solution to the articulatory inversion of that acoustic value.

The process presented above in theory would allow us to answer the question, what are all the kinematics that map on the a given acoustic. However implementation on this solution has presented unconvincing results. An experiment is presented here where a matrix of partial derivatives is found and is used to predict the acoustics resulting in a small deviation to a known kinematic (where this kinematic value is not used to construct the matrix of partial derivatives). The methods for calculating the partial derivatives is the same as presented in the previous paragraph. Figure J.2 shows the difference between the known acoustic value, and the predicted acoustic value.

Figure J.1: Figure showing the original acoustic and the predicted acoustic

There may be few reasons that the prediction from the above method is
a) The region over which linearity is assumed is too big
b) The kinematic values are not static within a frame
c) The kinematic values give an incomplete description of the vocal tract.
While point a) could be true, we will consider an example to show that this may not be the only reason. The figure below shows two different kinematic configurations found from the dataset which are very similar. The difference in values between these two configurations is given in the table below. Also shown are the acoustic values, which show a substantial difference. The source filter model predicts some functional relationship from kinematics to acoustics. However the figure below suggests that the same (or very similar) kinematic configuration can produce very different acoustics.

Point b) suggests that the kinematic values are not static in the 30ms regions and therefore the acoustic transfer function doesn’t represent the configuration of a static vocal tract. It has been shown in literature \cite{33} that the speech articulators move with most of the power in the 0-15Hz frequency region. We window the kinematic signal forever 30ms which corresponds to 33.33Hz. The maximum movements of kinematics from the dataset are shown in table J.1.
<table>
<thead>
<tr>
<th></th>
<th>Max Range of Movement (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tongue Tip X</td>
<td>7.94</td>
</tr>
<tr>
<td>Tongue Tip Z</td>
<td>6.33</td>
</tr>
<tr>
<td>Tongue Dorsum X</td>
<td>7.3</td>
</tr>
<tr>
<td>Tongue Dorsum Z</td>
<td>8.02</td>
</tr>
<tr>
<td>Tongue Back X</td>
<td>7.67</td>
</tr>
<tr>
<td>Tongue Back Z</td>
<td>6.71</td>
</tr>
<tr>
<td>Jaw X</td>
<td>2.27</td>
</tr>
<tr>
<td>Jaw Z</td>
<td>3.6</td>
</tr>
<tr>
<td>Upper Lip X</td>
<td>1.5</td>
</tr>
<tr>
<td>Upper Lip Z</td>
<td>3.2</td>
</tr>
<tr>
<td>Lower Lip X</td>
<td>1.6</td>
</tr>
<tr>
<td>Lower Lip Z</td>
<td>3.09</td>
</tr>
</tbody>
</table>

Table J.1: Maximum kinematic movement seen in a 30ms window for dataset from speaker A

It is therefore possible for one kinematic configuration to have the tongue at a static position x and another kinematic configuration to have the tongue moving from x-y/2 to x+y/2 in 30ms. However when averaged over 30ms, both these kinematic configurations look the same.

The last option presented is point c), that the kinematics captured by the EMA system produce and incomplete description of the vocal tract. Research has shows that the shape of the tongue [54] and the position of the velum [14], among other factors also contribute to the vocal tract transfer function. Additionally, while the independent source-filter model assumes the glottal source to be independent of the vocal tract filter function, studies have shown this may not to be completely true. There additional parameters are not captured in the current experiment. While model of the vocal tract do exists that parametrize the vocal tract shape by the positions of the kinematics, some widely used model use the constriction of the various regions of the vocal tract to accurately describe it. Thus, the 13 kinematic position parameters we use to describe the vocal tract may be inadequate.

Regardless of the reasons of the poor performance of predictions using this method of assuming linearity, it seems clear that we should be cautious of using this method to find
articulatory fibers. We hence do not make a further attempt to complete the kinematic set. Without completing the articulatory fibers, we cannot hope to arrive at a precise solution to acoustic to kinematic mapping. We hope that the dataset is big enough to include multiple points from articulatory fibers from the sounds, and we can therefor strive for the improved performance.
Bibliography


