

© 2013 Erin Christine Rusaw

MODELING TEMPORAL COORDINATION IN SPEECH PRODUCTION
USING AN ARTIFICIAL CENTRAL PATTERN GENERATOR NEURAL
NETWORK

BY

ERIN CHRISTINE RUSAW

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Linguistics
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Doctoral Committee:

Professor Jennifer S. Cole, Chair
Professor Gary Dell
Professor Mark Hasegawa-Johnson
Assistant Professor Ryan Shosted
Assistant Professor Annie Tremblay, University of Kansas

Abstract

Researchers have made great strides in developing formal and biologically plausible models of the speech production and planning system. Models based on oscillators have been successful in simulating gestural dynamics, and more recently, some suprasegmental timing patterns. This dissertation explores how different levels of phonological organization (the gesture, syllable, stress group, and phrase) are coordinated in speech production by measuring the effects of that coordination on acoustic and behavioral measures of speech production. It accomplishes this by modeling those coordination patterns using an artificial neural network (ANN) model which incorporates oscillators, inspired by central pattern generators (CPGs), a type of neural circuit which underlies other animal behaviors.

Included in this thesis are a description of the model, named the Neural Oscillator Model of Speech Timing and Rhythm (NOMSTR), and three empirical studies designed to test NOMSTR's usefulness as a tool to model interactions between levels of phonological structure and to simulate those interactions' effects on speech timing. Chapter 4 describes a study of NOMSTR's ability to model the metrical structures of French and English utterances, through comparing the syllable durations and locations of accents and phrase boundaries in spontaneous utterances and repetitions of those utterances with the syllable durations and locations of accents and phrase boundaries in the simulations of the utterances. Chapter 5 describes a study investigating the interactions between levels of prosodic structure in French and in English by measuring the effects of accent-boundary proximity on syllable duration; as well as simulations of those effects using the model. Chapter 6, comprising the second part of the thesis, describes the background of studying speech errors as errors in speech timing, and of using oscillator-driven and ANN models to simulate speech errors; it also describes an investigation of how syllable structure (and super-syllabic structure) affects speech error distribution, using data from a speech error production study and an extension to NOMSTR.

Major findings of Study 1 that are an artificial neural network model based on oscillators can be useful for simulating prosodic timing, even given the variability of timing in natural speech—NOMSTR was very successful in generating simulations of spontaneous utterances with a variety of different timing structures, in both French and English. This suggests that despite non-rhythmic influences on spontaneous speech, such as syllable

content and idiosyncratic duration changes, much of the temporal structure of spontaneous speech can be modeled by a system whose timing rhythmic and *regular*, if not isochronous.

Study 2 showed that In English, accented syllables near a phrase boundary were found to undergo more lengthening than adjacent unaccented syllables. This was an interaction which had been described previously (Turk & Shattuck-Hufnagel 2007), but had not been explained using existing models of prosodic timing. This effect was simulated in NOMSTR by modeling a prosodic structure in which accents and phrase boundaries inhibit (thus lengthening) the syllable, while phrase boundaries excite accents, providing nearby accents greater syllable lengthening power. In French, the lengthening of syllables with Initial Accents, or syllables immediately following the end of an accentual phrase, was found to be lessened in the presence of an upcoming IP boundary. Because of the coincidence of Intonational and Accentual Phrase boundaries in French, NOMSTR provided two possible ways to simulate this effect: either with AP boundary crowding or by modeling an inhibitory connection between the IP and AP nodes. Study 3 found that principles of the Articulatory Phonology model of suprasegmental structure can be integrated into a model of serial speech production in order to simulate aspects of speech error behavior which other models have been unable to explain, such as C/V error asymmetry, and error dependence.

This dissertation demonstrates how an artificial neural network which incorporates oscillators can be a powerful tool for modeling the interactions between elements of phonological structure and for stimulating speech timing patterns, and additionally that a common underlying model architecture can be useful for stimulating multiple languages and multiple levels of structure.

To my husband

Acknowledgments

This dissertation would not have been possible without the guidance and the help of many people. First and foremost, my utmost gratitude to my advisor, Dr. Jennifer Cole, who has supported me throughout my graduate career and the dissertation process. She has guided my development as a scholar through many stimulating meetings, both in-person and over Skype; thorough and constructive feedback on dozens of papers and presentations; opportunities to teach and travel to present my work; and lots of advice.

To my other committee members: Dr. Gary Dell, for his vast font of psycholinguistics knowledge, and always insightful contributions to research discussions and talks; Dr. Ryan Shosted, for teaching me the ins and outs of phonetics, and for requiring his linguistics students to do math and use MATLAB; Dr. Annie Tremblay, for her suggestions about my French language materials and analyses; and Dr. Mark Hasegawa-Johnson, for his expertise in computational modeling of speech and for on-the-job training in conference organization.

Thanks also to: Dr. Rakesh Bhatt, for his enthusiastic supervision of my qualifying paper, even though I didn't go on to do my thesis in socio; Dr. Louis Goldstein and the rest of the Speech Production and Articulation kNoWledge Group (SPAN) at the University of Southern California for providing me with an academic and intellectual home-away-from-home as I was finishing this dissertation; Dr. Rhanor Gillette and his T.A. Jeff Brown, who taught me about central pattern generators, and responded encouragingly to my idea to apply the concept to speech.

To Cathy Penny, Geraldine Moore, and Mary Ellen Fryer, Linguistics Department and School of Languages, Cultures, and Linguistics staff, for their prompt, cheerful, and expert help with all things administrative and logistical.

To the members of LSO, F³, and ExPhon, and my other grad school colleagues, especially (but not limited to) Karen Lichtman, Matt Garley, Amanda Heunsch, Daniel Scarpace, Tim Mahrt, Nicole Wong, Eunice Chung, John Jang, Shawn Chang, Meg Osfar, Lisa Pierce, Stephanie and Casey Coughlen, and Meryl and Scott Hajek, for surrounding me with an awesome community of scholars and friends.

Finally, deepest thanks to my family, who have never been less than completely

supportive of me in my pursuits. To my parents, for teaching me the value of higher learning and continuing intellectual growth, through their words and their lives. To my sister, for being a constant friend, and for motivating me by demonstrating how awesome it is to be graduated and gainfully employed. To my in-laws, for their love and encouragement, and for their understanding all the times I worked on my dissertation during holiday visits. To my husband I owe the most thanks, as he has encouraged me to follow my dreams my whole adult life, and has supported me emotionally (all of the time), as well as financially (some of the time) and physically (occasionally).

My doctoral research was funded in part by National Science Foundation award 1155592 to Erin Rusaw and Jennifer Cole, and by National Institutes of Health award HDO44458 to Gary Dell, Cynthia Fisher and Jennifer Cole.

Table of Contents

List of Figures.....	viii
List of Abbreviations.....	xii
Chapter 1 Introduction.....	1
Chapter 2 Speech Rhythm and CPG Models.....	8
Chapter 3 Neural Oscillator Model of Speech Timing and Rhythm (NOMSTR).....	15
Chapter 4 Prosodic Timing in English and French Utterances.....	20
Chapter 5 Interaction Between Prosodic Level and Its Effect on Prosodic Timing.....	57
Chapter 6 Effects of Phonological Structure on Speech Errors.....	72
Chapter 7 Conclusions.....	96
Appendix A Production Studies.....	100
Appendix B NOMSTR Scripts for MATLAB/Freemat.....	116
References.....	162

List of Figures

1-1	Diagram of the traditional gestural score (left) and gestural coupling score (right) of the English word “spot” /spot/. Solid connecting lines indicate in-phase coupling, and dashed lines out-of-phase coupling.....	5
2-1	Diagram of the respiratory CPG in the Lymnaea snail (sensory and motor projections removed). This model was first hypothesized from behavior and later verified in vitro and in vivo.....	12
3-1	Diagram of the core NOMSTR architecture.....	16
3-2	Graph of the output over time of NOMSTR’s 3 thresholded nodes with oscillators set at 3 different frequencies but no connection weights between the thresholded nodes. Labeled on this graph is the period of the small-period oscillator in arbitrary units used for this example output.....	17
4-1	A diagram of of English Subject o's production of English Utterance 1 and the simulation of Utterance 1.....	23
4-2	Proportional differences in duration of each syllable from the mean – English Subject o's production of English Utterance 1 vs. Simulated English Utterance 1.....	24
4-3	Syllable durations, English Utterance 1.....	25
4-4	Proportion of syllable durations longer or shorter than mean – all productions of English Utterance 1 vs. Simulated English Utterance 1.....	26
4-5	Proportional differences in duration of each syllable from the mean – all productions of English Utterance 1 vs. Simulated English Utterance 1.....	27
4-6	A diagram of of English Subject o's production of English Utterance 2 and the simulation of Utterance 2.....	28
4-7	Proportional differences in duration of each syllable from the mean – English Subject o's production of English Utterance 2 vs. Simulated English Utterance 2.....	29
4-8	Syllable durations, English Utterance 2.....	31
4-9	Proportion of syllable durations longer or shorter than mean – all productions of English Utterance 2 vs. Simulated English Utterance 2.....	31
4-10	Proportional differences in duration of each syllable from the mean – all productions of English Utterance 2 vs. Simulated English Utterance 2.....	32

4-11	A diagram of of English Subject o's production of English Utterance 3 and the simulation of Utterance 3.....	33
4-12	Proportional differences in duration of each syllable from the mean – English Subject o's production of English Utterance 3 vs. Simulated English Utterance 3.....	34
4-13	Syllable durations, English Utterance 3.....	35
4-14	Proportion of syllable durations longer or shorter than mean – all productions of English Utterance 3 vs. Simulated English Utterance 3.....	36
4-15	Proportional differences in duration of each syllable from the mean – all productions of English Utterance 3 vs. Simulated English Utterance 3.....	36
4-16	Average residual difference in syllable durations from simulation, English	38
4-17	A diagram of the syllable durations and prosodic structures of French Subject o's production of French Utterance 1 and the simulation of Utterance 1.....	40
4-18	Proportional differences in duration of each syllable from the mean – French Subject o's production of French Utterance 1 vs. Simulated French Utterance 1.....	41
4-19	Syllable durations, French Utterance 1.....	42
4-20	Proportion of syllable durations longer or shorter than mean – all productions of French Utterance 1 vs. Simulated French Utterance 1.....	43
4-21	Proportional differences in duration of each syllable from the mean – average of all productions of French Utterance 2 vs. Simulated French Utterance 2.....	44
4-22	A diagram of of French Subject o's production of French Utterance 2 and the simulation of Utterance 2.....	45
4-23	Proportional differences in duration of each syllable from the mean – French Subject o's production of French Utterance 2 vs. Simulated French Utterance 2.....	46
4-24	Syllable durations, French Utterance 2.....	47
4-25	Proportion of syllable durations longer or shorter than mean – all productions of French Utterance 2 vs. Simulated French Utterance 2.....	48
4-26	Proportional differences in duration of each syllable from the mean – all productions of French Utterance 2 vs. Simulated French Utterance 2.....	48
4-27	Syllable durations, “corrected” productions of French Utterance 2 and Simulated French Utterance 2.....	49

4-28	A diagram of of French Subject o's production of French Utterance 3 and the simulation of Utterance 3.....	50
4-29	Proportional differences in duration of each syllable from the mean – French Subject o's production of French Utterance 3 vs. Simulated French Utterance 3.....	50
4-30	Syllable durations, French Utterance 3.....	52
4-31	Proportion of syllable durations longer or shorter than mean – all productions of French Utterance 3 vs. Simulated French Utterance 3.....	52
4-32	Proportional differences in duration of each syllable from the mean – all productions of French Utterance 3 vs. Simulated French Utterance 3.....	53
4-33	Average residual difference in syllable durations from simulation, French.....	54
5-1	Lengthening by syllable in English phrase-final words.....	59
5-2	Inhibitory connections necessary to simulate English results.....	60
5-3	Excitatory and inhibitory connections necessary to simulate English results.....	61
5-4	Simulation of English penultimate accent condition	62
5-5	Simulation of English antepenultimate accent condition.....	63
5-6	Simulated lengthening by syllable in phrase-final words.....	64
5-7	Syllable positions analyzed for duration in French sentences.....	65
5-8	Effect of distance from IP boundary on Initial Accent lengthening.....	66
5-9	Simulation of Initial Accenting on a French 4-syllable target word.....	68
5-10	Simulation of Initial Accent lengthening prevented due to Accentual Phrase oscillator period.....	69
5-11	Simulation of Initial Accent lengthening prevented due to Intonational Phrase oscillator inhibiting Accentual Phrase oscillator	70
6-1	Diagram of syllable structure in Articulatory Phonology.....	73
6-2	Diagram of traditional syllable structure.....	75
6-3	Errors by segment type.....	77

6-4	Errors by syllable position.....	77
6-5	Independence of C and V errors.....	78
6-6	Occurrence of isolated errors.....	79
6-7	Diagram of the architecture of NOMSTR with the gesture string output extension....	80
6-8	Oscillators and tuned neuron layer.....	82
6-9	Tuned neuron layer and nucleus output layer.....	83
6-10	Tuned neuron layer and onset/coda output layer.....	84
6-11	Nucleus output layer and onset/coda output layer.....	85
6-12	Excitatory connections between the syllable thresholded node and nucleus output nodes.....	86
6-13	Example output of nucleus output nodes.....	88
6-14	Example output of onset/coda output nodes.....	88
6-15	A correctly simulated [s].....	89
6-16	An erroneously simulated [s].....	90
6-17	Simulated errors by segment type.....	91
6-18	Simulated errors by syllable position.....	92
6-19	Independence of simulated C and V errors.....	93
6-20	Occurrence of isolated simulated errors.....	94

List of Abbreviations

ANN	Artificial Neural Network
AP	Articulatory Phonology
AP	Accentual Phrase
CVC	Consonant Vowel Consonant (syllable structure)
CPG	Central Pattern Generator
IA	Initial Accent
IP	Intonational Phrase
NOMSTR	Neural Oscillator Model of Speech Timing and Rhythm
ONC	Onset Nucleus Coda (syllable structure)

Chapter 1

Introduction

1.1 Introduction

The dual goals of linguistic science are to understand the nature of language in terms of its component structures and their patterns of coordination in words, phrases, and larger utterances, and to understand the nature of the cognitive, perceptual and motor systems that support human linguistic behavior. A specific challenge is to understand how a universal capacity for language rooted in human biology can give rise to a remarkable range of patterned variation across distinct languages and dialects, among speakers, and as a function of the communication task. Here I restrict my focus to language production in speech, setting aside the parallel concerns of language perception and the analysis of language expressed through signed or written modalities. Due to the limits of technology and human subject research, the biological systems at the root of spoken language production—the neural circuits responsible for speech planning and production and their sensory-motor projections to the vocal tract—are difficult to study directly and must be inferred from observations of the behavior of the motor system at its peripheral level in speech articulation and from the acoustic speech output.

Drawing on advances in technologies for vocal tract imaging and signal processing, researchers have made great strides in recent years in developing formal and biologically plausible models of the speech production system. A leading example of this approach is research in the Articulatory Phonology paradigm, begun by Browman & Goldstein (1986), which investigates temporal coordination in phonology. Articulatory Phonology (AP) research applies task dynamics, a general model of bodily coordination developed to describe skilled actions, to model articulator coordination in the speech production system. The initial focus of work in AP was modeling gestural and inter-

gestural dynamics in terms of articulator coordination relations (Browman & Goldstein, 1989; Saltzman et al., 2008). But a complete model of articulatory timing must also consider how temporal coordination at higher levels of phonological encoding, such as the syllable, stress foot, and phonological phrase impacts the temporal dynamics of speech. AP research has begun to incorporate prosodic (“supragerstural”) dynamics into a model of speech production, especially making use of mathematical models of coupled oscillators to describe coordination between the syllable, foot, and phrase (O’Dell & Nieminen, 1999; Saltzman et al., 2008).

Despite the early success of this approach in modeling speech timing data, a model based on task dynamics and coupled oscillators is not complete until its relationship to neural structures and their behavior can be determined. Because speech is generated in the brain—a neural network—I focus on artificial neural networks as a biologically plausible model of speech production. The research in this thesis seeks to understand the how different levels of phonological organization (the gesture, syllable, stress group, and phrase) are coordinated in speech production by measuring the effects of that coordination on acoustic and behavioral measures of speech production, and by modeling those coordination patterns using an artificial neural network (ANN) model which incorporates oscillators, inspired by central pattern generators (CPG), a type of neural circuit which underlies other animal behaviors. A major innovation this thesis contributes to the study of prosody and speech timing is the Neural Oscillator Model of Speech Timing and Rhythm (NOMSTR), which is designed to be a flexible tool for investigating the systems which affect speech rhythm and timing through simulation.

In order to construct the first version of this model and gain a greater understanding of speech timing at the suprasegmental level, this thesis investigates the following research questions: 1) can an ANN model based on oscillators be useful for simulating prosodic timing, given the variability of timing in natural speech?; 2) What is the relationship between the syllable, stress group, and phrase, in terms of their temporal coordination, and how does this relationship differ between languages with differing prosodic systems (specifically English and French)?; 3) can the same model

architecture be used to model these relationships and simulate their effects on prosodic timing in both languages?; 4) How is the coordination of syllable components with each other and with higher levels of phonological structure reflected in speech timing errors (slips of the tongue)?; and 5) can an ANN model based on oscillators be useful for modeling those interactions and simulating their effects on speech errors?

The first three of these questions, which concern prosody, are addressed in the first part of this thesis. Chapter 2 reviews the literature describing the rhythm and prosody systems of French and English. Chapter 3 discusses the nature of CPGs and the benefits of using this type of model for speech timing, and describes the architecture of NOMSTR, the CPG-inspired model created for this purpose. Chapter 4 describes a study of NOMSTR's ability to model the metrical structures of French and English utterances, through comparing the syllable durations and locations of accents and phrase boundaries in spontaneous utterances and repetitions of those utterances with the syllable durations and locations of accents and phrase boundaries in the simulations of the utterances. Chapter 5 describes two studies, investigating the interactions between levels of prosodic structure in French and in English by measuring the effects of accent-boundary proximity on syllable duration, as well as simulations of those effects using the model.

Chapter 6, comprising the second part of the thesis, addresses the final two research questions. This chapter describes the background of studying speech errors as errors in speech timing, and of using oscillator-driven and ANN models to simulate speech errors; it also describes an investigation of how syllable structure (and super-syllabic structure) could affect speech error distribution, using data from a speech error production study and an expanded version of NOMSTR. Before delving into prosodic timing or speech timing errors specifically, I will first provide some background on the study of the temporal organization of speech.

1.2 Background

One source of evidence for temporal coordination in speech production is found in duration patterns of the units of production. The durational patterns of speech have

been studied at the level of individual segments or articulatory gestures, and at “higher” levels of prosodic organization, as briefly reviewed here.

There is a long history of research in phonetics examining the durational properties of segments (e.g., Crystal and House 1988; 1990). Research in Articulatory Phonology takes a different perspective, eschewing the segment-centered approach and instead modeling durational patterns as the reflection of the temporal coordination of overlapping articulatory gestures. In AP, speech gestures are described in terms of both temporal properties (onset and offset time, duration, velocity) and physico-spatial properties (equilibrium point target, stiffness, damping). Each gesture follows a trajectory which begins at its onset time, reaches its target, and ends at its offset time. The gestures in an utterance are related by the degree to which they overlap in time—their relative timing (Browman & Goldstein, 1992). For example, the English word *dad* (/dæd/) is represented by a fairly long gesture whose target is the vocal tract configuration for the vowel /æ/, and two shorter gestures whose targets are the closure of the tongue blade against the alveolar ridge (the initial and final /d/). The first alveolar closure gesture begins at the same time as the vocalic gesture, and so is entirely overlapped by it, and the second alveolar closure gesture begins at the vocalic gesture’s offset. Initially, the temporal relations between gestures were described using points at which two gestures were synchronized (Browman & Goldstein, 1995), but in later work the coordination between gestures was established by associating each gesture with an oscillator and controlling the relative timing, or phase relations, between oscillators using mathematical coupling, which establishes stable timing relationships (Saltzman & Byrd, 2000).

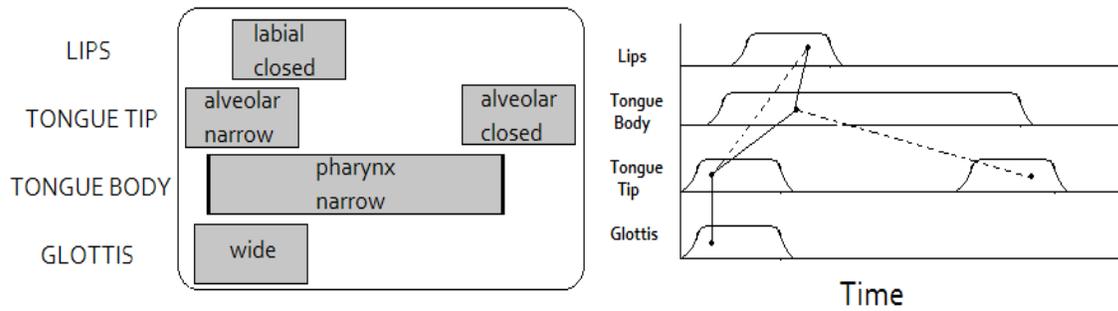


Figure 1-1: Diagram of the traditional gestural score (left) and gestural coupling score (right) of the English word “spot” /spɒt/. Solid connecting lines indicate in-phase coupling, and dashed lines out-of-phase coupling (Saltzman et al. 2008).

Nam & Saltzman (2003) extended the use of coupled oscillator relationships between articulatory gestures by showing how specifying coupling relations between gestures in each syllable position can model the differences in coordination between the onset and coda and the nucleus, as observed in prior speech studies (Browman & Goldstein, 1988; Byrd 1996). Coupled oscillators are also used to model the coordination between prosodic structures at different levels above the syllable. Suprasegmental coupled oscillator models began with Cummins & Port (1998) to model the tendency of English-speaking subjects speaking to a metronome to produce stressed syllables at harmonics of the metronome duration. O’Dell and Nieminen (1999), and Barbosa (2002) use coupled oscillator models to describe inter-language differences in the relationship between syllables and stress groups. This type of suprasegmental coupled oscillator model is incorporated into the AP framework in Saltzman et al. (2008), which describes the effect of suprasegmental oscillators and suprasegmental “temporal modulation gestures” on gestural timing.

In a different vein of research, durational patterns in speech have been investigated in relation to rhythm and prosodic organization. Much of the early research on speech timing addressed the question of rhythm with a focus on isochrony of the prosodic elements of speech, based on the regular timing of syllables, stresses or pitch accents (e.g. Pike, 1945). Although the existence of truly regular time patterns in normal speech has been debunked in subsequent work (Lehiste, 1977; Dauer, 1983; Morton et

al., 1976; Tuller & Fowler, 1980; Cummins, 2005) properties of speech timing continue to inform models of prosodic organization, with a focus on variability of timing across stretches of speech (Grabe & Low 2002). Taken in the broader context of prosodic organization, rhythmic patterns reflect the temporal coordination of syllables, stress groups, and possibly larger prosodic units. Phonological models define prosody in terms of multiple layers of hierarchically organized structure, comprising repeating elements of different sizes that influence the repeating elements of the next smallest size contained within them (Nespor & Vogel, 2007). Repeating elements which have been proposed for spoken language are (by increasing size) the syllable, the stress foot or stress group, the phonological word, and several sizes of phonological phrases (e.g. Silverman et al., 1992; Hayes, 1995; Buckley, 2009). These levels of repeating structure define the prosodic hierarchy, and are the basis of the metrical phonology model of stress at the word and phrase levels. Studies of the temporal patterns of syllables, stresses, accents, and phrases in speech provide an important source of evidence for prosodic structure in phonological encoding (e.g., Edwards & Beckman 1988; Edwards, et al. 1990; Wightman et al. 1990, among many others).

In addition to the studies of speech timing as it relates to rhythm classification and gestural timing, speech timing is also implicated in a number of studies using other methods and experimental tasks, including:

- 1) Speech cycling studies, in which the location of stressed syllables in a repeated phrase is measured in relation to the cycle of the entire phrase, demonstrate that speech has the capacity to be very temporally regular and cyclical (e.g. Port & Cummins, 1998).
- 2) Phrase-final lengthening studies, which compare the durations of syllables in pre-boundary and non-pre-boundary conditions, illustrate the influence of prosodic phrase structure on the constituent syllables (and gestures) (e.g. Scott, 1982; Turk & Shattuck-Hufnagel, 2007).
- 3) Research on speech errors, which shows how slips of the tongue result from mistiming or mis-ordering of segments or gestures during the production of an utterance (e.g. Dell et al., 1997; Goldstein et al., 2007).

Taken together, these studies point to the central role of temporal coordination in speech production for determining not only the temporal patterns of gestures, segments or syllables, but also for the phenomenon of speech rhythm, the effects of prosodic context on duration, and the role of temporal coordination in speech errors. The work described in the following chapters contributes further to this line of inquiry by testing the success of neural network models of temporal coordination in modeling these various manifestations of speech timing.

Chapter 2

Speech Rhythm and CPG Models

2.1 Background

Returning to the theme of the Introduction, one of the fundamental goals of a linguistic model of speech production, such as the AP model, is to understand the basis of speech production in the biological system of the human brain and vocal tract, universal to all humans. In order to do this, it must be determined which aspects of any individual language are due to the properties of the universal system, and which are specific to an individual language.¹ Thus, a successful model of speech production must have a core architecture that reflects the properties shared between languages, as well as variables that can be changed to simulate the differences between languages. Focusing on the temporal patterns of speech, a successful model must have an architecture that represents the elements that are related through temporal coordination (e.g., gestures, syllables, feet, phrases), as well as parameters that can be varied to account for differences in timing patterns across languages.

The studies in the first section of this dissertation focus on American English and French, languages whose prosodic systems have been well-studied, and which exhibit strikingly different properties at the level of prosodic organization and in their temporal patterns. Evidence from experiments on speech timing is examined in an effort to understand which aspects of temporal coordination among units within or across prosodic levels are shared between languages and which are language specific. Traditionally, English and French have been offered as prime examples of the two opposing language rhythm classes, stress-timed and syllable-timed (Pike, 1945), and even in later models of rhythm classification that eschew this dichotomy, English and French are presented as distinct rhythmic types.

Work on the prosodic organization of French and English claim that in both

¹ The model should also provide variables to account for differences across speakers, or due to factors related to the communication context. Here I restrict my focus to variation across languages.

languages syllables and words are grouped into phonological phrases, but with differences in the number and type of phrase levels, and in the status of pitch accents or prominence within the phrase. (see Di Cristo, 1998; and Wells, 2006 for reviews). French is considered to have at least two, and possibly three, hierarchically ordered levels of phonological phrase structure: the accentual phrase, the intonational phrase, and possibly the intermediate phase. Non-reduced syllables near the right edges, and sometimes the left edges of these phrases exhibit pre-boundary effects like lengthening and pitch-accent assignment (DiCristo, 1998; Jun & Fougeron, 2000; Hirst et al., 2007). English is described as having intonational and intermediate phrases, with final lengthening at the end of each. But unlike French, English is also claimed to have prominence (“accent” or “stress”) which is not associated with prosodic phrasal boundaries (Hirst, 1998). As a note, in this thesis, and following Bolinger (1958) “stress” is used to refer to the status of a syllable in a lexical item which refers to its relative strength or prominence compared to the other syllables in the word, and “accent” is used to refer to prominence at the phrase level, expressed in English through lengthening, increased intensity, hyperarticulation, and pitch accent. In this account, “stress” is the trait of being accentable; unstressed syllables cannot be accented.

This difference between the French and English prosodic systems is associated with their differential treatment of stress at the word level. In English, which has lexical stress, each word has a partly idiosyncratic stress assignment, which may be lexically contrastive (Hayes, 1982). At the prosodic phrase level, nuclear and pre-nuclear accents are assigned to stressed syllables. In French, stress is predictable: the last full syllable of a word receives primary stress, and the first syllable in a word may receive an optional secondary stress (Di Cristo, 1998). French phrasal accent or final accent (FA) is assigned to the last stressed syllable in a prosodic phrase. This is similar to the English nuclear pitch accent, which also occurs at the right edge of phrases, except that because of the regularity of French stress assignment, the final stress syllable is always the rightmost full vowelised syllable in the phrase, which is very often also the final syllable in the phrase; in English the nuclear accent may be several syllables removed from the phrase boundary, possibly even in the non-final word (Hirst, 1998).

Prosodic structure affects temporal patterns in both English and French. In English there is lengthening due to lexical stress, phrasal accent, and at the right edge of prosodic phrases, while in French the primary durational effect from prosody is phrase-final lengthening. In French, because of the way the acoustic effects of final accenting coincide with the acoustic effects of phrase boundaries, it is difficult to examine the interaction between accent and phrasing on duration, which obscures any durational evidence for the temporal coordination between stress/accent and prosodic phrase structure. The finding of increased effects of lengthening as prosodic boundary strength increases (Di Cristo, 1998) suggests the lengthening effect may be compounded at successively higher levels of prosodic phrasing, but because of the temporal overlap between stress/accents and phrase boundaries it is extremely difficult to tease apart their effects.

In English, there is clearer evidence for the separate yet interacting effects of accent and phrasing on temporal patterns, as the last accented syllable of a phonological phrase may occur some distance from the phrase's right edge (Hirst, 1998). Turk & Shattuck-Hufnagel (2007) found that accenting and phrase-final lengthening effects interact in English in that accented syllables near a phrase boundary show additional lengthening (compared to the same syllable phrase-medially), even though unaccented syllables intervening between the accented and phrase-final syllables do not.

It may be possible to examine the interaction of the effect of multiple prosodic levels in a similar way in French by focusing on the French initial accent (IA). IAs were found by Atésano et al. (2007) to sometimes appear on initial (secondary stress) syllables at the left edge of prosodic phrases of several sizes, even as small as the phonological word. In some conditions of that study, speakers produce IAs on the first syllables of words at the right edge of phonological phrases. This thesis investigates whether there is a similar interaction of the lengthening effects of different prosodic tiers in French as in English by applying Turk & Shattuck-Hufnagel's method and comparing the durations of IA syllables in phrase-final words with the durations of those same syllables phrase-medially.

2.2 Artificial neural network (connectionist) models of speech production and CPGs

Having discussed the nature of the system I am attempting to understand and model, I turn now to the nature of the model which will be used in this work. Connectionist or neural network models have been commonly used in linguistics, especially psycholinguistics, to model language and speech. Several studies by Dell and colleagues (Dell 1998; Dell et al. 1997; Dell et al. 1999) have proposed neural network models for phonological and lexical access in speech production, as well as for serial ordering. The WEAVER++ model produced by Roelofs (2000) has also been successfully used to simulate speech production at multiple structural layers from the conceptual level down to the segment level. Artificial neural networks (ANNs) are useful for modeling speech because (i) they leverage the power of groups of biologically-inspired artificial neurons to process information, with the understanding that since the human brain is made up of neurons, neurons are a good way to model the linguistic data that the brain produces and perceives; and (ii) because their potential for hierarchical structure is consistent with hierarchical linguistic models of speech. As Dell et al. (1999) point out, though, standard connectionist models of language lack the ability to simulate time progression, which is essential to speech production. Over the past few decades, studies from both the cognitive science and linguistic communities have proposed that a specific type of neural network which does have this ability, a central pattern generator (CPG), may be responsible for temporal coordination in speech (e.g. Lund & Kolta, 2006; Barlow & Estep, 2006).

A central pattern generator is a circuit of neurons in the central nervous system of an animal which generates a coordinated patterned output, or sometimes several different patterns of output (Delcomyn, 1980). CPGs are most commonly indicated in locomotor behavior (flight and gait), but they have been described for a variety of behaviors (e.g. Wilson, 1961; Grillner, 1985; Taga 1995; Ijspeert, 2001). CPGs are commonly powered by sets of coupled oscillator neurons, which are responsible for

generating their rhythmic output (Delcomyn, 1980). For example, Figure 2-1 shows a diagram of the simple CPG responsible for the cyclical respiration behavior in the *Lymnaea* pond snail (Taylor & Lukowiak, 2000). The pattern of inhaling and exhaling in the snail is driven by the oscillating output of interneurons IP3I and VD4, which alternate firing due to mutual inhibition.

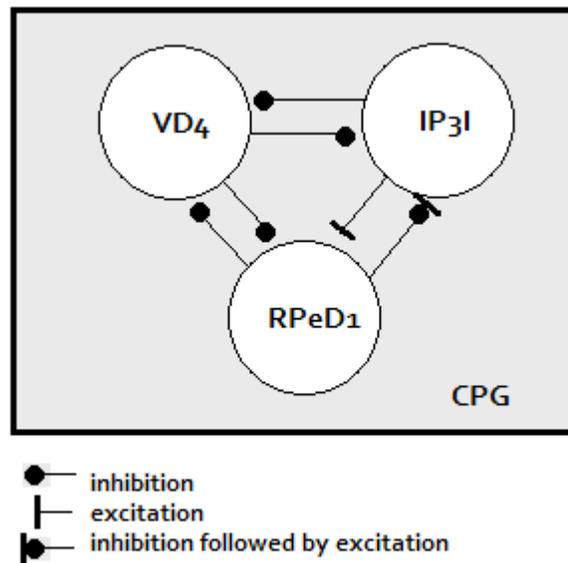


Figure 2-1: Diagram of the respiratory CPG in the *Lymnaea* snail (sensory and motor projections removed). This model was first hypothesized from behavior and later verified *in vitro* and *in vivo* (Taylor & Lukowiak 2000).

More complex CPGs, with more cells and connections, often have the ability to generate multiple stable output patterns and coordinate several connected tasks using multiple connected tiers of oscillators, such as in vertebrate locomotion.

CPGs have the ability to produce both the cyclical nature and flexibility of speech timing, and a CPG model can be used to represent the hierarchical nature of prosodic phonology. The successful use of coupled oscillators in Articulatory Phonology to model the coordination between levels of prosodic elements also provides evidence that a model using coupled oscillators may be appropriate for temporal coordination of prosodic units. Speech production has much in common with other behaviors produced by or modeled with CPGs, such as locomotion: underlying repetitive motions and a

perceived or possible isochrony (Port & Tajima, 1999), flexibility in timing and pattern (Tajima, 1998), adjustment to sensory and proprioceptive feedback (Gracco & Abbs, 1988), and the ability to integrate non-rhythmic motor tasks (Browman & Goldstein, 1989). Because of the compatibility of CPGs with established models of speech production, as well as the similarity of speech to other CPG-driven behaviors, and its shared musculature with centrally patterned behaviors such as respiration, several researchers have suggested that a CPG may underlie the gross motor patterns of speech production. Von Euler (1983) provides a diagram of a theoretical CPG for respiration, which he claims would necessarily be recruited during speech, while Smith (1992), Barlow & Estep (2006), and Lund & Kolta (2006) propose that speech production may recruit CPGs used to control sucking, masticating, and swallowing behavior as well as respiration. Gracco & Abbs (1988) discuss how somatosensory feedback may interact with a centrally generated motor pattern to achieve articulatory targets in speech.

A CPG model of prosodic phonology, if successful, will only be a small piece of the speech production system model. The role of a CPG in speech production is to integrate multiple levels of linguistic structure—syllables, phrasing, and prominence information—into one output pattern, and apply that output to the string of gestures which comprise an utterance. Such a system allows a single sound stream, resulting from a single motor stream, to express information pertaining to multiple layers of information without speakers having to store the exact articulatory pattern of each utterance individually, which is of course impractical. The system of prosodic coordination is only one of many influences on the final temporal properties of an utterance. The intrinsic length of gestures, syllables, and words, as well as disfluencies, pauses, and message planning time also influence speech timing. These outside influences are the reason that even if the output of the system of prosodic coordination is perfectly cyclical and regular, speech, especially spontaneous speech, is not.

2.3 Interim Summary

From the previous work on the prosodic organization of spoken language, artificial neural network models of speech, and central pattern generators, we can

conclude the following:

(1) Spoken language timing is generated by a multi-tiered, cyclical system, which produces a motor pattern and sound stream, and properties of this system can be inferred from observations of the speech signal and motor behavior in speech production.

(2) Although presumably generated by the same underlying mechanism, languages vary significantly in prosodic temporal organization. French and English are examples of two languages with well-studied prosodic systems that have quite different properties.

(3) Spontaneous speech is not very regular, because there are other influences on the timing of an utterance besides its prosodic organization.

(4) ANNs have been successful in modeling aspects of speech production, and a CPG model is particularly suited to modeling speech timing.

Chapter 3

Neural Oscillator Model of Speech Timing and Rhythm (NOMSTR)

The task of constructing and testing a general model of timing in relation in speech production structure is much larger than the scope of a thesis. The goal of this work is to take a first step in testing a neurologically plausible model of temporal coordination of prosodic units. The studies in this dissertation investigate the temporal behavior of two very different prosodic systems, and speech behavior at three distinct levels of structure, and apply the same model to each, in an attempt to build a model that is holistically accurate and generalizable.

The basic model used in this work, NOMSTR, is an artificial neural network created in MATLAB using modeling principles from Anastasio (2009). The model comprises three half-center oscillators which serve as the inputs to three thresholded integrate-and-fire artificial neurons, as shown in Figure 3-1.

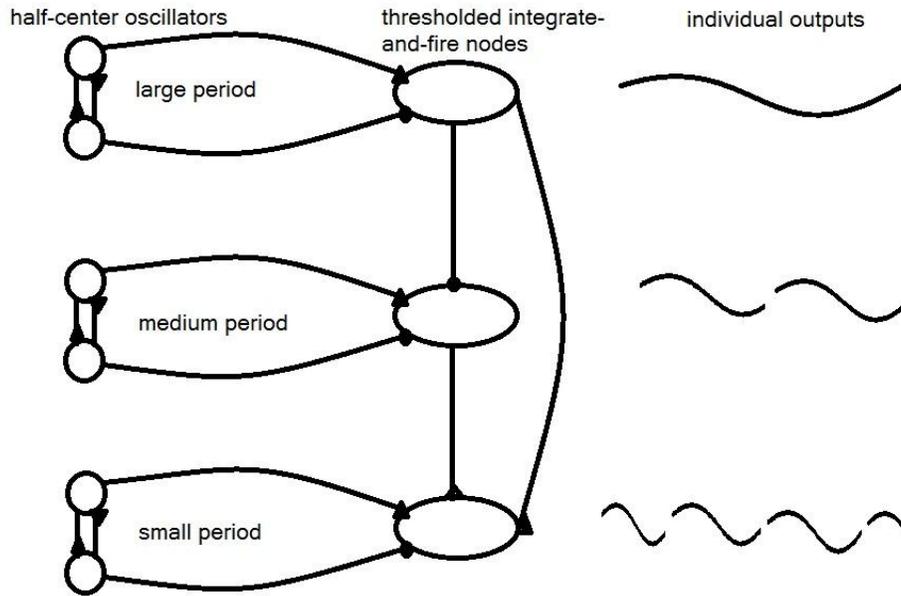


Figure 3-1: Diagram of the core NOMSTR architecture

One cell of each half-center provides excitatory input to its connected thresholded node, and one cell provides inhibitory input to the same. The resulting level of activation in each of the thresholded nodes is modeled as a sinusoid as shown in Equation 1. The three half-centers oscillate at three different frequencies, so that of the three half-center-thresholded node pairs, there is a small period, medium period, and large period set.

$$(1) \quad \text{activation} = \text{amplitude} * \sin(2 * \pi * \text{frequency} * ((\text{time}) + \text{phase}));$$

Together these represent three metrical levels of phonological organization, such as syllable, stress group, and phrase; or syllable, accentual phrase, and intonational phrase. All three thresholded nodes are connected to each other via excitatory or inhibitory connections, so that the total activation including inputs from other nodes can be described by Equation 2.

(2) total activation = (sinusoidal activation * weight of input from oscillator) + (output of node 1 * weight of connection 1) + (output of node 2 * weight of connection 2) + (output of node 3 * weight of connection 3)

If the activation level of a node does not reach the set threshold (which was generally not varied for the simulations in this thesis), it will not output, but if its activation level is above threshold it produces an output which reflects the sinusoidal curve of its oscillator. Figure 3-2 shows an example output, with no connection weights set between the thresholded nodes.

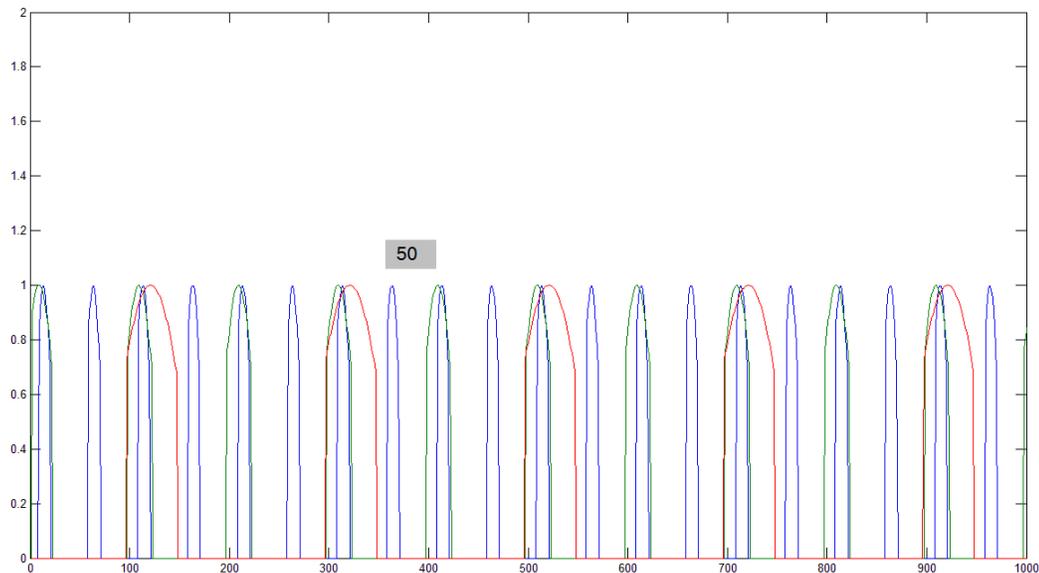


Figure 3-2: Graph of the output over time of NOMSTR's 3 thresholded nodes with oscillators set at 3 different frequencies but no connection weights between the thresholded nodes. Labeled on this graph is the period of the small-period oscillator in arbitrary units used for this example output.

NOMSTR simulates three metrical levels, but it does not have separate

mechanisms for simulating prominence and phrasing (accents vs. phrase-boundary effects). In this model, prominences such as accents in English, and phrases are both modeled as cycles in the activation and firing rate of the artificial neurons. Bursts in activity from the thresholded integrate-and-fire nodes represent both prominences and phrase boundaries – essentially accents are modeled as another type of phrase in terms of neural behavior. The duration between accents, for example, is measured as the duration between bursts of the thresholded node from the oscillator set representing the accent-level metrical tier, in the following studies generally the medium-period set. Syllable duration is measured as the duration between bursts of the thresholded node from the oscillator set representing the syllable, generally the small-period set. Phrase length is measured as the duration between bursts of the thresholded node from the oscillator set representing the phrase, in the following studies generally the large-period set. Manipulable variables in the model are the intrinsic frequency of each oscillator, and the connections between each of the thresholded integrate-and-fire nodes. Without outside interference, the nature of each thresholded node is to produce a burst of activity at the same period as its input oscillator. Excitatory or inhibitory influences on a node can have the effect of accelerating or delaying the next burst, respectively (and increasing or decreasing the simulated firing rate, but that is not the current focus). As a result, the effect of one node firing can induce another node to fire simultaneously, or accelerate or delay the next burst of another node, lengthening or shortening the duration of the node's current period. When used to model speech prosody, this can simulate prosodic events like the coincidence of a prominence with a phrase boundary, the coincidence of the boundaries of two phrases of different size; or the lengthening or shortening of syllable durations at prominences or phrase boundaries.

NOMSTR provides a tool to investigate the influences on speech timing from three distinct sources—the speech production and planning system, the prosodic system of a particular language, and structure of a particular utterance – by providing three levels of modeling. NOMSTR's architecture, a neural network based on a CPG, is designed to model part of the universal speech planning system, the brain. This architecture is also universal; it is used to create all of the simulations in this thesis,

regardless of the language or behavior being simulated. NOMSTR models the interactions between prosodic levels within a language languages through interactions between the thresholded nodes (excitation and inhibition) used to model. All simulations within a language use the same pattern of interactions between the thresholded nodes. Finally, the period and phase of the oscillators can be adjusted to simulate the speech/syllable rate and accent or boundary density of a particular utterance; the period and phase are set only once for each utterance. Parameters are never changed mid-utterance in any of the studies.

The goal of the experiments described in the next two chapters is to investigate the temporal patterns of syllables, accents, and phrases in French and English, two languages traditionally described as having very different prosodic organizations. Experiment 1 specifically focuses on comparing the temporal regularity and overall temporal structures of the two languages, as well as whether repetition can allow the underlying rhythmic structure of an utterance to become clearer. Experiment 2 focuses on the temporal effects syllables, accents, and phrase boundaries have on each other when they interact. An appropriate model of the temporal patterns defined over prosodic elements must account for the different timing patterns in each of these languages, and be able to simulate both systems from the same underlying architecture. The simulation sections of these chapters test NOMSTR's ability to simulate the temporal patterns and interactional effects found in both languages by adjusting the connections between the three levels of the model.

Chapter 4

Prosodic Timing in English and French Utterances

4.1 English

This study uses NOMSTR to simulate the prosodic structure of spontaneous utterances in English and French, focusing on the durational effects of accenting and phrase boundaries on syllables, and the location of affected syllables within each utterance. The data are utterances which were produced spontaneously in direction-giving tasks, as well as repetitions of these same utterances produced by other speakers in a task designed to filter out influences on the timing of the utterance besides the structure (e.g. disfluencies, planning time). Simulations are generated of the prosodic timing of each spontaneous utterance. The simulations are compared both to the original utterance, and to the average across all productions of the utterance, in order to evaluate how well the simulation reproduces the underlying prosodic structure of the utterance, rather than just the idiosyncratic timing of the original production.

4.1.1 Procedure

The English condition of this experiment uses extant materials from corpora. Three utterances were selected for analysis from the English corpus described in Cole and Shattuck-Hufnagel (2011). Each utterance was originally produced spontaneously during a map task in Shattuck-Hufnagel et al. (2004); in Cole and Shattuck-Hufnagel (2011); the original speaker who spontaneously produced each utterance is referred to here as English Subject 0. 10 speakers listened to recordings of the original utterances and repeated each 3 times, including pauses between each repetition to clear the auditory buffer. My first step was to create a simulation of the original 3 spontaneous utterances. For these utterances, I recorded syllable durations (as measured from one center to the next (c.f. Browman and Goldstein 1988), and the locations of accented syllables and phrase boundaries, which were identified by the agreement of two ToBI-

trained transcribers for Cole and Shattuck-Hufnagel (2011).

For each utterance, these three measurements were used as the starting points for setting the three wavelength parameters in NOMSTR. Specifically, the wavelength of the short oscillator was set as the average duration of all non-accented syllables in an utterance, the wavelength of the medium oscillator was set as the average duration of all inter-accent intervals in an utterance, and the wavelength of the long oscillator was set as the average duration of all phrases in an utterance. Next I adjusted the phases, amplitudes, and wavelengths of the oscillators for each utterance so that the simulation more closely matched the utterance data (values of model parameters used to simulate each utterance can be seen in Appendix B.3). In the next section I describe how I compared the simulations to the data and evaluated how well each simulation models its utterance.

4.1.2 Analyses

As stated in the introduction, the role of a central pattern generator in speech production is to integrate multiple levels of linguistic information into a single output pattern. In the simulation of the three utterances in this section, NOMSTR is used to integrate the information about accenting and phrasing in each utterance into the durations of the simulated syllables. Due to the nature of the model used in this thesis, it is not possible to use a single measure of how well it simulates the rhythmic structure of an utterance. Because NOMSTR is not a linear or linear-based model, traditional measures of goodness-of-fit, such as least squares, are not appropriate. Also, NOMSTR simulates several aspects of the data: the location of phrase boundaries and accented syllables, the degree of their effect on syllable duration, and the actual duration of syllables; a single measure cannot evaluate NOMSTR's ability to simulate all of these aspects of the data at once.

In order to help evaluate NOMSTR's simulations, I compare it in this section to a null or dummy "model" consisting of a basic metrical structure which is constructed using the same information from each actual utterance that is fed into NOMSTR in order to produce the simulation of that utterance (labeled in figures as H0). I compare

NOMSTR’s simulation of each spontaneous utterance to the null model in its ability to simulate the temporal pattern of that utterance—the relative positions of metrical elements in the utterance and their effects on each other. At the syllable level, the temporal pattern of the utterance’s metrical structure can be viewed in the location and degree of syllable lengthening (i.e. which syllables are long, which are short, and how long/short are they?). Next I will evaluate the ability of each simulation to model its utterance’s storable, reproducible metrical structure, not just a simulation of the original speaker’s idiosyncratic production of the utterance. To this end, I will compare the location and degree of syllable lengthening in each simulation with the location and degree of syllable lengthening over all repetitions of the utterance it is simulating. I deal with relative and raw durations separately in my analysis because the model developed in this thesis is mainly concerned with rhythmic structure and the interaction between prosodic levels, not with absolute speech rate, so it does not contain an overarching “clock” mechanism which can accelerate or decelerate the entire system. It is possible that the actual speech production system could contain such a mechanism, though, which would allow speakers to produce utterances in which the raw durations of syllables or phrases are very different even though the rhythmic patterns or structures are the same. In order to avoid differences in overall speech rate from confounding my comparison between the data and the simulation, I will deal first with the ability of the simulation to predict variation in the data, and then with its ability to predict raw durations.

4.1.3 Utterance 1

4.1.3.1 Utterance 1 Simulation Compared to Utterance 1 Original Production

Figure 4-1 shows an illustration of Utterance 1 as spontaneously produced by English Subject 0 during a map task and its simulation. Syllable durations are represented by the length of the red and blue bars, the locations of phrase boundaries in the original production are noted by black vertical lines, and accented syllables in the original are marked with asterisks. The locations of peaks of the accent nodes are marked by green horizontal lines, and the locations of peaks of the phrase node are

marked by black horizontal lines.

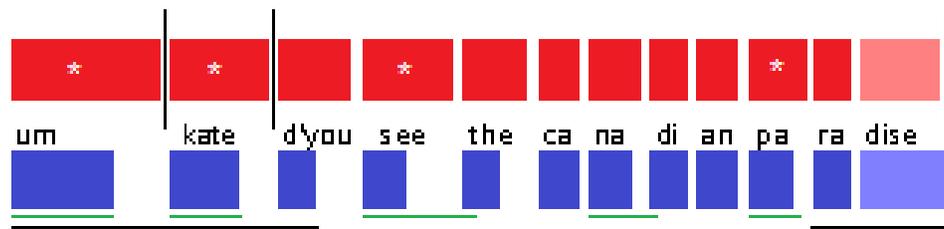


Figure 4-1: A diagram of of English Subject o's production of English Utterance 1 and the simulation of Utterance 1

As this figure shows, all of the syllables in English Subject o's production that "touch" (immediately precede or follow) a phrase boundary are under the influence of the phrase oscillator peaks in the simulation. Additionally, in the simulation, the penultimate syllable just touches the leading edge of the second phrase oscillator peak, but as the syllable durations show, that small influence is not enough to lengthen that syllable in the simulation, and correspondingly, the penultimate syllable in the real utterance is quite short (only one syllable is shorter). Regarding accenting, all of the syllables in the original production that have pitch accents are under accent oscillator peaks in the simulation. In the simulation there is one syllable (*pa*) under an accent oscillator peak which may not be accented in the real utterance; although *pa* could be considered "stressed" as the metrical head of the word *paradise*, in English Subject o's production it is not lengthened, nor does it contain a pitch accent.

Next I will discuss the simulation's ability to predict relative syllable durations, which may also be thought of as its ability to explain variation in syllable duration across the utterance. In Figures 4-2, 4-3, and 4-4, the durations of real and simulated syllables are presented as compared to a kind of null hypothesis, the null hypothesis in this case being that there is no variation in syllable duration across the utterance. The null hypothesis is presented as the baseline, zero on the y-axis, in each of these figures, with syllables diverging from the baseline in either the positive (longer) or negative (shorter)

direction. Figure 4-3 compares the relative length of syllables in the original utterance and the simulation.

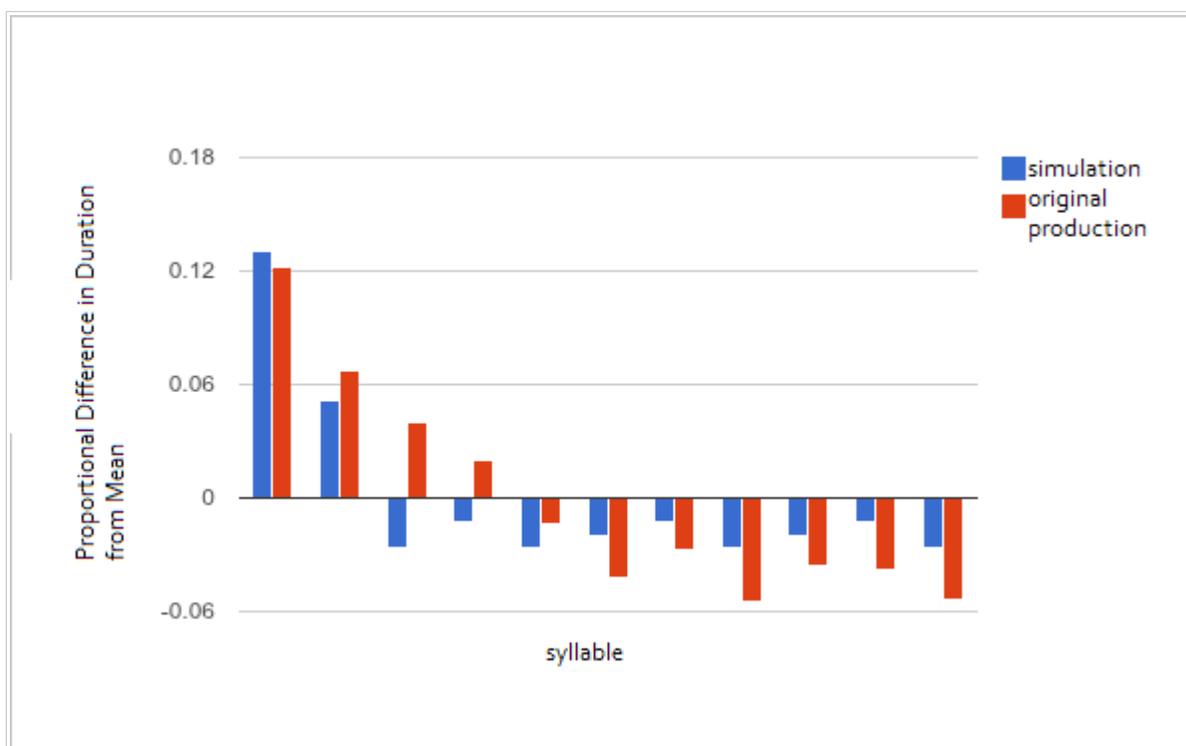


Figure 4-2: Proportional differences in duration of each syllable from the mean – English Subject O's production of English Utterance 1 vs. Simulated English Utterance 1

This figure shows the relative length of syllables as compared to the average length of all syllables in the utterance. Zero on the y-axis marks the average line, and bar height shows how much longer (positive on the y-axis) or shorter (negative) than average each syllable is. The simulation matches the general trend of shortening over time and the longer/shorter alternation of the syllables in the original, though somewhat underpredicts the shortness of the shortest syllables. The major difference between the two is on the syllables *d'you* and *see* (/dju/ or /dʒu/ and /si/), which are relatively long in the original utterance. Based on the declining trend, it is possible that the length of these two syllables is due to post-boundary lengthening, in which case the simulation has underpredicted the strength and/or scope of the phrase boundary, but it is also possible that these syllables are longer than average for some other reason, such

as the length of their segments or an idiosyncrasy in English Subject o's production, which the simulation could not predict.

4.1.3.2 Utterance 1 Simulation Compared to All Utterance 1 Productions

Having discussed the ability of the simulation of Utterance 1 to model the original production of Utterance 1, I will now discuss the ability of the simulation to model the utterance as produced by all speakers. First, Figure 4-4 shows the raw durations of each syllable in the utterance for the original production by English Subject o, the mean duration of each syllable over all productions of the utterance, and the duration of each syllable in the simulation.

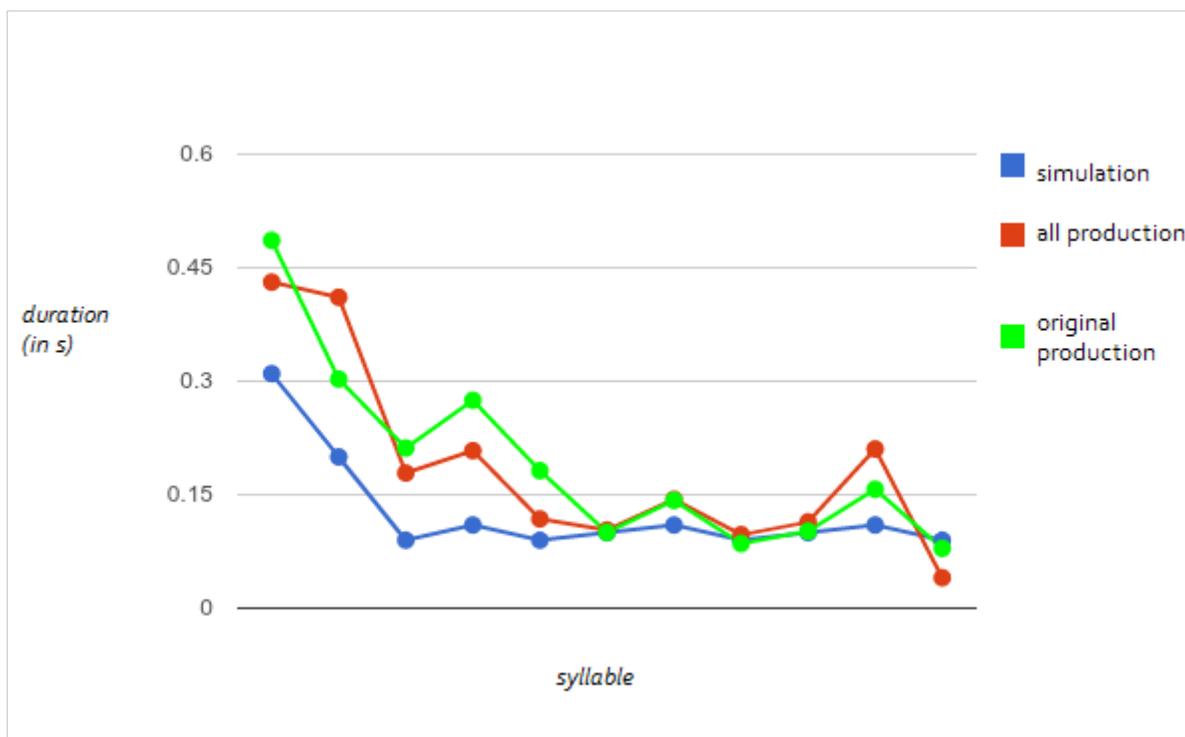


Figure 4-3: Syllable durations, English Utterance 1

As Figure 4-4 shows, the simulation follows the pattern of longer and shorter syllables in the data, but tends to underestimate the length of the longer syllables.

Next I will discuss the ability of the simulation to predict relative syllable

durations across all productions of the utterance. Figure 4-5 shows how well the simulation of Utterance 1 predicts the location of lengthened syllables over all productions of Utterance 1 by evaluating the duration of syllables in the simulated utterance as longer/shorter than the average duration of the corresponding syllable in all imitated utterances. The blue line shows which of the simulated syllables are longer or shorter than average (here only a binary distinction is made: a value above zero is longer, while a value below zero is lower). The red bars show for each syllable the proportion of imitated productions in which that syllable is longer or shorter than average. The less consistent the duration of the syllable is relative to the average across productions, the smaller the bar will be; if a given syllable is longer than average in exactly half of the productions and shorter in half, its bar length would be zero.

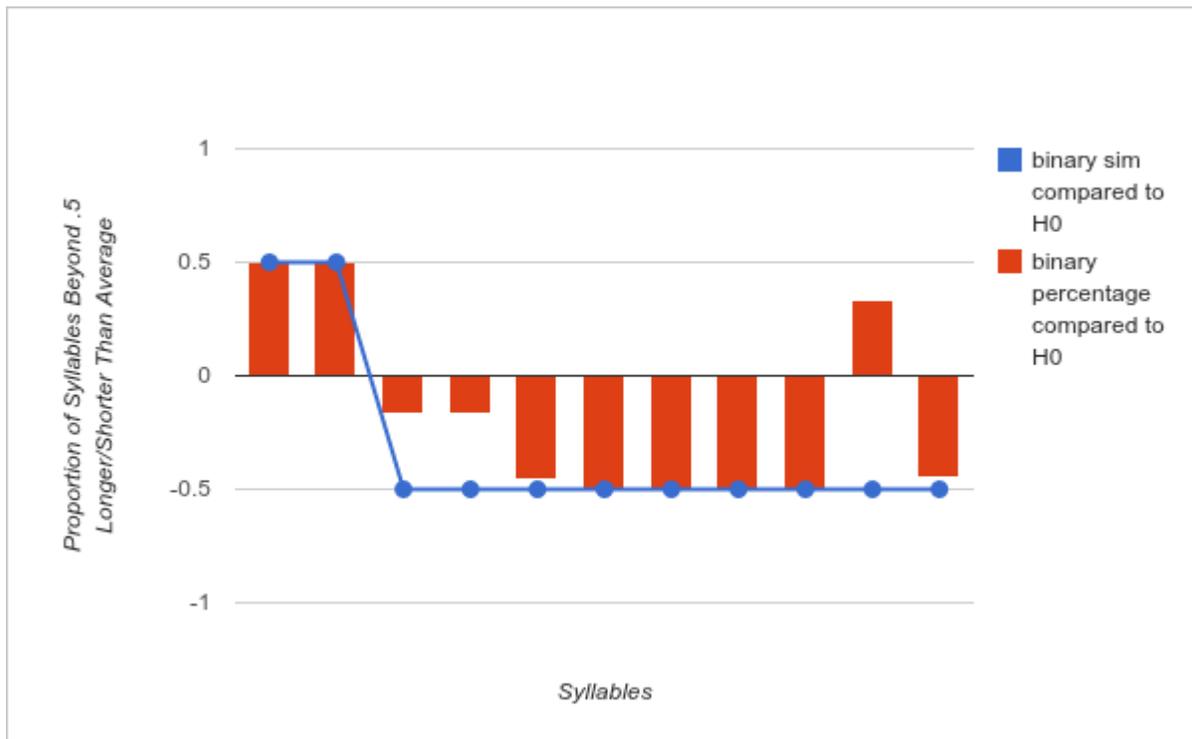


Figure 4-4: Proportion of syllable durations longer or shorter than mean – all productions of English Utterance 1 vs. Simulated English Utterance 1

The simulation correctly predicts the location of lengthened and shortened syllables across all productions of Utterance 1 for every syllable save one. Interestingly,

the penultimate syllable *pa* is lengthened in a majority of imitative repetitions. As noted in section 4.2.1, the presence of an accent oscillator peak on *pa* was one difference between the Utterance 1 simulation and the original production it was based on. Although English Subject o’s production of Utterance 1 did not show lengthening or a pitch accent on *pa*, that syllable was longer than average in 83% of the productions of Utterance 1. This suggests that speakers producing this utterance may be likely to accent this syllable, as the simulation predicts.

While Figure 4-4 shows the likelihood that each syllable will be long or short among all productions of Utterance 1, it doesn’t take into account the degree of variation in duration. Figure 4-5 compares the average variation of each syllable’s duration from the mean across all productions of Utterance 1 to its variation from the mean in the simulation.

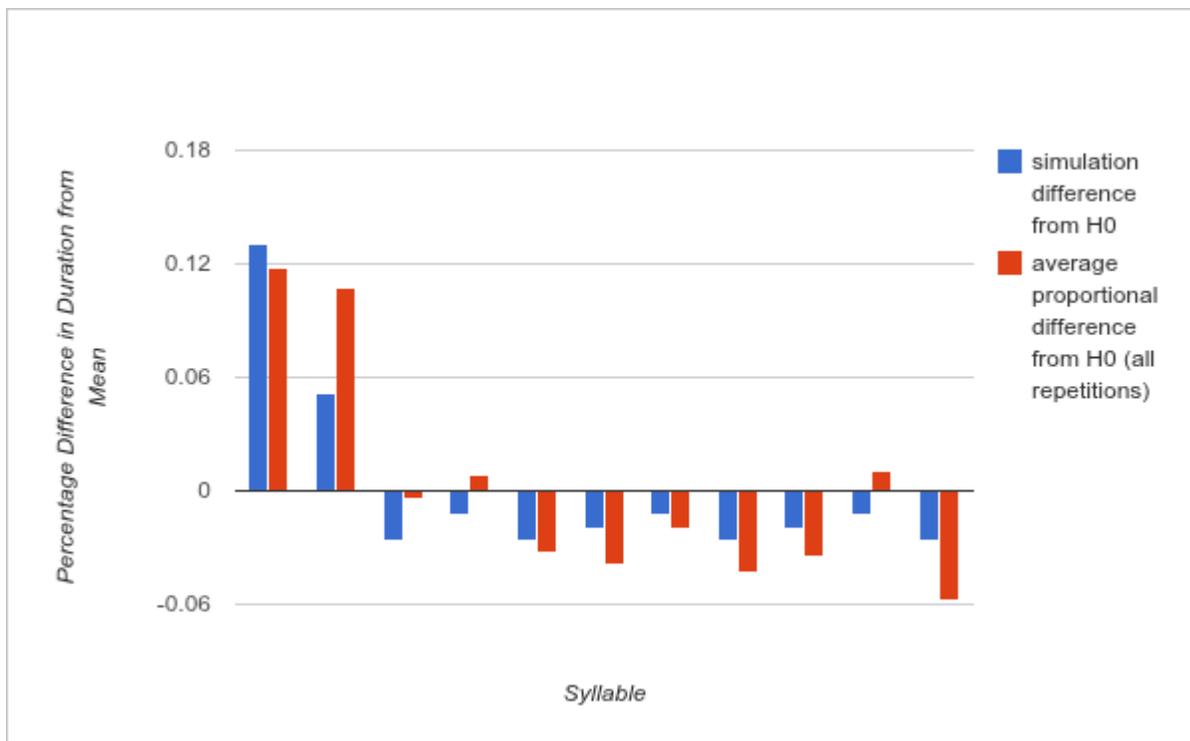


Figure 4-5: Proportional differences in duration of each syllable from the mean – all productions of English Utterance 1 vs. Simulated English Utterance 1

As Figure 4-5 shows, the simulation predicts variation in syllable durations across all productions of Utterance 1 even better perhaps than it predicts variation in the original utterance it was based on. The average duration variance from the mean is more similar to the simulation than that in the original utterance in 8/11 syllables (73%). The most significant divergence of the means from the simulation is the second syllable, *kate* (/kejt/), with that syllable being much more lengthened on average across all productions than in either the original utterance or the simulation.

4.1.4 Utterance 2

4.1.4.1 Utterance 2 Simulation Compared to Utterance 2 Original Production

Figure 4-6 shows an illustration of utterance 2 as spontaneously produced by English Subject 0 during a map task, and its simulation. As in Figure 4-1, syllable durations are represented by the length of the red and blue bars, the locations of phrase boundaries in the original production are noted by black vertical lines, and accented syllables in the original are marked with asterisks. The locations of peaks of the accent nodes are marked by green horizontal lines, and the locations of peaks of the phrase node are marked by black horizontal lines.

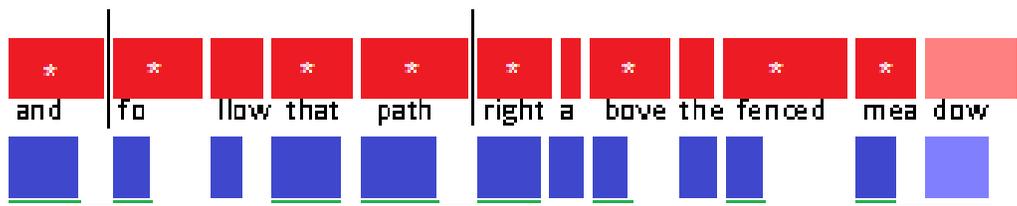


Figure 4-6: A diagram of of English Subject 0's production of English Utterance 2 and the simulation of Utterance 2

As this figure shows, each syllable that touches (immediately precedes or follows) a phrase boundary in the original production touches a phrase oscillator peak in the simulation. The final phrase oscillator peak in the simulated utterance also covers the

penultimate syllable, which of course does not directly “touch” a phrase boundary in the original production, but as studies of pre-boundary lengthening in English show (e.g. Shattuck-Hufnagel & Turk, 1998), could be influenced by the phrase boundary following the final syllable. The simulation also correctly identifies 8 of the 9 accented syllables and 2 of the 3 unaccented syllables, incorrectly placing the accent oscillator peak only in the final word where it peaks on the second syllable rather than the first.

As with Utterance 1, I will next discuss the ability of the simulation of Utterance 2 to predict relative syllable durations, or the variation in syllable durations across the utterance. As in the previous section, zero on the y-axis in each of these figures represents the null hypothesis that all syllables are of the same duration (they do not vary from the mean), with simulated and observed syllable durations diverging from the mean in either the positive (longer) or negative (shorter) direction. Figure 4-7 compares the relative length of syllables in the original utterance and the simulation.

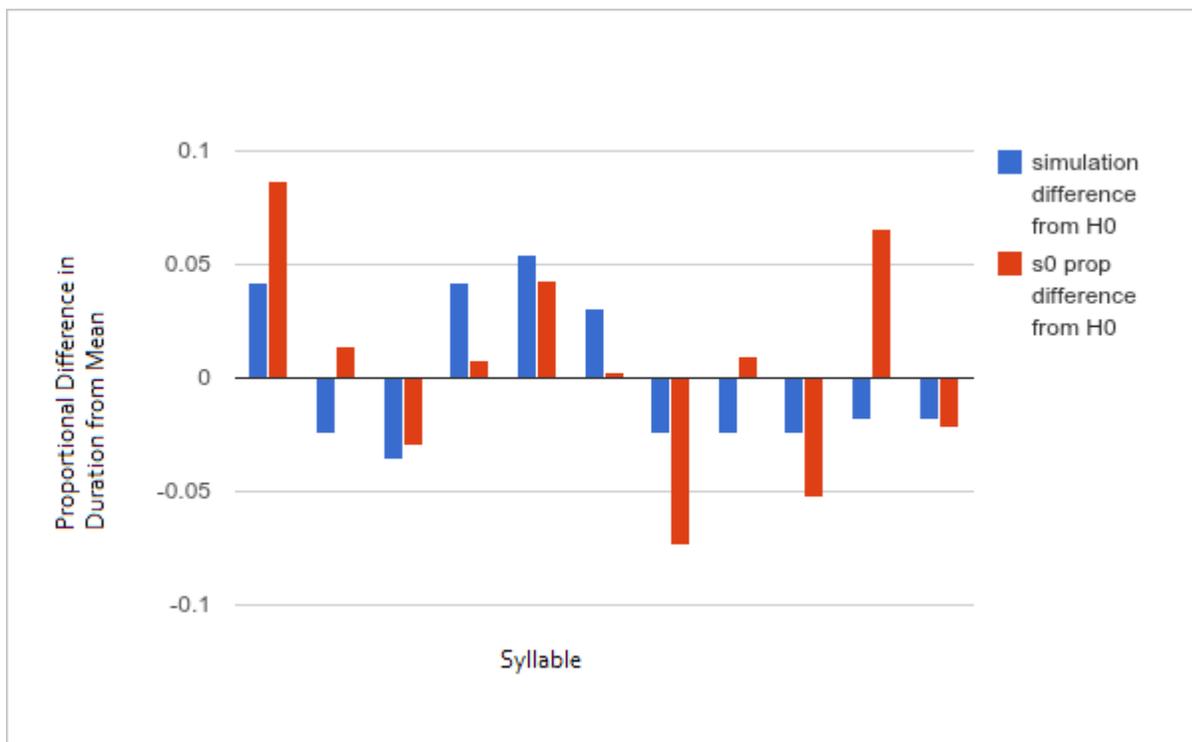


Figure 4-7: Proportional differences in duration of each syllable from the mean – English Subject o's production of English Utterance 2 vs. Simulated English Utterance 2

As Figure 4-7 shows, the original production of Utterance 2 has a very different pattern of long and short syllables from Utterance 1. Utterance 1's syllables generally decline in length over time; the early syllables (the first 4 in English Subject o's production, the first 2 on the average) are longer than average, and the following syllables are all near or below average length. The syllable durations in the original production of Utterance 2 vary much more over time, and the simulation does not track those variations as clearly. The simulation does follow the utterance in the initial decline in duration across the first 3 syllables, an increase in the fourth and fifth syllables, and a decrease after that. The most significant difference between the simulation and the production is at the penultimate syllable, *fenced* (/fenst/). This syllable illustrates what may be behind some of the differences between the simulation and production: *fenced* is the syllable in Utterance 2 with the most segments, a variable that NOMSTR does not account for, and which could contribute to it being the second-longest syllable in the utterance, and the simulation far under-predicting its duration. Conversely, the seventh syllable, which contains the fewest segments, consisting of only a schwa (the first syllable in *above*), is the shortest syllable in the utterance, and its duration is overpredicted by the simulation.

4.1.4.2 Utterance 2 Simulation Compared to All Utterance 2 Productions

Comparisons between the simulation and the mean of all productions of Utterance 2 are presented in Figures 4-8, through 4-10. Figure 4-8 compares the raw duration of each simulated syllable to its mean duration across all productions of the utterance, as well as to its raw duration in English Subject o's original production. Figure 4-9 displays the syllables which are longer or shorter than average in simulated Utterance 2 and the proportion of productions of Utterance 2 in which that syllable is relatively long or short, while Figure 4-10 compares the relative duration of each syllable in simulated Utterance 2 to its average relative duration across all productions of the utterance.

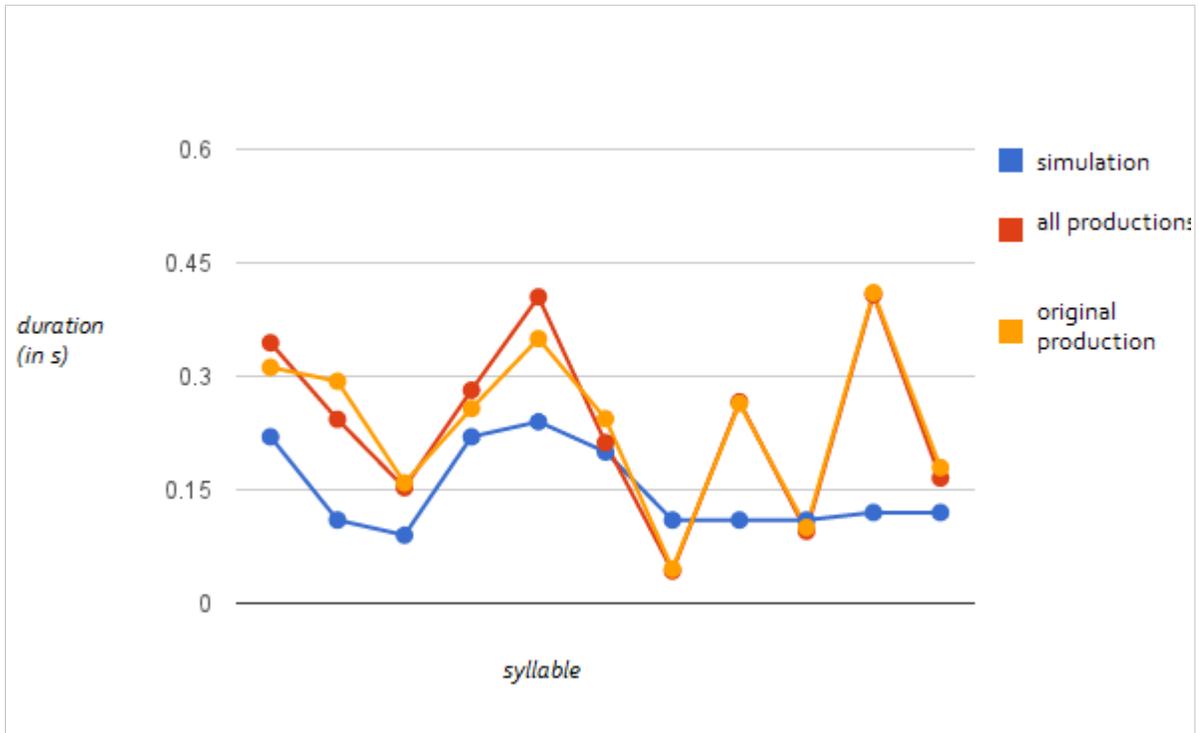


Figure 4-8: Syllable durations, English Utterance 2

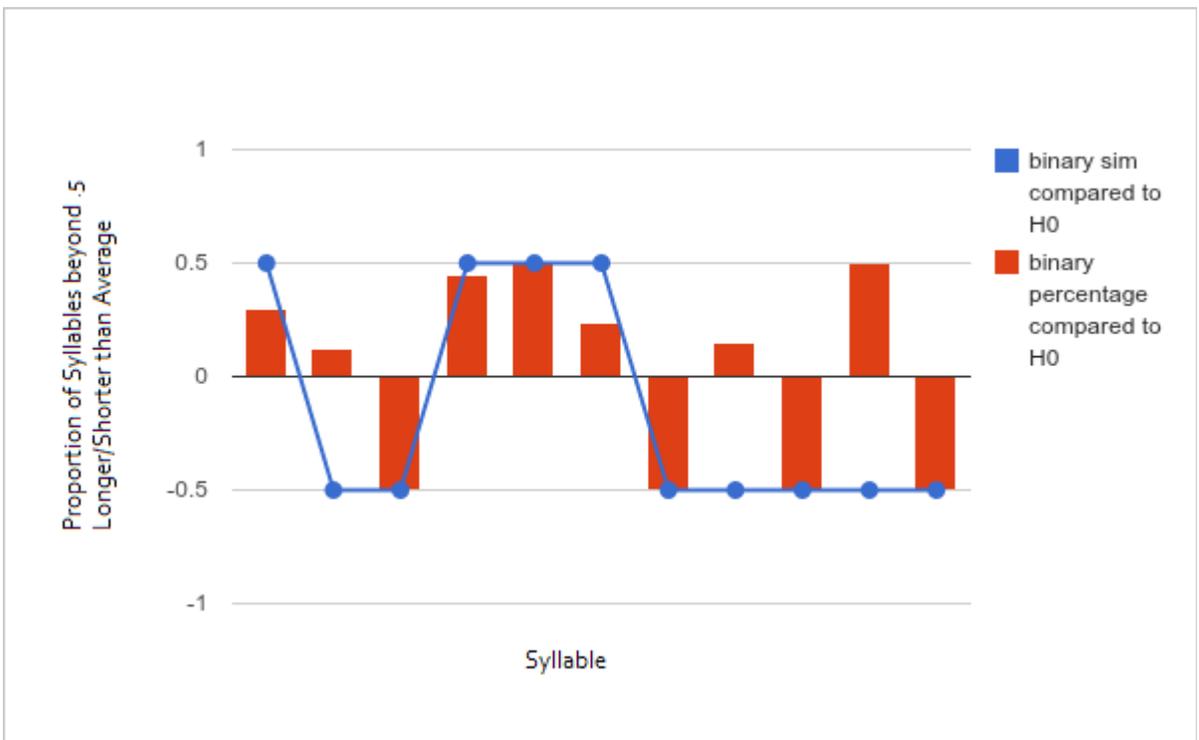


Figure 4-9: Proportion of syllable durations longer or shorter than mean – all productions of English Utterance 2 vs. Simulated English Utterance 2

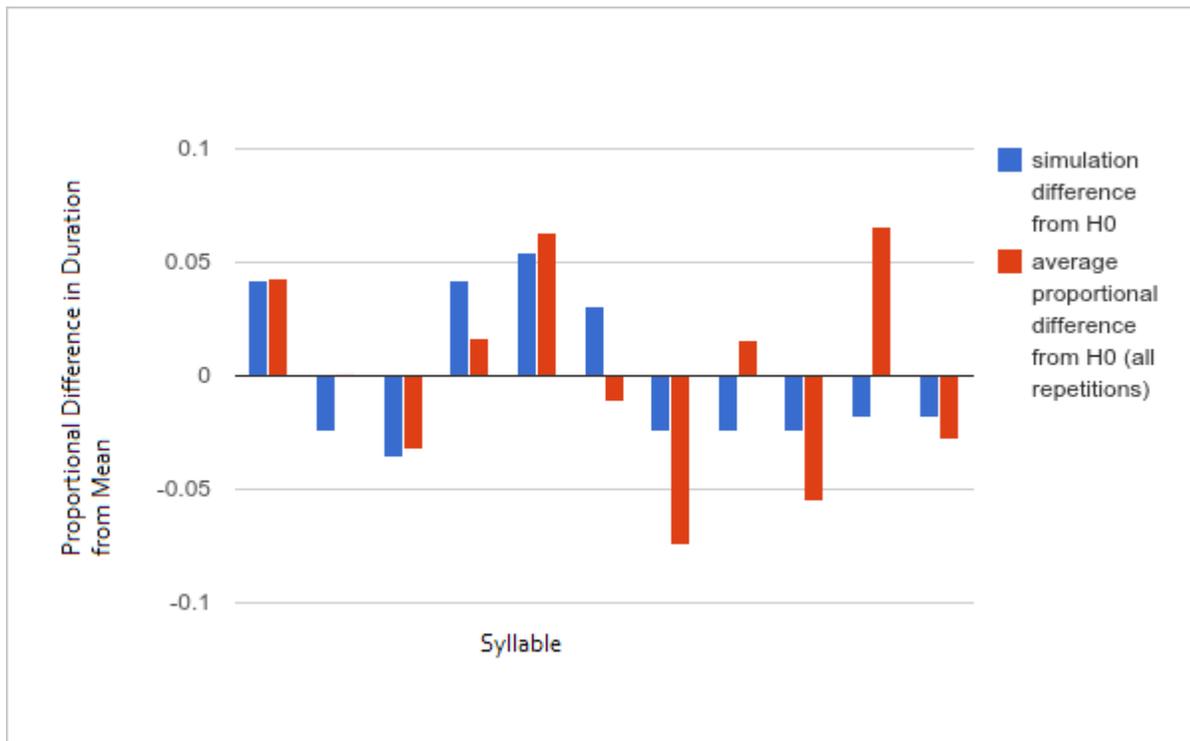


Figure 4-10: Proportional differences in duration of each syllable from the mean – all productions of English Utterance 2 vs. Simulated English Utterance 2

Figures 4-9 and 4-10 paint much the same picture as the comparison between the simulation and the original production. Figure 4-9 clarifies the relationship between the actual and simulated versions of Utterance 2: the simulation tracks the real syllable durations, although somewhat underpredicting them, except for the syllables /bʌv/ and /fɛnst/. Although these syllables are accented in the simulation as well as in the actual productions, they are much longer in the real productions than in the simulation.

4.1.5 Utterance 3

4.1.5.1 Utterance 3 Simulation Compared to Utterance 3 Original Production

Figure 4-11 is an illustration of the syllable durations and prosodic structures of the original production of Utterance 3 and its simulation.

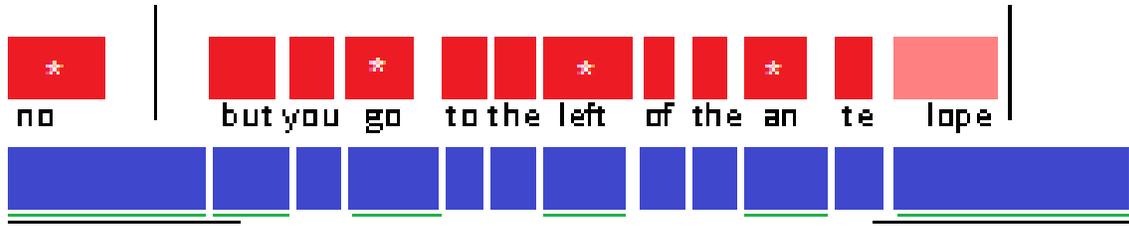


Figure 4-11: A diagram of of English Subject o's production of English Utterance 3 and the simulation of Utterance 3

As shown in Figure 4-11, the simulation correctly predicts the locations of the phrase boundaries; both syllables which in the original production precede a phrase boundary are covered by a phrase oscillator peak in the simulation, as well as the second syllable, which immediately follows a phrase boundary in English Subject o's production. The simulation correctly predicts an accent oscillator peak on all 4 of the syllables accented in the original production, and correctly predicts no accenting on 6 of the 8 unaccented syllables in the original production (the second and last syllables being the exceptions).

Figure 4-12 compares the relative length of syllables in the original utterance and the simulation.

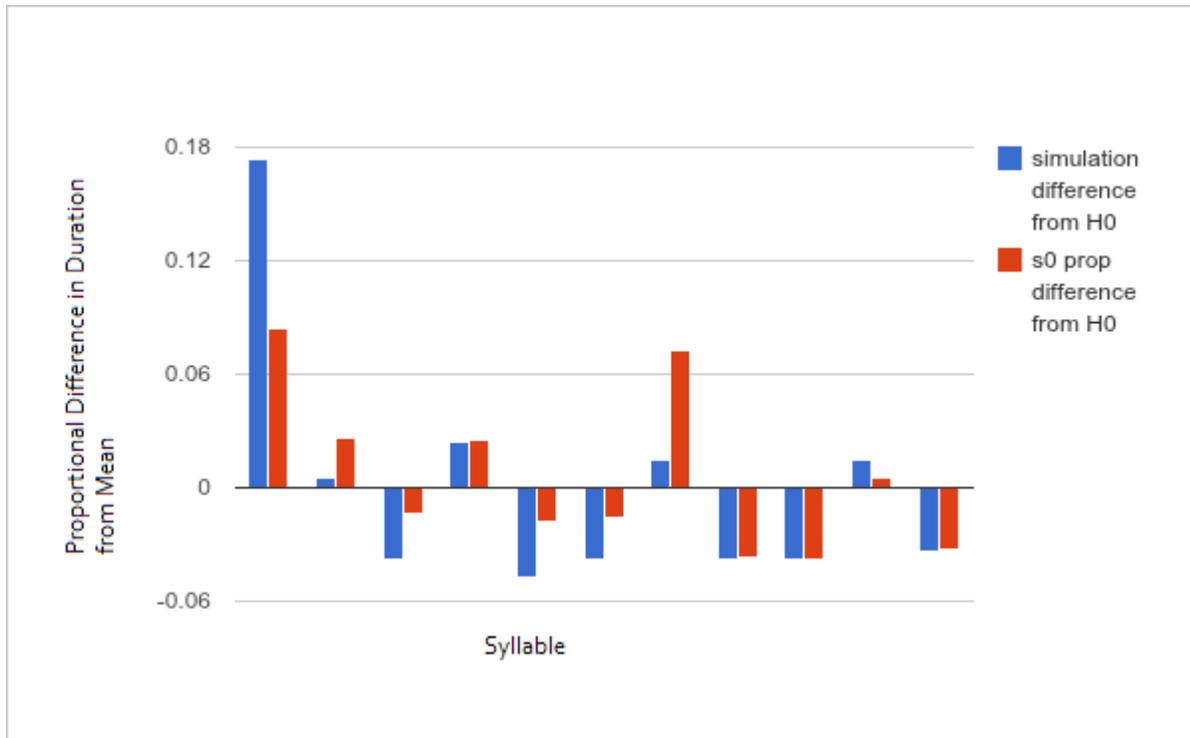


Figure 4-12: Proportional differences in duration of each syllable from the mean – English Subject o's production of English Utterance 3 vs. Simulated English Utterance 3

Where Utterance 1 follows a shortening trend over time, and Utterance 2 had a few very long and a few very short syllables, English Subject o's production of Utterance 3 alternates between syllables that are slightly longer and slightly shorter than average. The simulation of Utterance 3 tracks these alternations, and correctly predicts which syllables are longer and shorter than average. The major difference between the simulation and English Subject o's production is that the simulation overpredicts the length of the first syllable and underpredicts the length of seventh syllable. The length of the seventh syllable in English Subject o's production may be another example of the influence of syllable content- that syllable is *left* (/left/) has the most segments out of the utterance's syllables.

4.1.5.2 Utterance 3 Simulation Compared to All Utterance 3 Productions

Comparisons between the simulation and the mean of all productions of Utterance 3 are presented in Figures 4-13 through 4-15. Figure 4-13 compares the raw

duration of each simulated syllable to its mean duration across all productions of the utterance, as well as to its raw duration in English Subject o's original production. Figure 4-14 displays the syllables which are longer or shorter than average in simulated Utterance 3 and the proportion of productions of Utterance 3 in which that syllable is relatively long or short, while Figure 4-15 compares the relative duration of each syllable in simulated Utterance 3 to its average relative duration across all productions of the utterance.

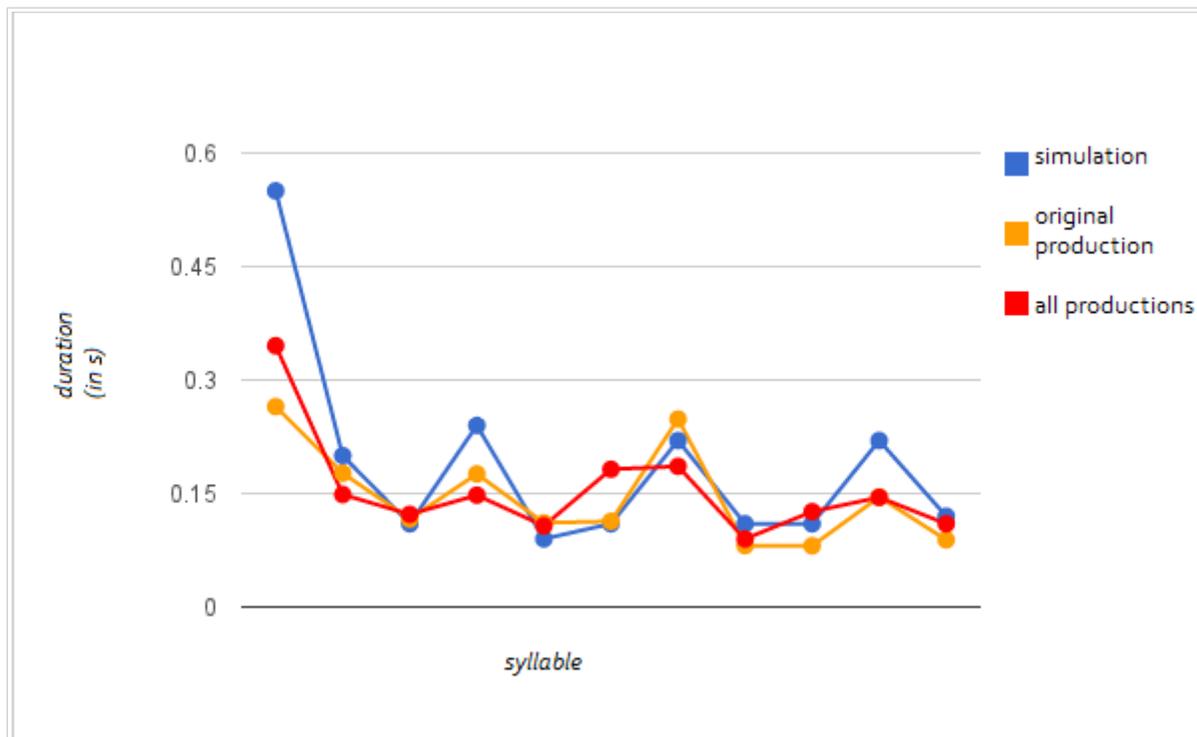


Figure 4-13: Syllable durations, English Utterance 3

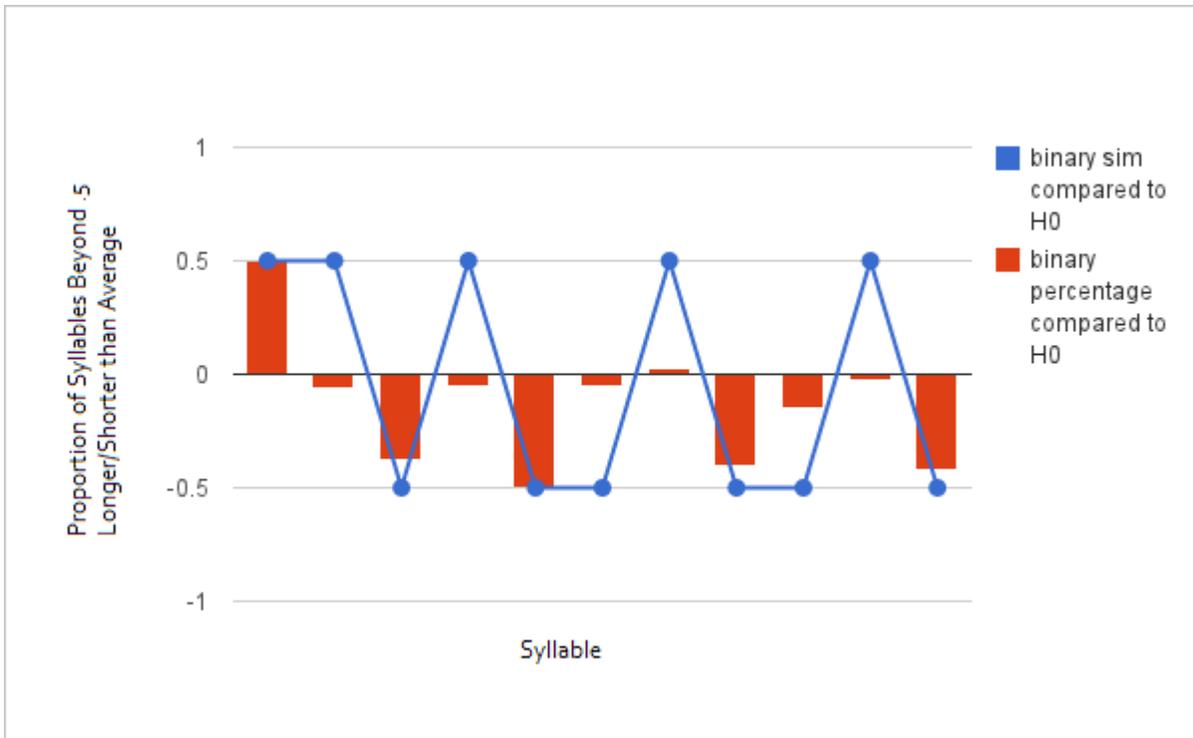


Figure 4-14: Proportion of syllable durations longer or shorter than mean – all productions of English Utterance 3 vs. Simulated English Utterance 3

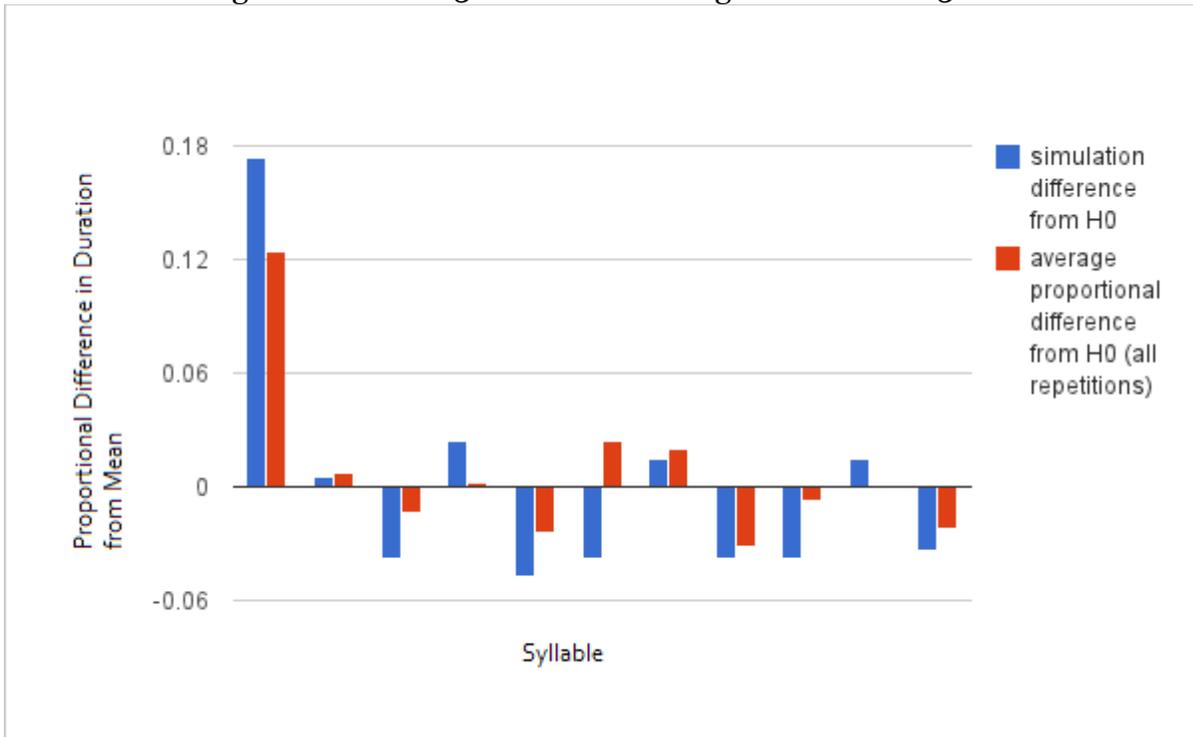


Figure 4-15: Proportional differences in duration of each syllable from the mean – all productions of English Utterance 3 vs. Simulated English Utterance 3

When syllable durations are averaged across all productions of Utterance 3, the differences between the simulation and the data in the first and especially the seventh syllable are reduced, although the simulation still overpredicts the lengthening of the first syllable. As Figures 4-13 and 4-14 show, the simulation tracks the increases and decreases in syllable durations well, with the possible exception of the sixth syllable, *the* (/ðə/).

4.1.6 Do English phrases become more regular with more repetition?

This section compares the temporal structure of the spontaneous utterances 1, 2, and 3 with the same utterances across multiple repetitions. As mentioned in Chapter 2, although the strict isochrony hypothesis has been debunked, studies on speech cycling and synchronous speech have shown that prosodic timing may naturally tend toward regularity, or predictability. The goal of comparing spontaneous and repeated utterances is to use repetition to filter out influences on the temporal structure of a phrase other than those that can be attributed to the underlying speech production system, such as pauses and disfluencies. The repetition task also allows the speakers maximum planning time to organize the temporal structure of the utterance. The simulations produced by NOMSTR are perfectly temporally regular (that is, predictable), and so provide a way to measure the regularity of the utterances' prosodic timing that takes into account the complex prosodic structure of an utterance, which simple isochrony measures do not. If the prosodic timing of an utterance becomes more regular as it is repeated more, the prosodic timing of the third repetition of that utterance should be more similar to the simulation than the first.

4.1.6.1 Methods

As mentioned in section 4.1.2, the nature of the simulations and data used in this experiment make it difficult to use a single metric or statistic to measure the overall difference between the simulated utterances and their actual productions. Because of the way the accent and phrase oscillators both influence syllable durations, in this study I use the difference in proportional syllable durations as a way of approximating the

overall difference between the simulated and actual utterances.²

4.1.6.2 Results

Figure 4-16 shows the overall difference between each simulated utterance and all first and third repetitions of that same utterance. A smaller residual indicates more similarity to simulation.

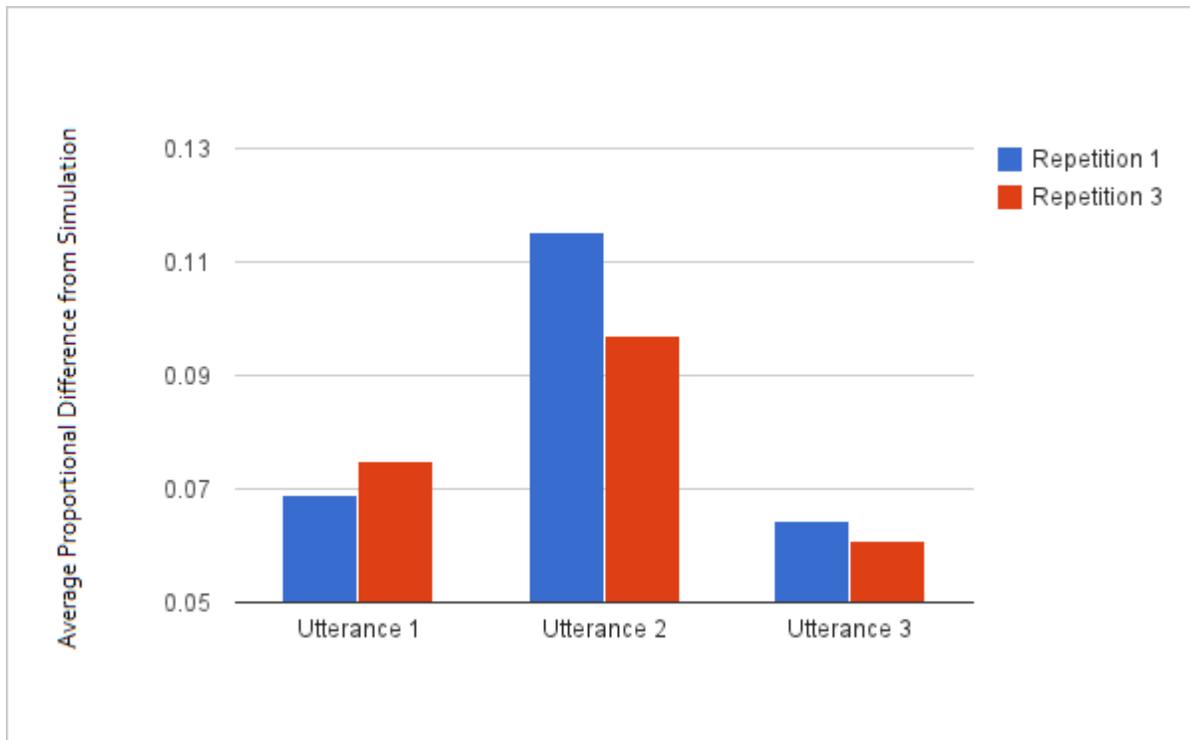


Figure 4-16: Average residual difference in syllable durations from simulation, English

The third repetitions of utterances 2 and 3 do have smaller residuals overall than the first repetitions, but paired t-tests showed no significant differences in the residuals of the 1st and 3rd repetitions for any of the utterances ($t(9)=-0.856$, $p=0.414$; $t(8)=0.324$, $p=0.754$; $t(9)=0.161$, $p=0.875$). This could be due to the fact that the prosodic timing of these utterances does not become more regular with more repetition, or that the difference over only three repetitions is not large enough to be significant. A future study involving more repetitions of each utterance could clarify this result.

² Measuring raw syllable duration residuals instead of proportional produced the same results.

4.2 French

4.2.1 Procedure

The French condition of this experiment uses three utterances from the Rhapsodie reference prosodic corpus of spoken French (Rhapsodie 2010). Each utterance was originally produced spontaneously during a direction-giving task; the original speaker who spontaneously produced each utterance is referred to here as French Subject 0. For the present study, 10 speakers were recorded in a sound-dampened booth listening to recordings of the original utterances and repeating each 3 times, with short pauses between each repetition to clear the auditory buffer. For these utterances, I recorded syllable durations (as measured from one c-center to the next (c.f. Browman and Goldstein 1988)), and the locations of accented syllables and phrase boundaries, as identified in the corpus. For each utterance, these three measurements were used as the starting points for setting the three wavelength parameters in the model. The wavelength of the short oscillator was set as the average duration of all non-accented syllables in an utterance, the wavelength of the medium oscillator was set as the average duration of all inter-accent intervals in an utterance, and the wavelength of the long oscillator was set as the average duration of all phrases in an utterance. Next the phases, amplitudes, and wavelengths of the oscillators for each utterance were adjusted so that the simulation more closely matched the utterance data (values of model parameters used to simulate each utterance can be seen in Appendix B.3). In the next section I describe how I compared the simulations to the data and evaluated how well each simulation models its utterance.

4.2.2 Analyses

As with the English version of this study, I will compare the simulation of each spontaneous utterance to a null model which is constructed using the information from each actual utterance that is fed into NOMSTR in order to produce the simulation of that utterance, to measure its ability to simulate the temporal pattern of that utterance - the relative positions of metrical elements in the utterance and their effects on each other. I will then evaluate the ability of each simulation to model its utterance's

underlying metrical structure, and not just the original speaker's idiosyncratic production of the utterance by comparing each simulation with to all repetitions of the utterance it is simulating.

4.2.3 Utterance 1

4.2.3.1 Utterance 1 Simulation Compared to Utterance 1 Original Production

Figure 4-17 is an illustration of the syllable durations and prosodic structures of the original production of Utterance 1 and its simulation.

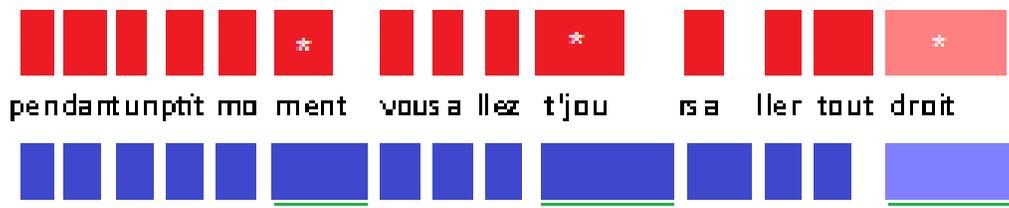


Figure 4-17: A diagram of the syllable durations and prosodic structures of French Subject O's production of French Utterance 1 and the simulation of Utterance 1

In Figures 4-18 through 4-21, the durations of real and simulated syllables are presented as compared to the null hypothesis that there is no variation in syllable duration across the utterance. The null hypothesis is presented as the baseline, zero on the y-axis, in each of these figures, with syllable diverging from the baseline in either the positive (longer) or negative (shorter) direction. Figure 4-18 compares the relative length of syllables in the original utterance and the simulation.

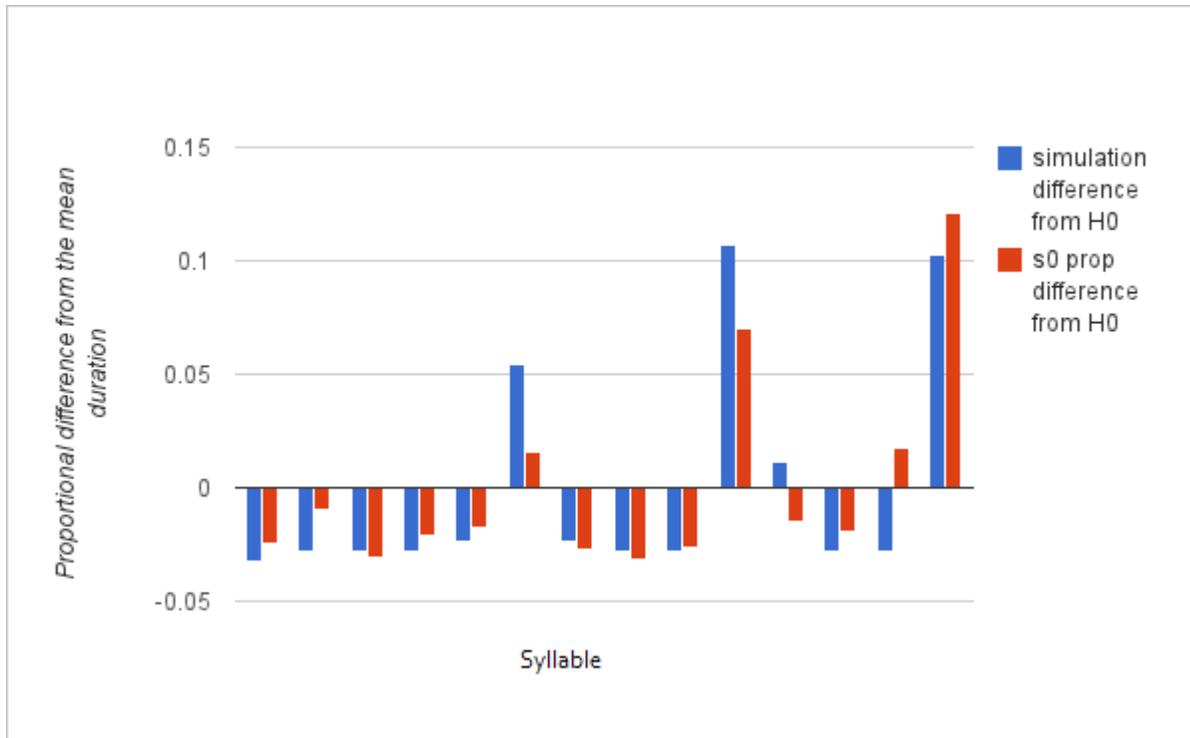


Figure 4-18: Proportional differences in duration of each syllable from the mean – French Subject O's production of French Utterance 1 vs. Simulated French Utterance 1

As with the English utterances, the simulation generally tracks the location of the long and short syllables in the original utterance, with a few differences, namely overestimating the degree of lengthening on the 6th and 10th syllables, and missing the lengthening on the penultimate syllable. Overall the simulation correctly predicts 75% (3 out of 4) lengthened syllables, and 90% of the shorter-than-average syllables.

4.2.3.2 Utterance 1 Simulation Compared to All Utterance 1 Productions

Figure 4-19 shows the raw durations of each syllable in the utterance for the original production by French Subject O, the mean duration of each syllable over all productions of the utterance, and the duration of each syllable in the simulation.

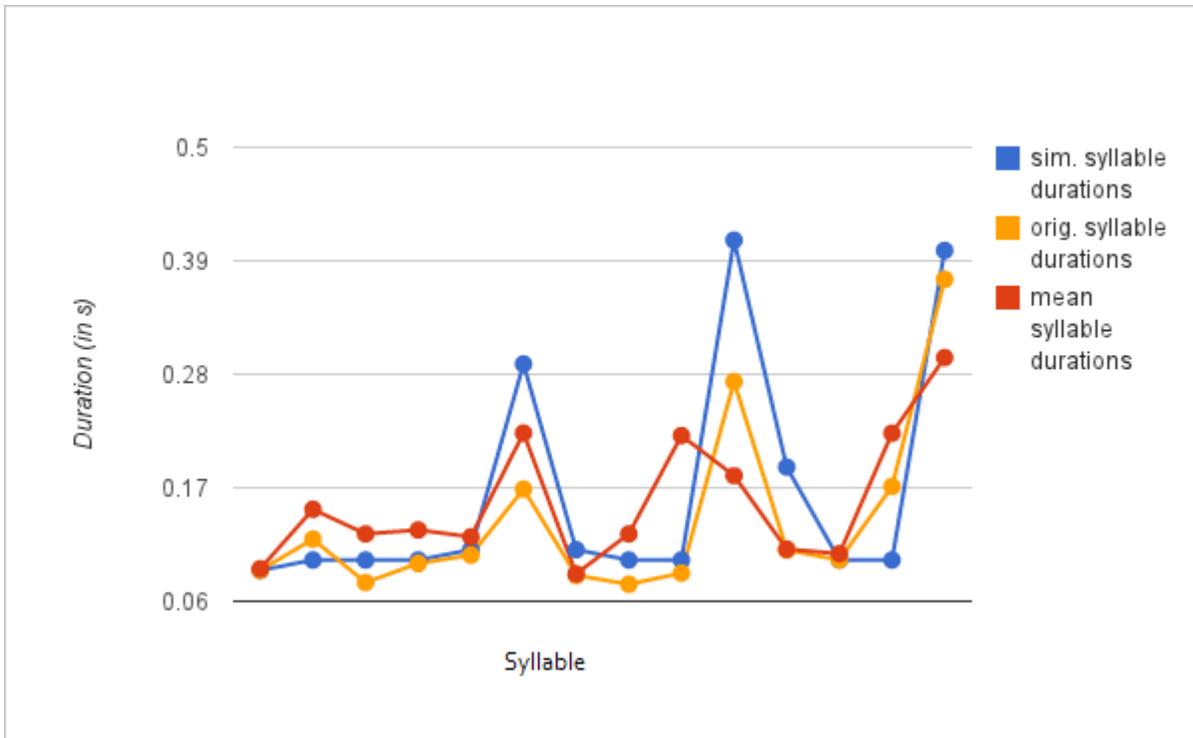


Figure 4-19: Syllable durations, French Utterance 1

Figure 4-20 shows the ability of the simulation to predict the location of lengthened syllables over all productions of Utterance 1 by looking at syllable duration in a long/short binary. The line shows which of the simulated syllables are longer or shorter than average, while the bars show which proportion of each syllable across all productions is longer or shorter than average.

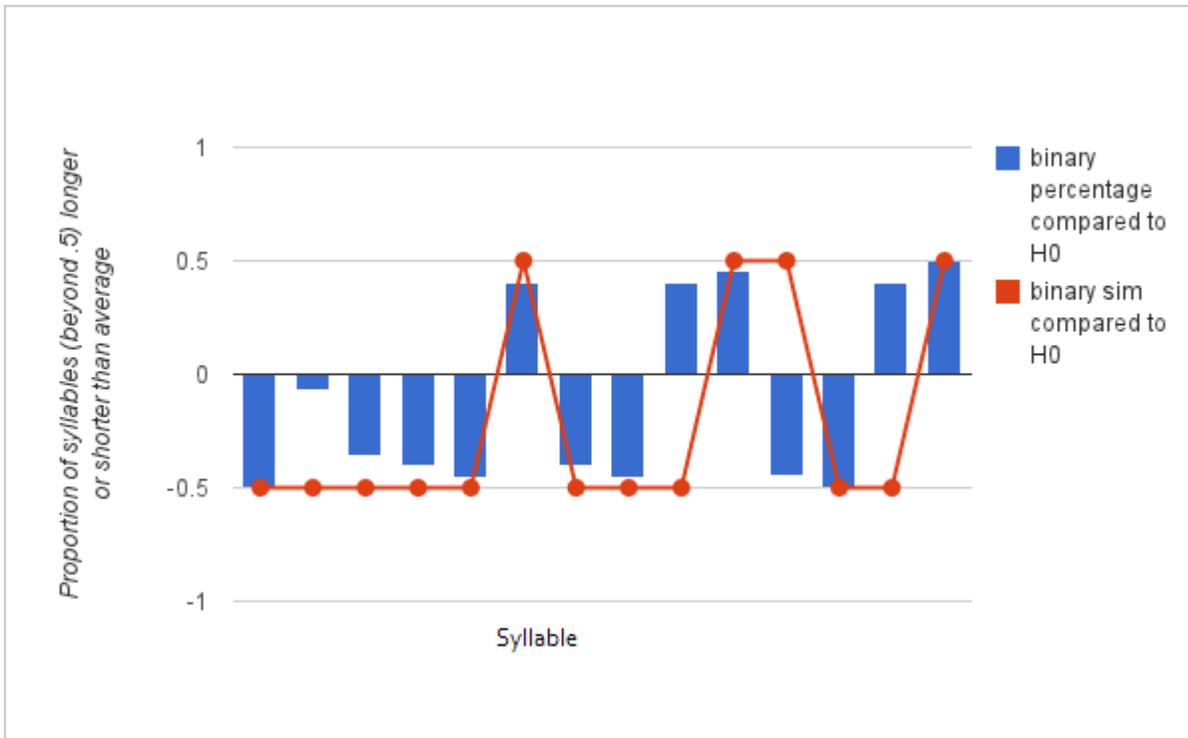


Figure 4-20: Proportion of syllable durations longer or shorter than mean – all productions of French Utterance 1 vs. Simulated French Utterance 1

In this figure, a slight mismatch can be seen between the syllable duration patterns in the simulation and the actual productions of the utterance. The simulation’s pattern of longer and shorter syllables lags slightly behind the pattern seen in the actual productions starting around syllable 9. This slight lag is likely due to NOMSTR’s difficulty in simulating both the relatively long stretch of unlengthened syllables at the beginning of the utterance with the comparatively short spans between lengthened syllables in the second half of the utterance.

Figure 4-21 compares the average variation of each syllable’s duration from the mean across all productions of Utterance 1 to its variation from the mean in the simulation, showing the degree to which each syllable is longer or shorter than expected, in the simulation and across all productions of the utterance.

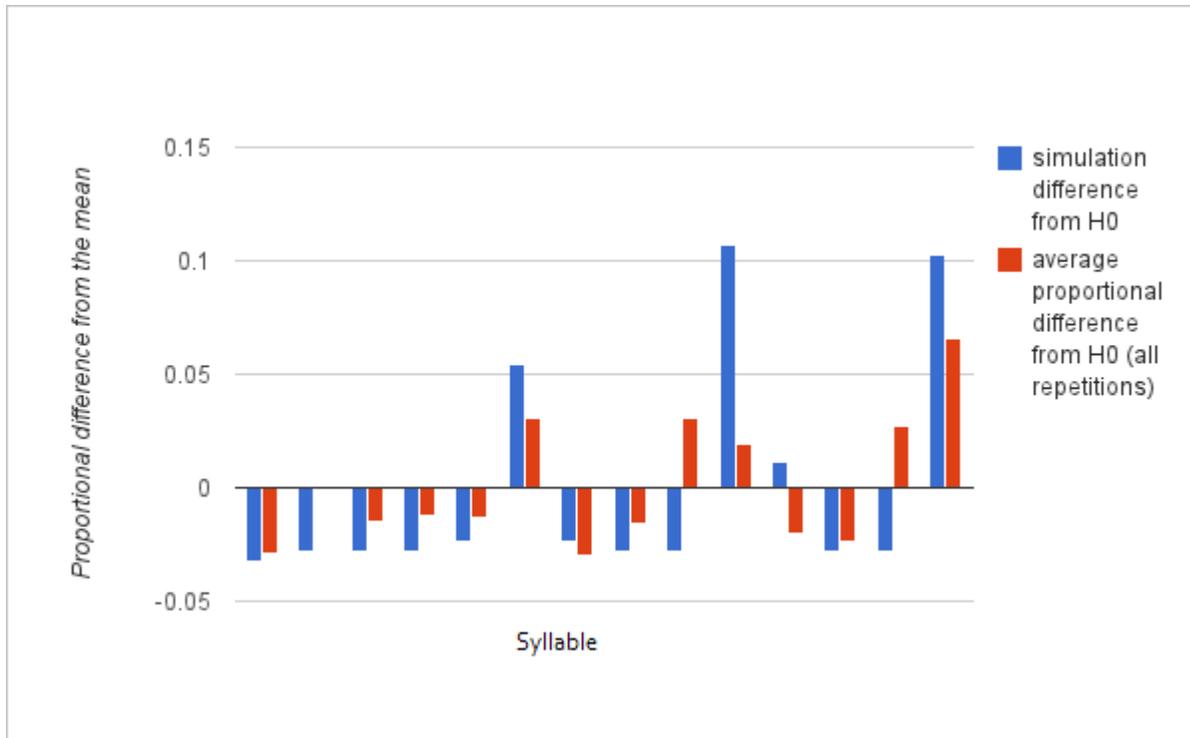


Figure 4-21: Proportional differences in duration of each syllable from the mean – average of all productions of French Utterance 2 vs. Simulated French Utterance 2

The pattern of syllable durations across all productions of Utterance 1 is very similar to that of the original production, except in syllables 9 and 10 (/le/ and /tʒu/). Here the mismatch in the location of lengthened syllables shown in Figure 4-22 can be seen, somewhat due to the fact that there seems to be a shifting of lengthening from the 10th syllable onto the 9th, as compared to the original production after which the simulation was modeled. The difference between the longest and shortest syllables is less extreme than in the original production as well, leading the simulation to overestimate the length of long syllable and underestimate the length of short ones.

The proportions plots show a very similar picture to the raw duration plot: overestimation of the length of lengthened syllables 6 and 10 by the simulation, as well as the pattern of lengthening and shortening in the second half of the utterance shifted about a syllable earlier over all the productions compared to the original and simulation.

4.2.4 Utterance 2

4.2.4.1 Utterance 2 Simulation Compared to Utterance 2 Original Production

Figure 4-22 is an illustration of the syllable durations and prosodic structures of the original production of Utterance 2 and its simulation.

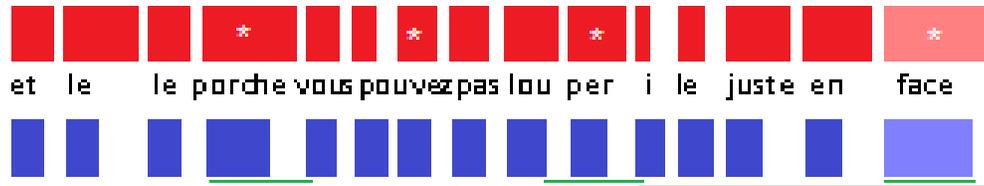


Figure 4-22: A diagram of of French Subject o's production of French Utterance 2 and the simulation of Utterance 2

I turn now to the ability of the simulation of Utterance 2 to predict relative syllable durations, or the variation in syllable durations across the utterance. As in the previous section, zero on the y-axis in each of the figures showing relative durations represents the null hypothesis that all syllables are of the same duration (they do not vary from the mean), with syllable diverging from the mean in either the positive (longer) or negative (shorter) direction. Figure 4-23 compares the relative length of syllables in the original utterance and the simulation.

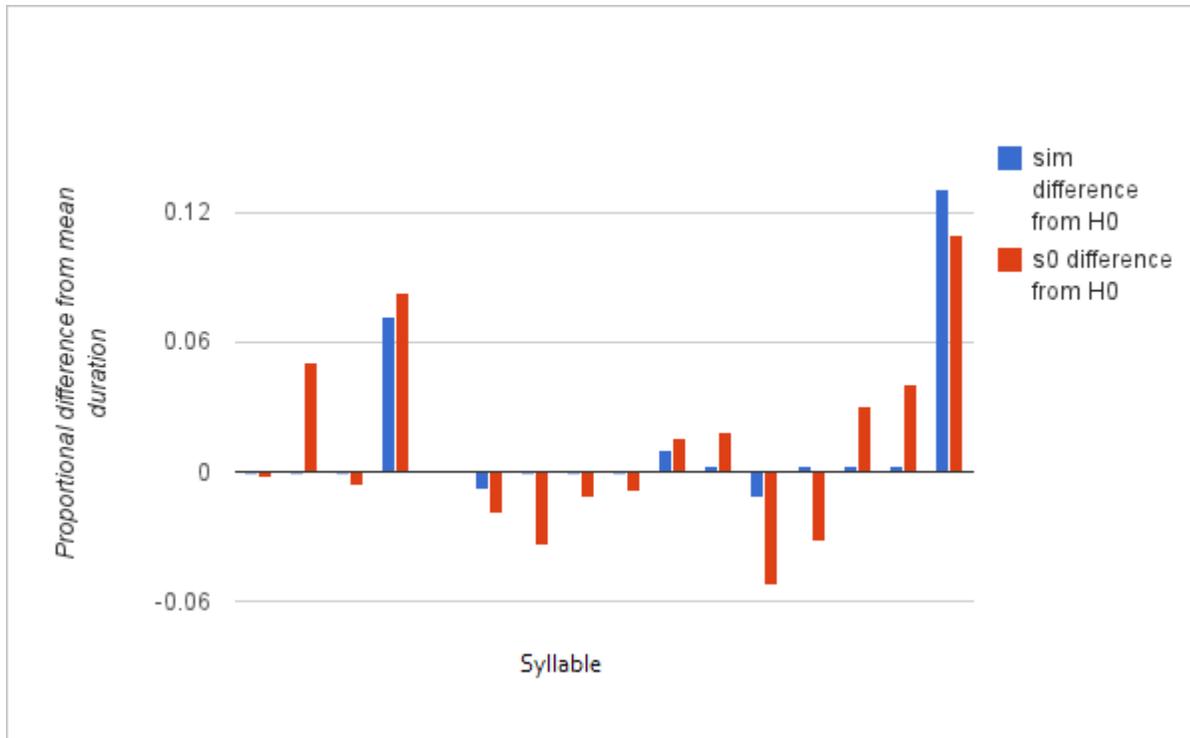


Figure 4-23: Proportional differences in duration of each syllable from the mean – French Subject o's production of French Utterance 2 vs. Simulated French Utterance 2

The syllables in Utterance 2 are much less consistent in their durations than those in Utterance 1, which means that the simulation, which has very regular timing, matches the actual utterance less well. The simulation does correctly predict the location of 85% (6 of 7) of the longer-than average syllables and 89% (8 of 9) of the shorter syllables, notably missing syllable 2, the first of two repetitions of the article *le*. This syllable in French Subject o's production of the utterance seems to be a disfluency, rather than part of the structure of the utterance, which of course NOMSTR cannot simulate. Interestingly, the repetitions of Utterance 2 by the some of the other subjects do not contain this long *le*, which I will discuss further in the next section. Overall, the simulation of this utterance underestimates the degree to which the syllable durations of the original production vary.

4.2.4.2 Utterance 2 Simulation Compared to All Utterance 2 Productions

Comparisons between the simulation and the mean of all productions of Utterance 2 are presented in Figures 4-24 through 4-26. Figure 4-28 compares the raw duration of each simulated syllable to its mean duration across all productions of the utterance, as well as to its raw duration in French Subject o's original production. Figure 4-24 displays the syllables which are longer or shorter than average in simulated Utterance 2 and the proportion of productions of Utterance 2 in which that syllable is relatively long or short, while Figure 4-26 compares the relative duration of each syllable in simulated Utterance 2 to its average relative duration across all productions of the utterance.

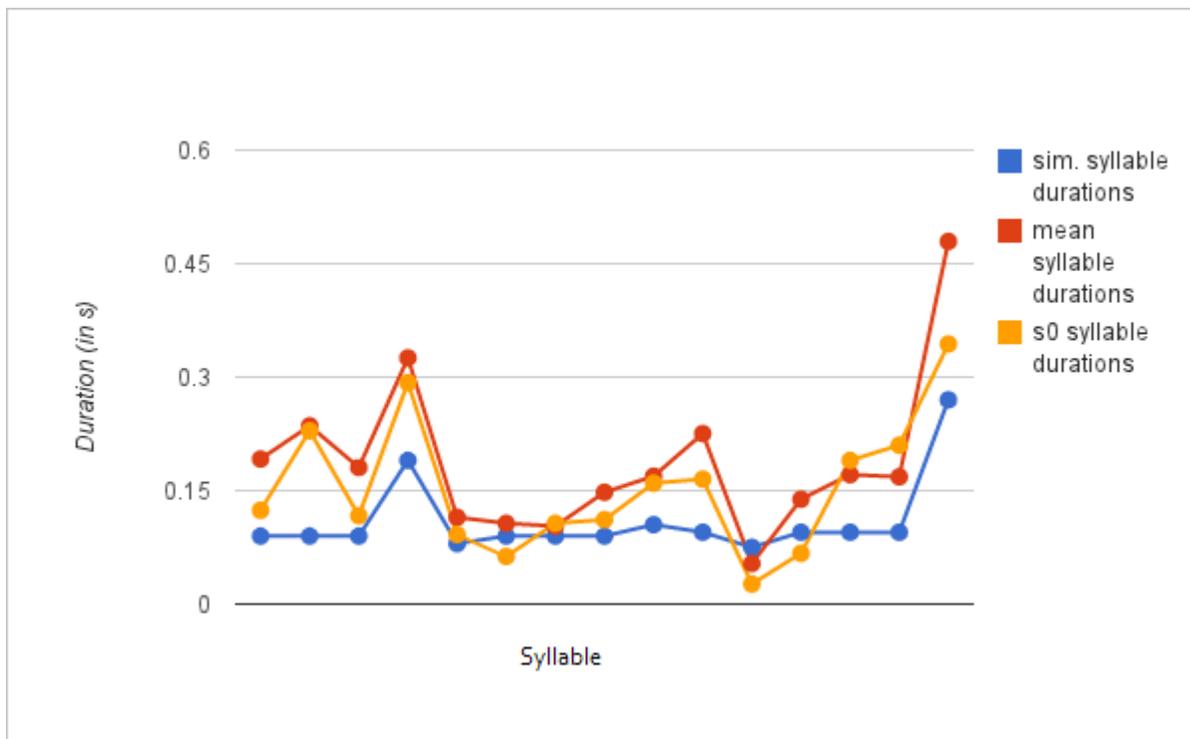


Figure 4-24: Syllable durations, French Utterance 2

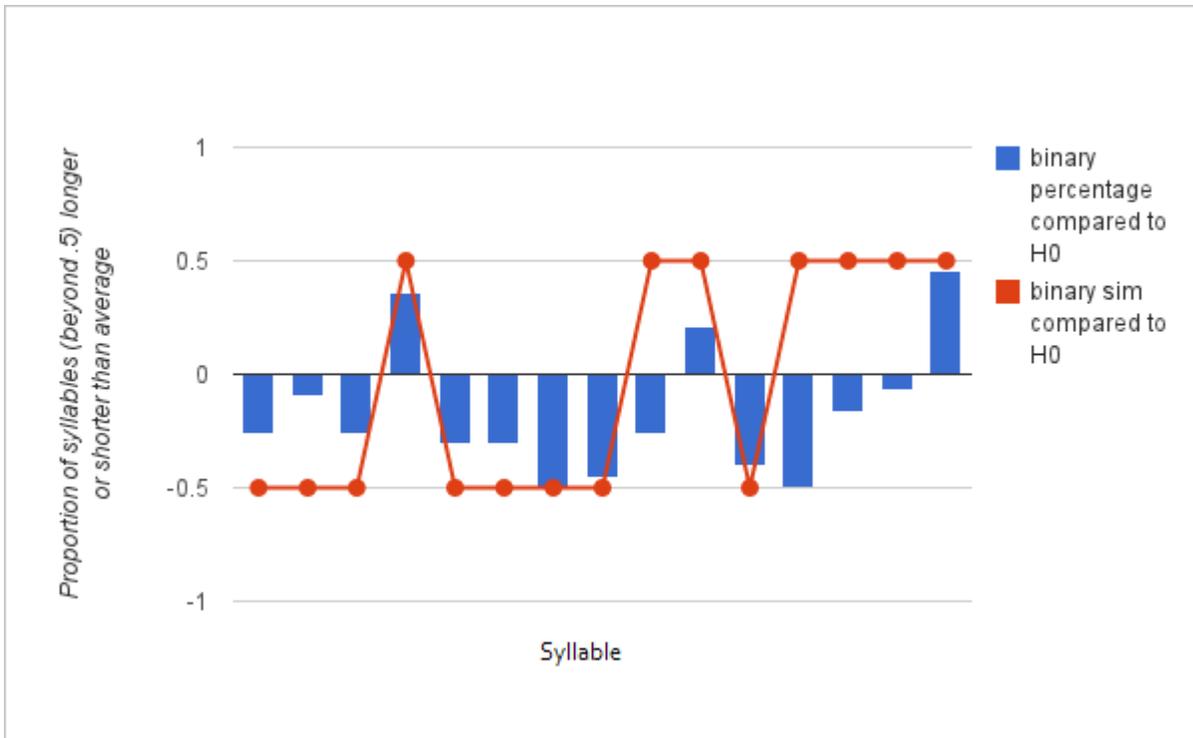


Figure 4-25: Proportion of syllable durations longer or shorter than mean – all productions of French Utterance 2 vs. Simulated French Utterance 2

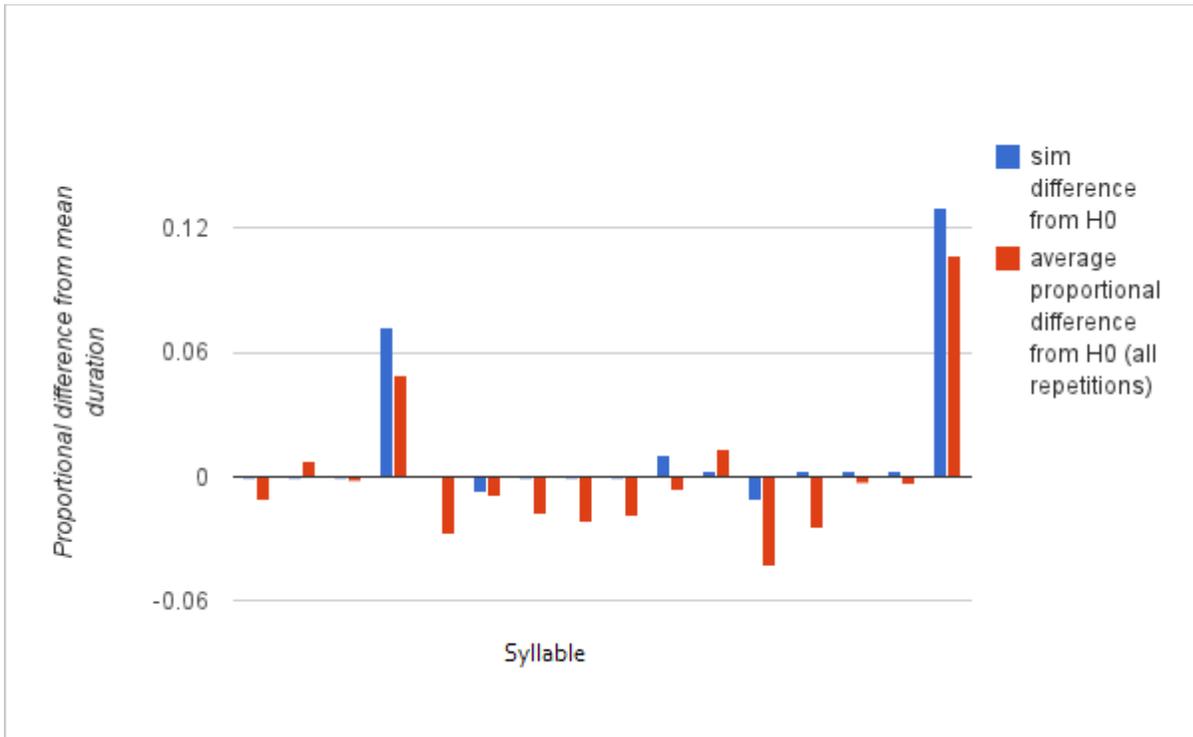


Figure 4-26: Proportional differences in duration of each syllable from the mean – all productions of French Utterance 2 vs. Simulated French Utterance 2

Figures 4-26 and especially 4-24 show how the simulation tracks the pattern of longer and shorter syllables produced by the speakers, but under-predicts the durations of the longer syllables.

As mentioned in the previous section, 4 of the 10 repetition subjects did not produce two successive versions of the syllable *le*, in essence eliminating the long, disfluent copy of the syllable produced by the original speaker. The graph of these productions of Utterance 2 is in Figure 4-27, below. This divergence from the original production suggests that these speakers either did not store this syllable in the prosodic structure of the utterance (the method used in this study was originally developed by Cole and Shattuck-Hufnagel (2011) to elicit storage and reproduction of utterance prosody), or that they chose to “repair” the utterance rather than faithfully reproduce the original.

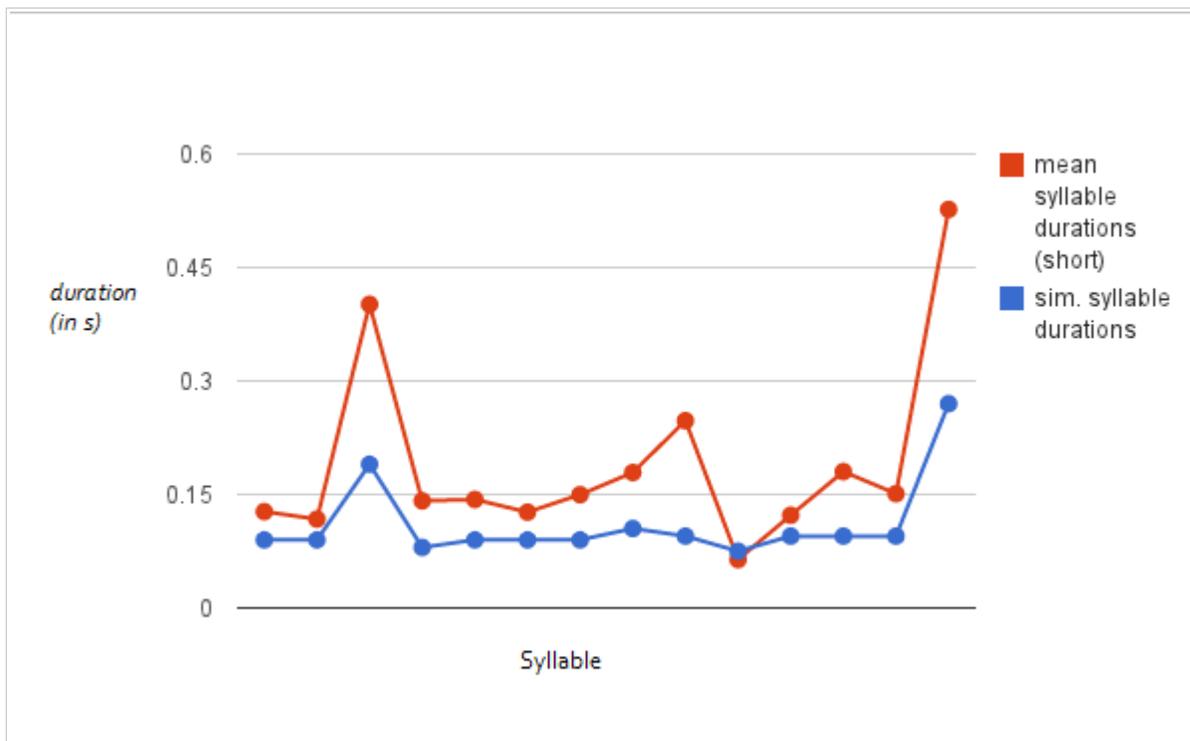


Figure 4-27: Syllable durations, “corrected” productions of French Utterance 2 and Simulated French Utterance 2

4.2.5 Utterance 3

4.2.5.1 Utterance 3 Simulation Compared to Utterance 3 Original Production

Figure 4-28 is an illustration of the syllable durations and prosodic structures of the original production of Utterance 3 and its simulation.

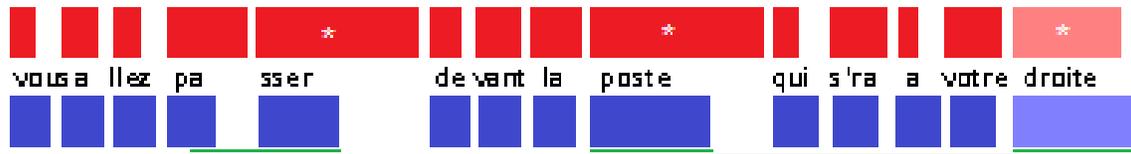


Figure 4-28: A diagram of of French Subject O's production of French Utterance 3 and the simulation of Utterance 3

Figure 4-29 compares the relative length of syllables in the original utterance and the simulation.

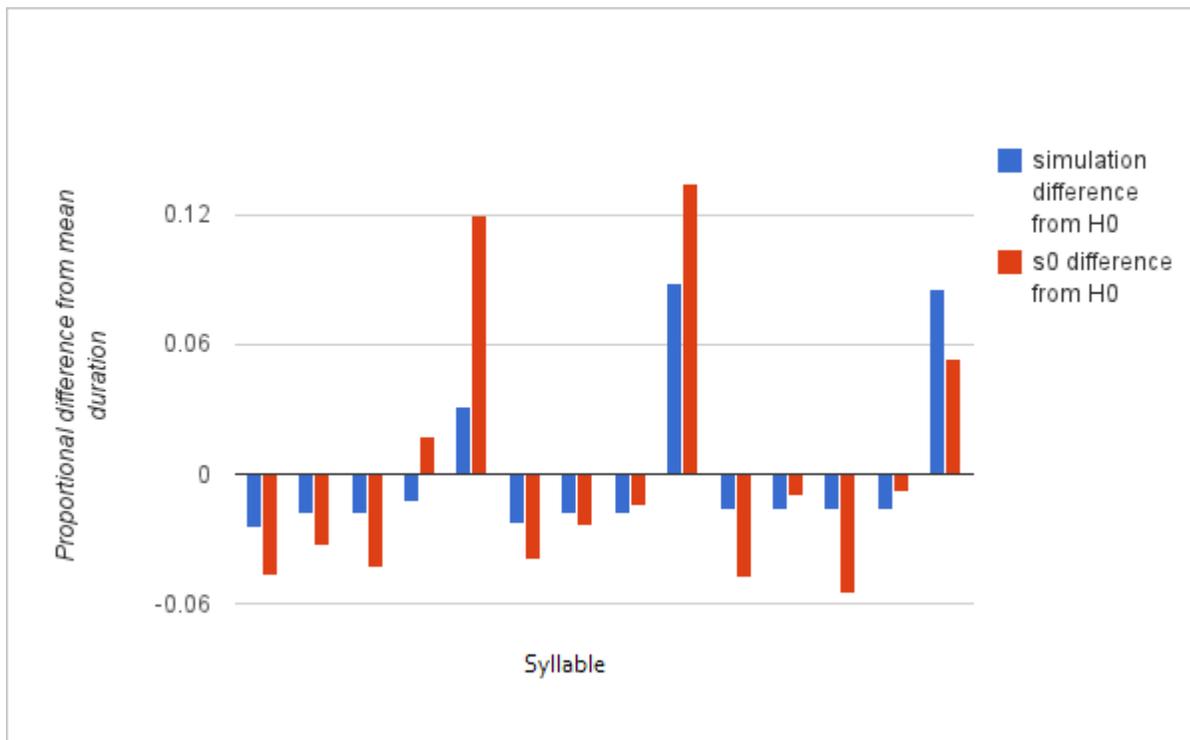


Figure 4-29: Proportional differences in duration of each syllable from the mean – French Subject O's production of French Utterance 3 vs. Simulated French Utterance 3

The simulation of Utterance 3 faithfully models the locations of longer and shorter syllables in the original utterance, with the small exception of the fourth syllable, /pa/, which is slightly short in the simulation and slightly long in the actual production. However, the simulation again underestimates the degree to which the syllable durations vary in the original, especially syllable 5, /se/, although that may be exacerbated by the inherent length of the /s/, as in some of the English utterances.

4.2.5.2 Utterance 3 Simulation Compared to All Utterance 3 Productions

Comparisons between the simulation and the average of all productions of Utterance 3 are presented in Figures 4-30 through 4-32. Figure 4-30 compares the raw duration of each simulated syllable to its mean duration across all productions of the utterance, as well as to its raw duration in French Subject O's original production. Figure 4-31 shows which syllables are longer or shorter than average in simulated Utterance 3 and the proportion of productions of Utterance 3 in which that syllable is relatively long or short, while Figure 4-32 compares the relative duration of each syllable in simulated Utterance 3 to its average relative duration across all productions of the utterance.

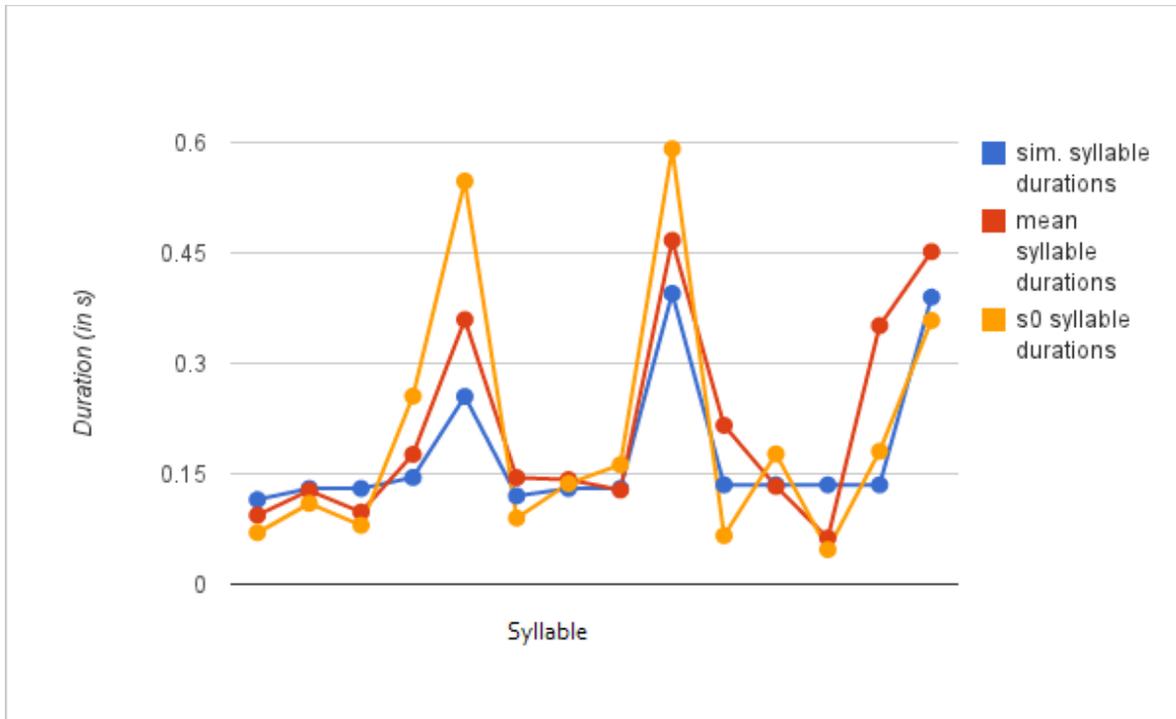


Figure 4-30: Syllable durations, French Utterance 3

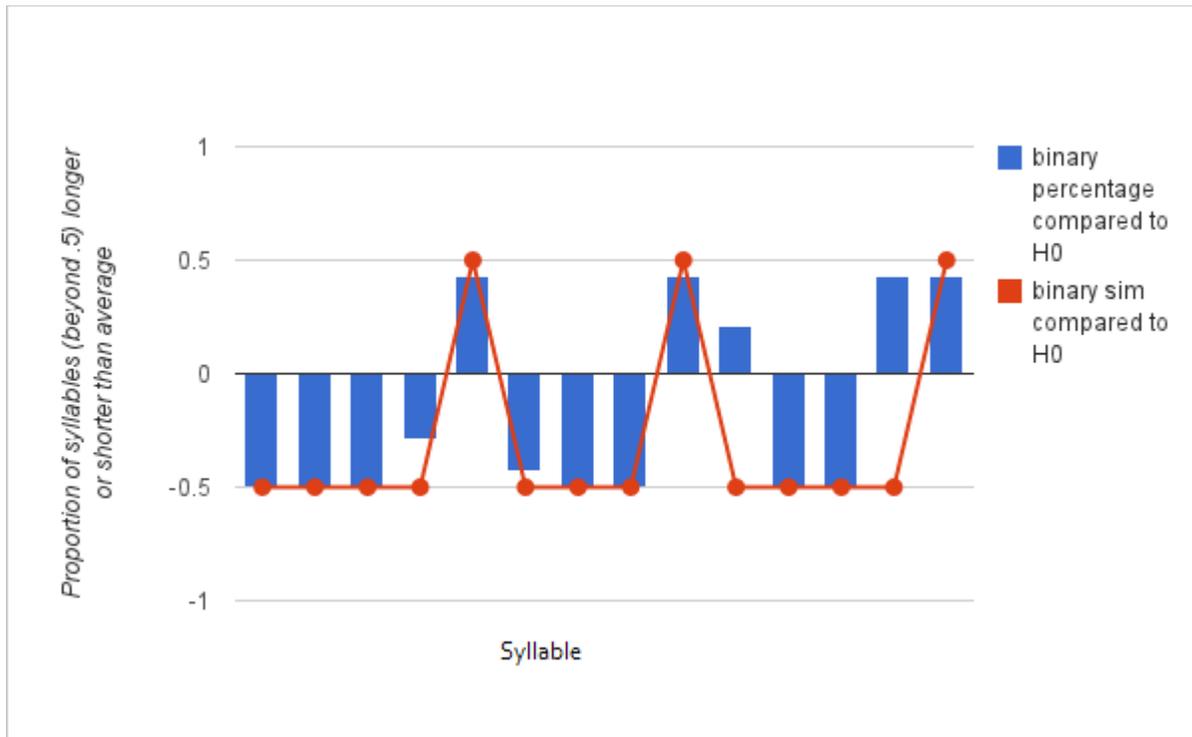


Figure 4-31: Proportion of syllable durations longer or shorter than mean – all productions of French Utterance 3 vs. Simulated French Utterance 3

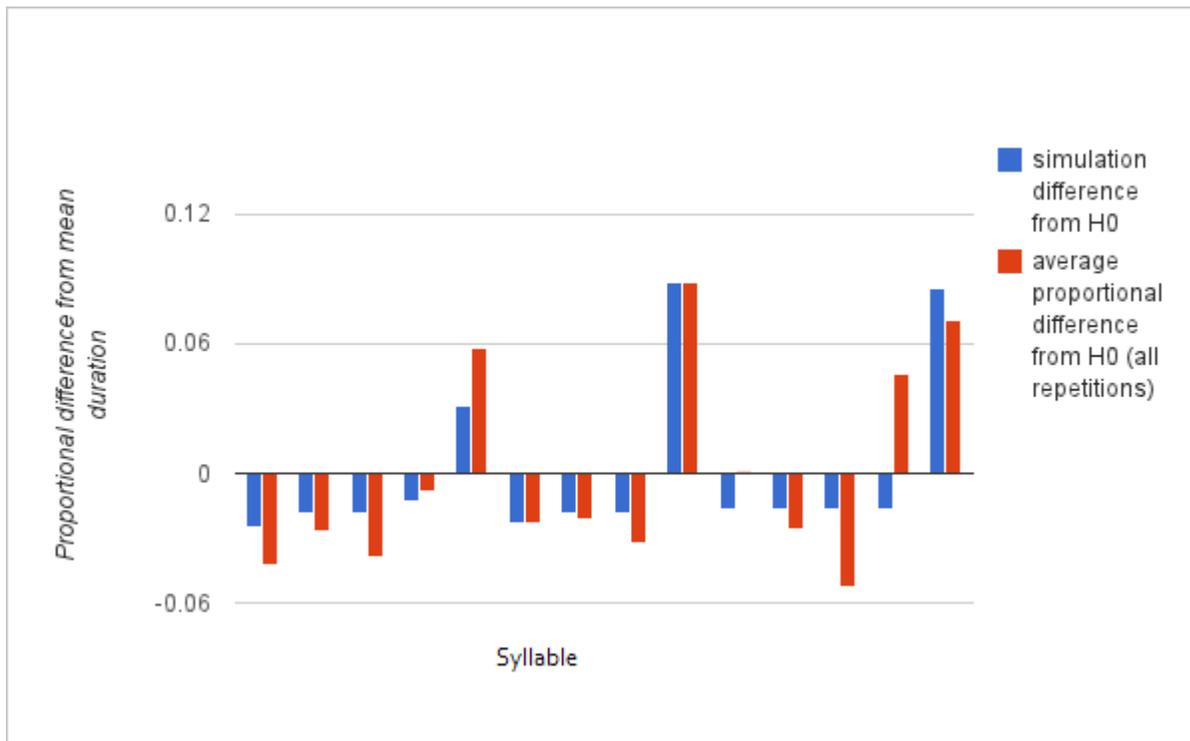


Figure 4-32: Proportional differences in duration of each syllable from the mean – all productions of French Utterance 3 vs. Simulated French Utterance 3

When syllable durations are averaged across all productions of Utterance 3, the lengthening of the accented syllables 5 and 9 (/se/ and /post/) is reduced, reducing the difference between simulation and data. Overall, there is slightly less variation in syllable durations as well, leading to a better match with the simulation. The exception is the penultimate syllable, which shows some lengthening that was not evident in the original production or the simulation.

4.2.6 Do French phrases become more regular with more repetition?

This section compares the temporal structure of the spontaneous utterances 1, 2, and 3 with the same utterances across multiple repetitions. If the prosodic timing of an utterance becomes more regular as it is repeated more, the prosodic timing of the third repetition of that utterance should be more similar to the simulation than the first.

4.2.6.1 Methods

The overall difference between the simulated and actual utterances was measured by calculating the difference in proportional syllable durations between the real and simulated syllables. The average residual difference across all first repetitions of each utterance was then compared to the average residual difference across all third repetitions of that utterance, to evaluate whether the third repetitions of an utterance were more similar to the simulation than the first repetitions.

4.2.6.2 Results

Figure 4-33 shows the overall difference between each simulated utterance and all first and third repetitions of that same utterance. A smaller residual indicates more similarity to simulation.

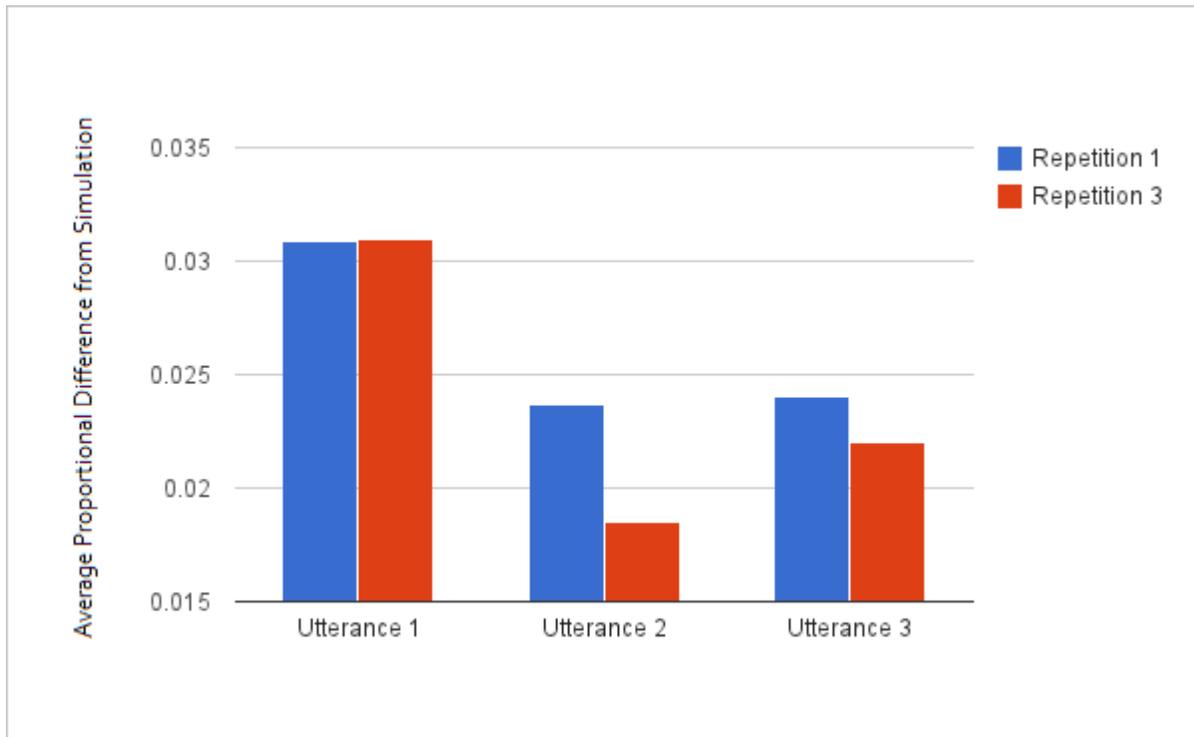


Figure 4-33: Average residual difference in syllable durations from simulation, French

Just as in the English data, in 2 of the 3 utterances, speakers' productions trend toward more similarity to the simulation with more repetitions, but none of the differences between repetition groups reaches significance ($t(9)=-0.071$, $p=0.945$; $t(9)=1.354$, $p=0.209$; $t(8)=1.611$, $p=0.146$).

4.3 Discussion

In answer to the first research question posed in Chapter 1, whether an artificial neural network model based on oscillators can be useful for simulating prosodic timing, given the variability of timing in natural speech, NOMSTR was very successful in generating simulations of spontaneous utterances with a variety of different timing structures, in both French and English. This suggests that despite non-rhythmic influences on spontaneous speech, such as syllable content and idiosyncratic duration changes, much of the temporal structure of spontaneous speech can be modeled by a system whose timing is rhythmic and *regular*, if not isochronous. It is not clear from these data, however, that removing disfluencies and planning time from an utterance via repetition can increase its temporal regularity or similarity to its temporally regular simulation.

The ability of the same basic system to model the prosodic timing structures of both languages provides some insight about the similarities and differences between the two prosody systems. First is the fact that at the level of timing and rhythm, prominences and phrases can be modeled in the same way; in NOMSTR, they are both cycles of an oscillator. The elimination of this distinction shows that the traditionally described difference between stress-accenting in English and accenting at the end of prosodic phrases in French is not necessarily a categorical difference at the level of the prosodic structure itself, but perhaps a difference in how the prosodic structure interfaces with the segmental, lexical, and syntactic information in an utterance; in what aspects of the "terrain" of an utterance (e.g. heavy syllables, syntactic phrase boundaries) are attractive anchor points for oscillation peaks. It is important to note that NOMSTR in its current form does not contain or account for any segmental information, which is of course a factor in the actual speech production system. Future

integration of NOMSTR with information about the segmental content of the syllables in an utterance could lead to even more accurate simulations.

Besides NOMSTR's inability to simulate idiosyncratic or segmentally-influenced variations in syllable duration, there was one consistent issue with the simulated utterances: they tend to underestimate the degree of variation in syllable durations across an utterance. This could be somewhat accounted for by segmental content, for instance by very short reduced vowels in English, but it may also be due to the heavy influence that oscillators have on their paired thresholded nodes (which produce the output) in the current model, perhaps causing the syllable thresholded node to be less flexible in its timing than it needs to be.

Finally, further investigation is needed into methods to compare utterances and simulations which may be based on the same underlying string, but are altered by the addition or removal of a syllable (or syllables). The most blatant example of this complication in these data is the deletion of *le* by some speakers in French Utterance 2, discussed in Section 4.2.4.2, but there are other places in the French utterances, and at least one place in the English utterances (*d'you*, [dju]), where the imitating speakers could insert an additional syllable into the phrase by producing a schwa which the original speaker had elided. In this study, imitated productions were assumed to be the same number of syllables as the original, unless evidence for an extra syllable was clear, which was very rare. Still, these data show that answering the question of whether utterances become more regular with repetition with certainty will require the ability to directly compare between utterances and simulations with different numbers of syllables. Further research in this area will require investigation into a valid method to do that comparison.

Chapter 5

Interaction Between Prosodic Level and Its Effect on Prosodic Timing

5.1 Introduction

The two studies described in this chapter measure the impact of accents and prosodic phrase boundaries, and their combined impact, on the duration of syllables. These effects are examined and compared in French (as spoken in France) and American English. The American English condition is a partial replication of Turk & Shattuck-Hufnagel (2007), which found that the accentual and phrase-final lengthening effects were compounded on accented syllables near a phrase boundary even when syllables between the accented syllable and phrase-final syllable remained unlengthened. The French condition applies durational measurements to French phrases with similar construction designed for Astésano et al. (2007), which found accentual lengthening on some syllables immediately following accentual phrase boundaries. The present study attempts to verify this finding and demonstrate how NOMSTR can be used to predict this type of interaction of lengthening effects.

5.2 English

5.2.1 Procedure

10 adult native English speakers were recorded producing sentences containing 3 proper nouns in the structures A or (B and C) or (A or B) and C. For example: *Please say 'Michigan or (Madison and Vatican) will play'*. OR *Please say '(Jamaica or Dakota) and Chicago will play'*. The nouns were either three or four syllables in length, with accent on the antepenultimate syllable. All nouns within a sentence contained the same number of syllables. The sentences were presented orthographically, placed one per slide in a PowerPoint slide show. Each sentence in the set appeared twice in the slides; once with a set of parentheses around the first two target words, and once with a set of

parentheses around the second two. The speakers were presented with the slides displayed on a computer screen and asked to read the sentences aloud, referring to the parentheses to indicate how each sentence should be disambiguated; specifically, subjects were told that they should use the parenthesis to tell them how to “group” or “chunk” the words in the sentence. The speakers were recorded in a sound-dampened booth using a high quality microphone.

5.2.2 Analysis

Target words were divided into syllables respecting word or morpheme boundaries, and otherwise following the maximum-onset principle (Pulgram, 1970; Noske, 1982), as in Turk & Shattuck-Hufnagel (2007). The duration of each syllable in the target words was measured. Lengthening was measured by subtracting the duration of each syllable in phrase-medial position from the duration of the same syllable in phrase-final position.

5.2.3 Results

As shown in Figure 5-1, below, in penultimate accent words (e.g. *Dakota*), the amount of lengthening in each syllable (as compared to itself the in non-phrase-final condition) increases with decreasing distance to the phrase boundary. In antepenultimate accent words (e.g. *Michigan*), however, the accented antepenultimate syllable shows lengthening ($t(90)=2.3, p<.05$), while the unaccented penultimate syllable does not show lengthening, even though it is closer to the phrase boundary.

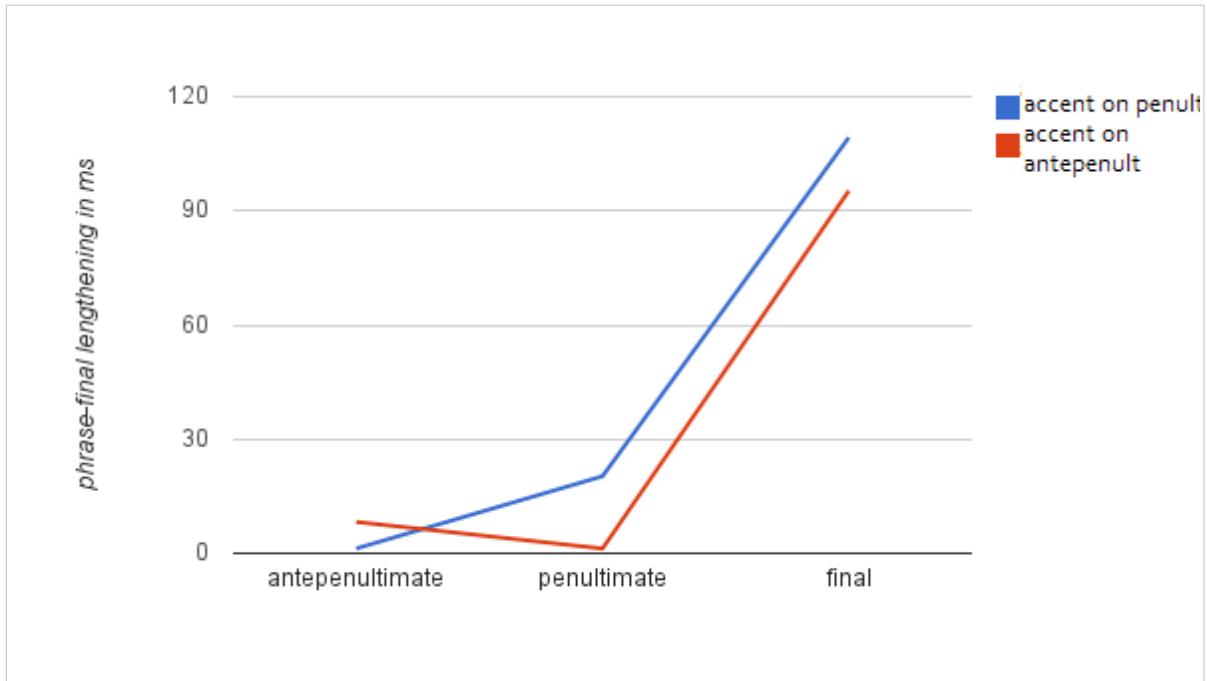


Figure 5-1: Lengthening by syllable in English phrase-final words

These results confirm the findings in Turk and Shattuck-Hufnagel (2007), suggesting that there is some interaction between the effects of the phrase boundary and the accent on the accented syllables. The next section demonstrates how this type of interaction can be simulated using NOMSTR.

5.2.4 Simulation

In order to simulated the interactions between the three prosodic levels seen in the English data (syllable, accent, and phrase, first the connections between the thresholded nodes in the model are adjusted so that the phrase and stress nodes inhibit the syllable node, causing syllable duration to increase under the influence of stress and near phrase boundaries (when the accent and phrase nodes fire, respectively).

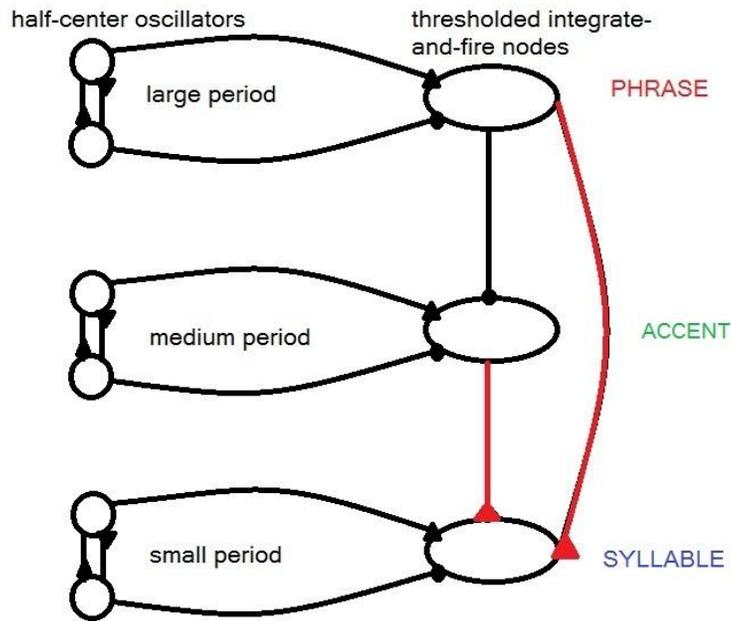


Figure 5-2: Inhibitory connections necessary to simulate English results

Next, the constructive interaction between the accent and phrase boundary in the data is simulated by creating an excitatory connection between the phrase and accent thresholded nodes in the model. This results in stronger activation of the accent node while the phrase node is firing (i.e. near a phrase boundary), and thus stronger inhibition of syllable node by the accent node in that context.

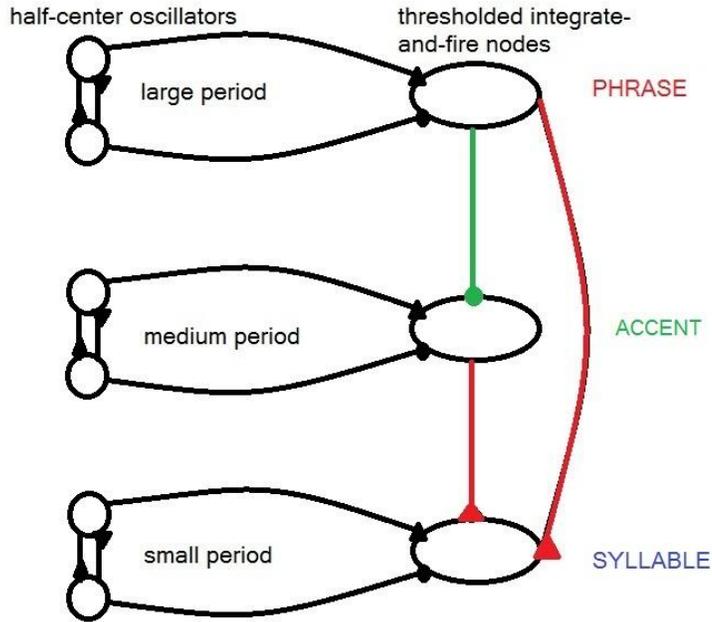


Figure 5-3: Excitatory and inhibitory connections necessary to simulate English results

With these settings, syllables could experience more accent lengthening near a phrase boundary even if the syllable was too far from the boundary (peak phrase node output) for the inhibition from the phrase node to be strong enough to cause significant lengthening on its own.

Figure 5-4 shows an example of output from the model with these settings when the phase of the accent oscillator is set so that its peak is coincident with the penultimate syllable of the phrase, simulating the penultimate accent condition (e.g. *Dakota*).

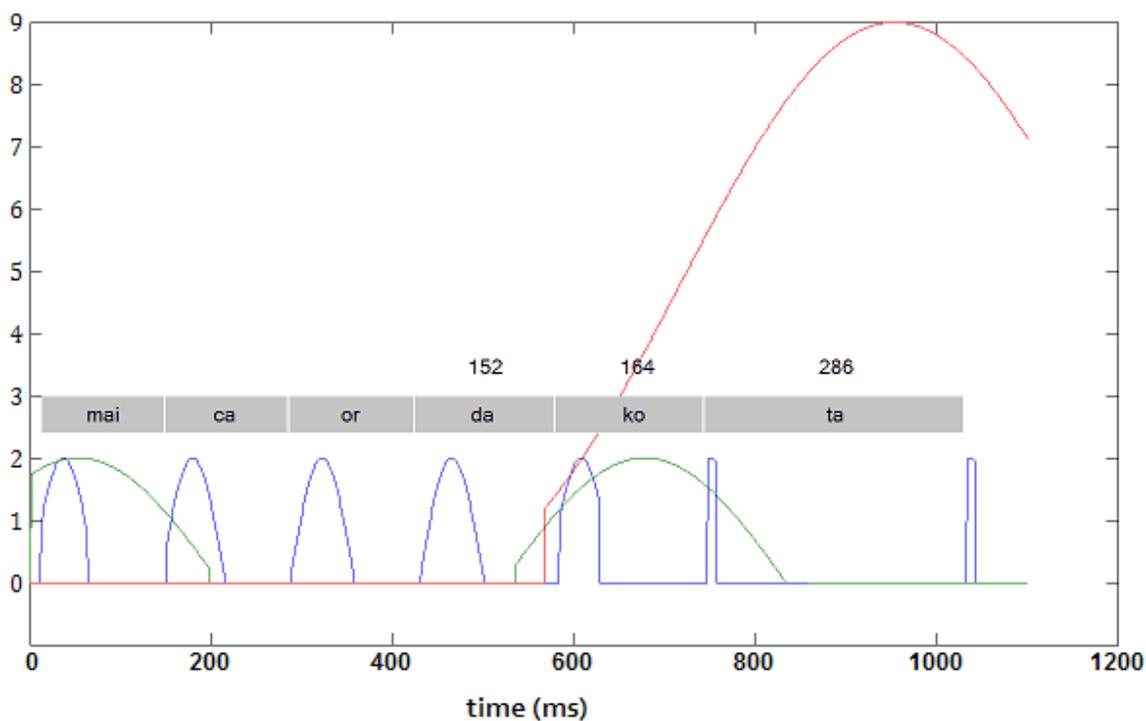


Figure 5-4: Simulation of English penultimate accent condition

Figure 5-5 shows an example of output from the model with the same connections and weights, but with the phase of the accent oscillator set so that its peak is coincident with the antepenultimate syllable of the phrase, simulating the antepenultimate accent condition (e.g. *Madison*).

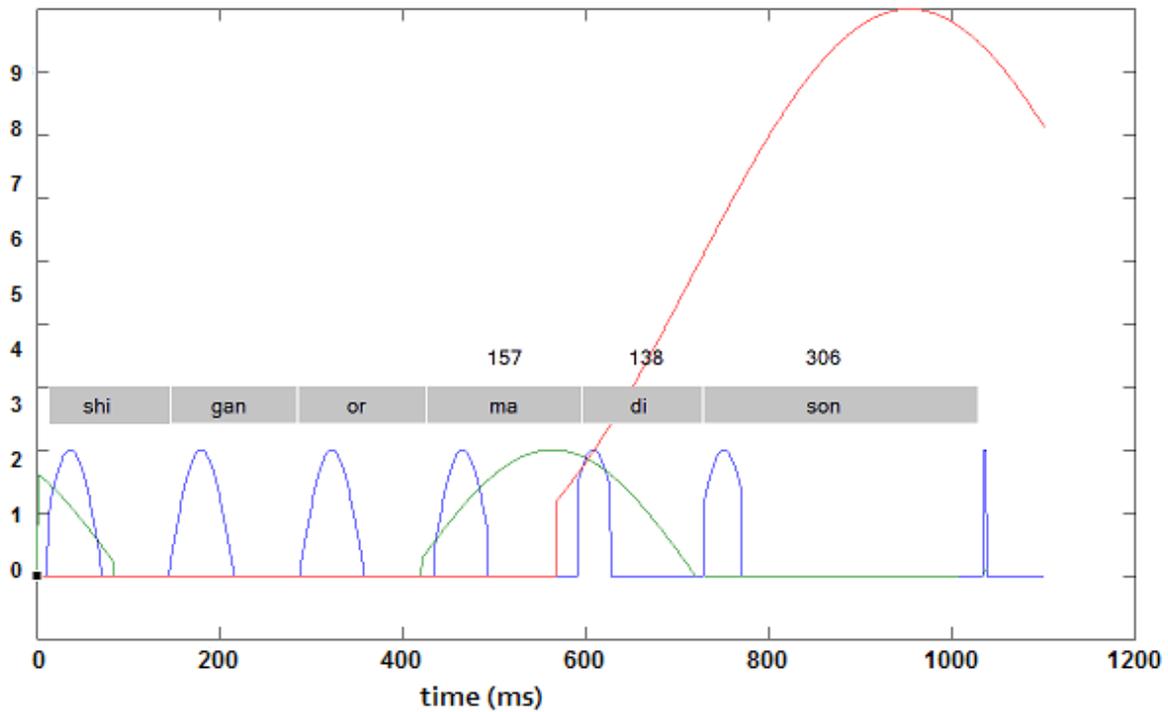


Figure 5-5: Simulation of English antepenultimate accent condition

The durations of the syllables in these simulations can be compared to the durations of accented and unaccented syllables that are not under the influence of a phrase node peak in the same model output in order to measure phrase-final lengthening in the simulation, much in the same way that lengthening was measured in the data by comparing the durations of syllables in phrase-final and non-final words. Figure 5-6 shows the phrase-final lengthening effects in the penultimate accent and antepenultimate accent simulations.

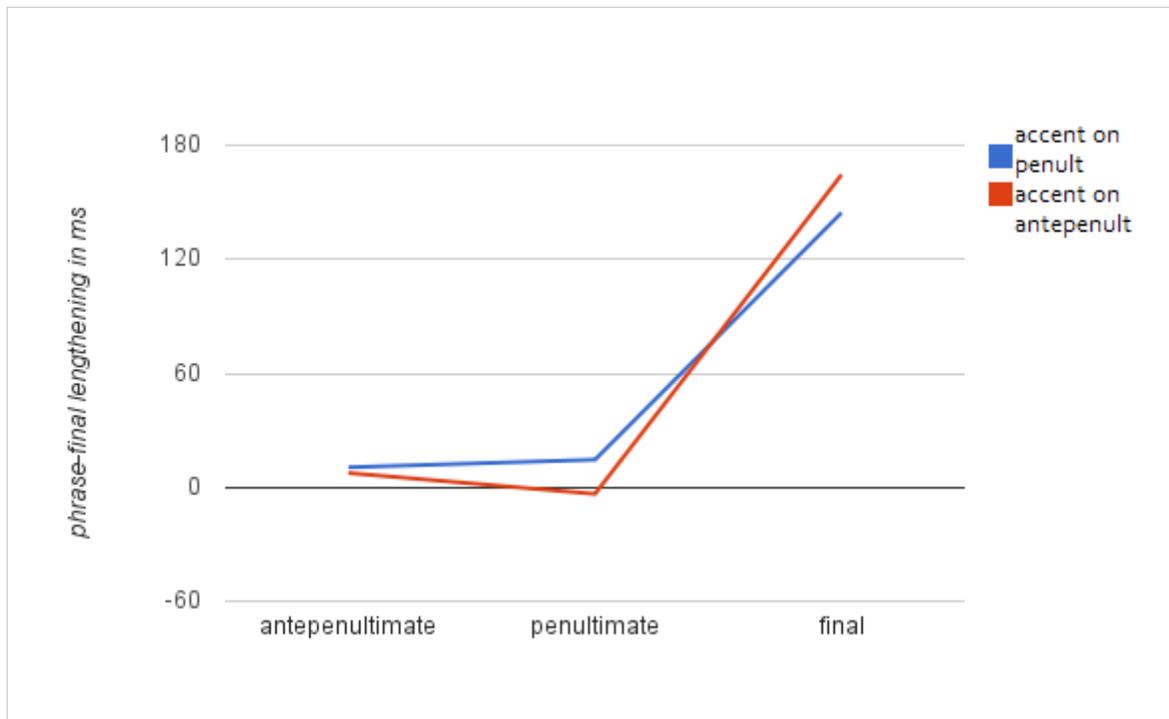


Figure 5-6: Simulated lengthening by syllable in phrase-final words

As in the actual data, the lengthening on the final syllable is much greater than that on the two preceding syllables in both conditions. In the penultimate accent simulation, as in data from the penultimate accent condition, the amount of lengthening slightly increases from antepenult to penultimate syllable. In the antepenultimate accent condition, though, the amount of lengthening on the penultimate syllable is less than on the antepenultimate syllable; this matches the decreasing-then-increasing lengthening pattern in the data from the antepenultimate accent condition.

5.3 French

5.3.1 Procedure

10 adult native speakers of French from France were recorded producing sentences containing two nouns and an adjective in the structures A et (B C) or (A et B) C. For example: *Les infamies et (les barbaries certifiées), en fait, sont relativement rares.* OR *(Les publications et les filmographies) militantes, en fait, sont souvent*

méconnues. The target words were either three or four syllables in length. The sentences were placed one per slide in a PowerPoint slideshow. Each sentence in the set appeared twice in the slides; once with a set of parentheses around the first two target words, and once with a set of parentheses around the second two. The speakers were presented with the slides on a computer and asked to read the sentences aloud, using the parentheses to indicate how each sentence should be disambiguated; specifically, subjects were told that they should use the parenthesis to tell them how to “group” or “chunk” the words in the sentence. The speakers were recorded in a sound-dampened booth using a high quality microphone.

5.3.2 Analysis

The duration of syllables in 3 positions within the third target word of each sentence was measured: the first syllable of 4-syllable words (pre-antepenultimate in the IP and word-initial), the second syllable of 4-syllable words (antepenultimate in the IP and not word-initial), and the first syllable of 3-syllable words (antepenultimate in the IP and word-initial). A table of these conditions is shown in Figure 5-7.

	4 syllables before IP boundary (pre-antepenultimate)	3 syllables before IP boundary (antepenultimate)
Word initial	first syllable of 4-syllable target words	first syllable of 3-syllable target words
Not word initial		second syllable of 4-syllable target words

Figure 5-7: Syllable positions analyzed for duration in French sentences

Lengthening due to Initial Accent (or post-Accentual Phrase-boundary lengthening) was measured by subtracting the duration of each syllable in a target word immediately following an Accentual Phrase (AP) boundary (in the (A B) C condition), from the

duration of the same syllable when the target word was in AP-medial position (in the A (B C) condition).

5.3.3 Results

Figure 5-8 shows the mean Initial Accent (IA) lengthening in IP antepenultimate and IP pre-antepenultimate syllables, as well as in the control antepenultimate syllables.

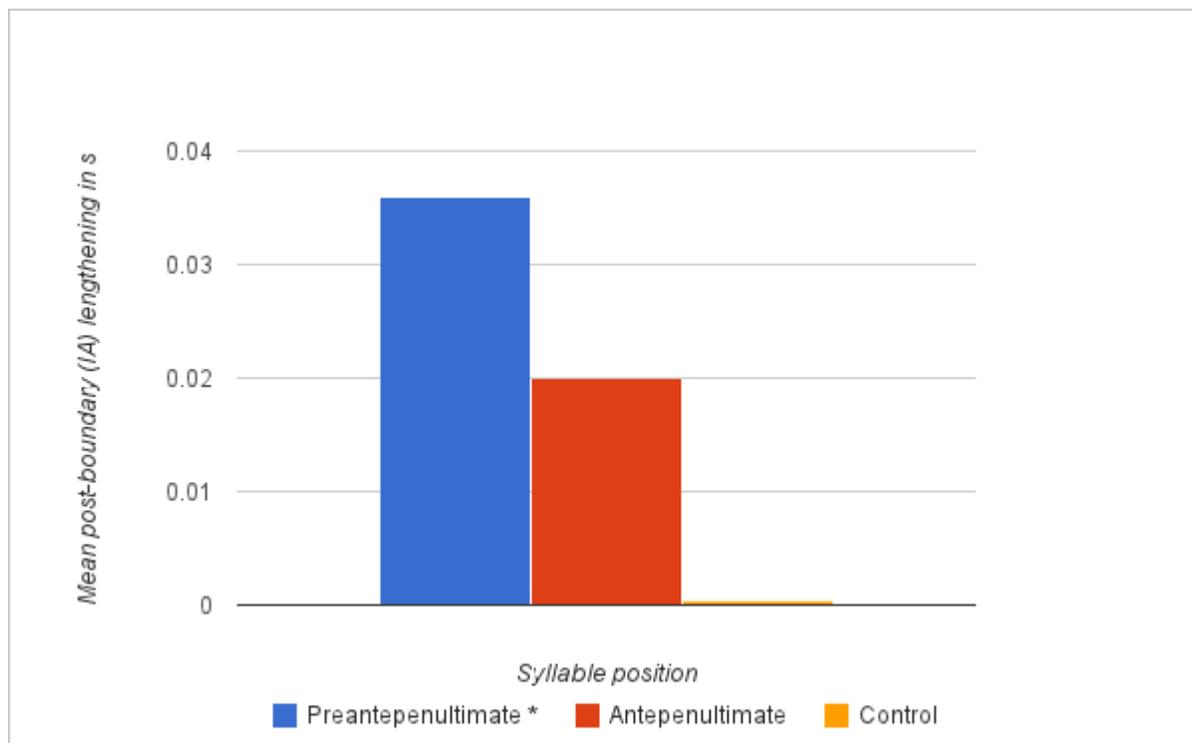


Figure 5-8: Effect of distance from IP boundary on Initial Accent lengthening

Significant IA lengthening was only found in syllables which were immediately post AP boundary and 4 syllables back from the end of the IP (pre-antepeultimate) ($t(57)=4.394$, $p<.001$). Syllables which were immediately post AP boundary and only 3 syllables back from the end of the IP (antepenultimate) did not show significant lengthening. As expected, the durations of the control syllables, which were antepenultimate but not word-initial, were virtually identical in both the AP-initial and AP-medial conditions.

These results suggest that closeness to the IP boundary actually prevents IA lengthening to some degree, since significant lengthening occurred in pre-antepenultimate syllables but not in antepenultimate syllables, which are closer to the end of the IP. In the next section I will describe how simulation of these results using NOMSTR suggests two possible mechanisms which might produce these results.

5.3.4 Simulation

First, what does an Initial Accent or post-boundary lengthening mean in the context of the model? Lengthening of a syllable following a boundary occurs in simulations when the peak of an accent or phrase node which has an inhibitory connection to the syllable node is positioned so that its tail end “spills over” into the post-accent or post-boundary syllable.

Figure 5-9 shows a simulation of the final syllables of a French 4-syllable (A B) C phrase (e.g. *Les bonimenteurs et les baratineurs fabulateurs, en fait, sont plutôt répandues*). The peak of the medium sized oscillator, serving as the Accentual Phrase oscillator, covers the last syllable of the previous AP, but spills over somewhat into the next syllable, which is the first syllable of the next AP and the pre-antepenultimate syllable of the Intonational Phrase. This is the condition which would lead to Initial Accent lengthening on the pre-antepenultimate syllable, as the AP thresholded node inhibits the syllable node during that cycle.

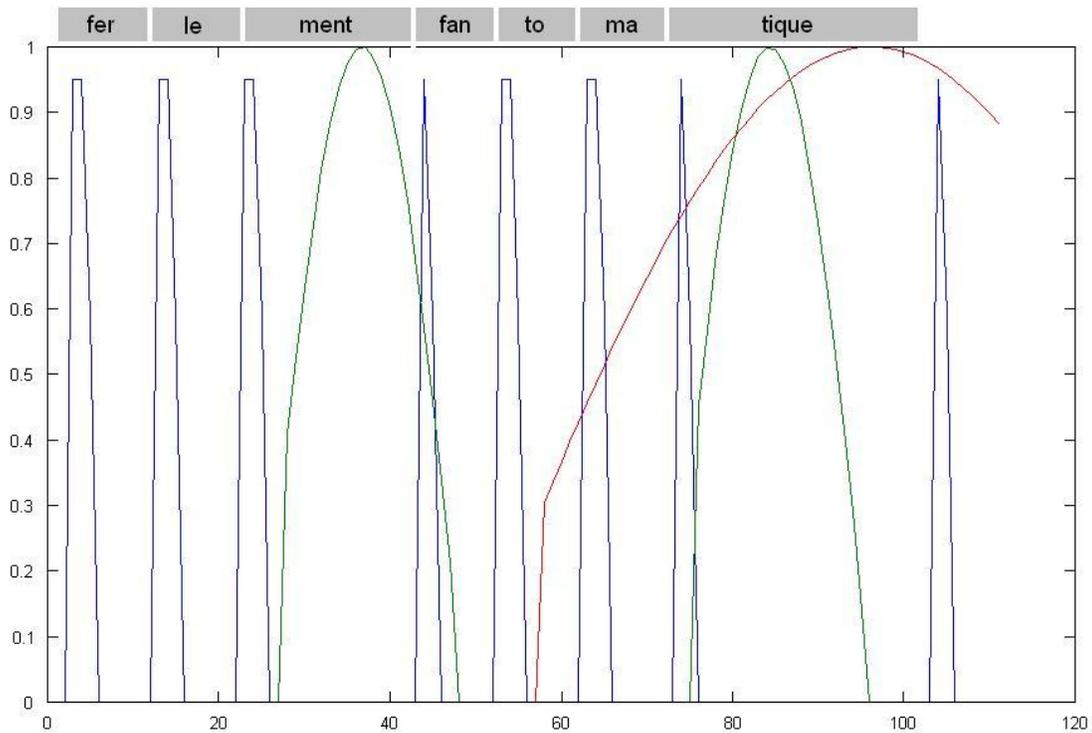


Figure 5-9: Simulation of Initial Accenting on a French 4-syllable target word

This spillover might be prevented on syllables closer to the end of the IP in two ways. First, since the final syllable in the IP is also the final syllable of an AP (as would always be the case for an IP-final full syllable in French), the AP node and likely the AP oscillator which drives it, must peak on the final syllable of the IP. It is possible that the wavelength of the AP oscillator, the natural distance between peaks of the AP node, is too long for the earlier peak to “spill over” into the post-AP boundary syllable when that syllable is only 1 syllable away from the next AP boundary, and thus the next peak of the AP node. Figure 5-10 shows this type of simulation.

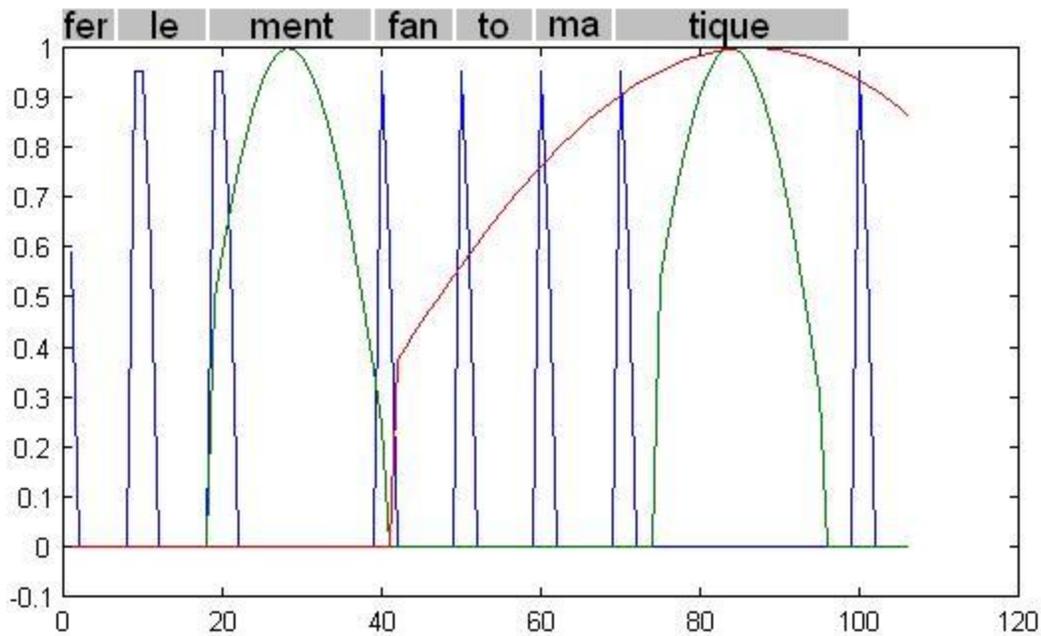


Figure 5-10: Simulation of Initial Accent lengthening prevented due to Accentual Phrase oscillator period

Second, it is possible that in French, the IP node has an inhibitory connection to the AP node. If the antepenultimate syllable of the IP is close enough to the IP boundary to be affected by the peak of the IP node, the output of the IP node would inhibit the AP node during that syllable cycle, reducing or preventing any effect of the AP node on the syllable. Figure 5-11 shows this type of simulation.

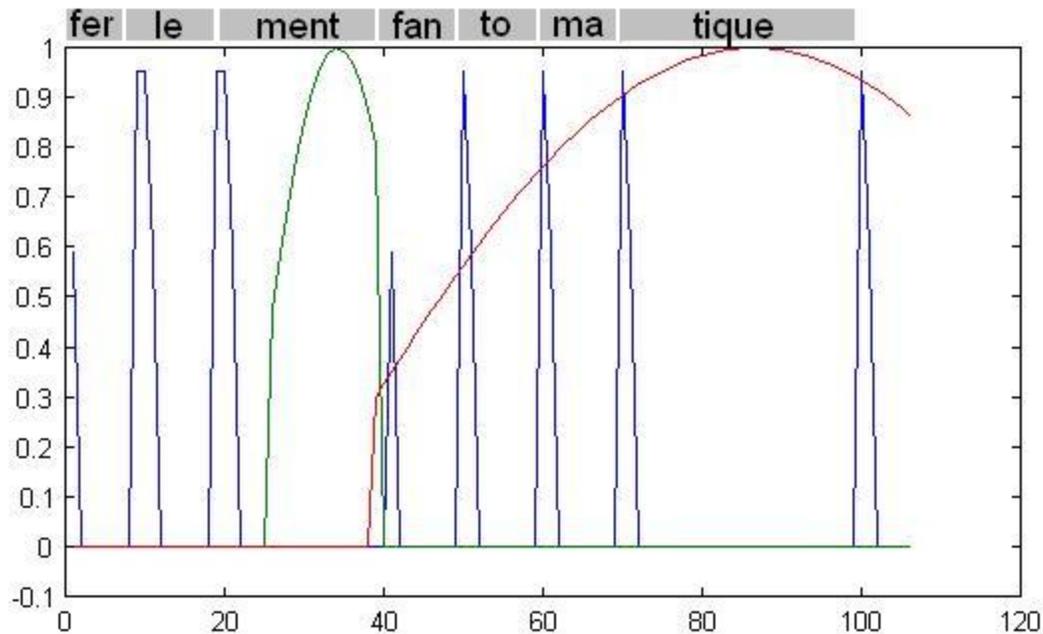


Figure 5-11: Simulation of Initial Accent lengthening prevented due to Intonational Phrase oscillator inhibiting Accentual Phrase oscillator

5.4 Discussion

The two studies in this chapter provide examples of how NOMSTR can be used to explore the second research question presented in the introduction: what is the relationship between the syllable, stress group, and phrase, in terms of their temporal coordination, and how does this relationship differ between languages with differing prosodic systems (specifically English and French)? In English, accented syllables near a phrase boundary were found to undergo more lengthening than adjacent unaccented syllables. This was an interaction which had been described previously (Turk & Shattuck-Hufnagel 2007), but had not been modeled using existing models of prosodic timing. I simulated this lengthening effect in NOMSTR by modeling a prosodic structure in which accents and phrase boundaries inhibit (thus lengthening) the syllable, while phrase boundaries excite accents, providing nearby accents greater syllable lengthening power. In French, the lengthening of syllables with Initial Accents, or syllables immediately following the end of an accentual phrase, was found to be

lessened in the presence of an upcoming IP boundary. Because of the coincidence of Intonational and Accentual Phrase boundaries in French, NOMSTR provided two possible ways to simulate this effect: either with AP boundary crowding or by modeling an inhibitory connection between the IP and AP nodes. As a note, this inhibitory connection did prove useful in creating a more accurate simulation of the third French utterance in Chapter 4, which may provide some weight in favor of that solution.

Chapter 6

Effects of Phonological Structure on Speech Errors

6.1 Background

The first two studies of this project focus on the effects of intralevel prosodic coordination on the timing of suprasegmental speech elements such as syllables and phrases, but speech production timing is important at the segmental or gestural level as well. This third section of the thesis is designed to examine a similar question at the level of the segment or gesture: how the nature of the coordination between segments and syllables and among the segments themselves can affect the serial timing of the segments in an utterance. Experiment 3 asks what effects, if any, syllable structure has on the distribution of speech errors across different syllable positions. The modeling portion of this section looks at what properties would be necessary in a model to simulate any asymmetries in error likelihoods among syllable positions, and then tests a hierarchical connectionist model of speech production which uses NOMSTR as its suprasegmental portion. This larger model will be tested to see whether it has the necessary elements proposed to simulate the speech error patterns found in experiment 3.

Accurate speech production relies on both the appropriate timing coordination between segments/gestures and the correct sequential timing of segments/gestures in a string. Patterns of gestural/segmental timing can be used to shed light on the organization of the speech production system in much the same way that prosodic timing patterns can. Most notably, work by Browman, Goldstein, Nam, Saltzman and their colleagues have used observations about the relative timing between gestures at different syllable positions to model the internal structure of the syllable, and how syllabic constituents are coordinated with each other. Browman and Goldstein (2000) propose a model to explain findings by Byrd (e.g. 1996) and others that the center of an

onset consonant or onset cluster is consistently timed with respect to the onset of the vowel gesture, while the onset of coda consonants are consistently timed with respect to the offset of the preceding gesture, with no clustering effects. In the Browman and Goldstein model, further explicated by Nam and Saltzman (2003), the coordination between gestures within a syllable is described using coupling and competition relations between oscillators representing each gesture: all onset consonants are coupled in phase with the onset of the nuclear vowel, but are competitive with each other, and all coda consonants are coupled in phase with the offset of the (vowel or consonant) gesture preceding them (see Fig. 6-1).

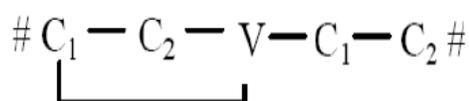


Figure 6-1: Diagram of syllable structure in Articulatory Phonology

One important method for investigating sequential gestural/segmental timing is the study of speech errors, or slips of the tongue. Speech errors, especially the common types of anticipations, perseverations, and exchanges, can be viewed as errors in the appropriate sequential timing pattern of the gestures/segments of a phrase or utterance. Many studies have investigated the types and frequencies of speech errors in order to shed light on the nature of gestural/segmental coordination and sequencing by examining the ways that the coordination and sequencing can break down. Goldstein et al. (2007) investigate speech errors resulting from the repetition of pairs of words differing only in the initial segment (e.g. cop top) and find that most errors result from the simultaneous production of the alveolar and velar gestures in the initial positions of all syllables. They describe how these types of errors could arise from pressures in the Articulatory Phonology syllable model, in which gestures coordinated by means of coupled oscillators may prefer to settle into a 1-to-1 ratio—that is, for each syllable-initial gesture there is exactly one corresponding nuclear vowel gesture used in both words—a stable coupling state for oscillators. In several papers, Dell (1984, 1985, 1986,

1988) proposes a hierarchical connectionist model to simulate how the segments in an utterance are produced in the correct sequential order, and uses speech error data to show how noise or errors in such a system could lead to the types of speech errors produced by real speakers. Dell's model represents multiple connected levels of phonological organization (e.g. features, phonemes, syllable constituent, and syllables), as well as activation of the network which spreads over time, so that the activation level of each output node is based on both its structural location within the phonological hierarchy but also its sequential location in the utterance. This type of model can successfully simulate many aspects of produced speech errors, such as the tendency of errors to adhere to intended syllable position, the increase in errors at faster production speeds, and the tendency for errors to operate over relatively short distances.

Vousden, Brown and Harley (2000) build on the use of artificial neural network (ANN) models for serial ordering in speech in the development of OSCAR, an ANN model that simulates serial order in memory (Brown 2000), as a model of speech production. OSCAR shares many similarities with earlier ANN models of speech production such as Dell's. First, its output is the serial selection or "firing" of artificial neurons in an output layer; in OSCAR these output nodes represent segments or phonemes. Second, it is hierarchical and relies on the association of each output node with its appropriate "location", called "phonological context vector" by Brown and colleagues, in the levels above it during the encoding or learning phase in order to select output nodes in the appropriate order during production. The major difference is that in the OSCAR model, levels of phonological structure larger than the segment are represented by a series of oscillators, which by their nature change state over time, allowing the phonological context to proceed through each state and serially select each output node in order without separately associating each instance of a segment (or syllable) with the segment that proceeds and follows it. The OSCAR model is successful in predicting the types, rates, and proportions of speech errors produced by speakers in a large corpus, as well as simulating the syllable position and distance constraints found in speech error data.

However, this type of speech production model does have some drawbacks. Some tendencies found to occur in speech error data are not predicted by any of these models. For instance, previous studies of speech errors have found that speakers are less likely to produce errors on vowels than on consonants (McKay 1970, Ellis 1980). Models which do not specify the structure of the phonological hierarchy above the segment, and which treat vowels and consonants the same (or nuclei and onsets/codas the same, which in English studies are often conflated), such as OSCAR, cannot predict this type of asymmetry in error frequencies. Models which rely on the traditional hierarchical syllable model (Fig. 6-2), as in Dell 1984, are also unable to simulate this effect, because the traditional syllable model does not represent the nuclear vowel differently from the consonants overall, and in fact places it in a symmetrical position with the coda consonant.

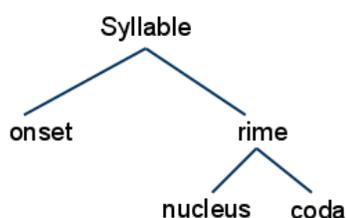


Figure 6-2: Diagram of traditional syllable structure

Additionally, articulatory studies of speech errors (e.g. Pouplier 2003, Goldstein et al. 2007) have found that most speech errors are not due to the wholesale exchange of one phone for another, but rather to the intrusion of an erroneous gesture, which may be co-produced with the intended gestures for that phone. Models of sequential phonological string encoding which use winner-take-all output networks in order to output a single segment (usually a phone) at a time cannot capture this aspect of error production.

6.2 Study

This study uses speech errors produced by speakers in a speeded repetition task to investigate whether there are in fact differences in the frequency of errors for different

syllable positions, or between vowels and consonants overall.

6.2.1 Procedure

Syllables were constructed from a set of 6 onset Cs, [b], [p], [v], [f], [d], and [t] (presented orthographically to subjects as *b, p, v, f, d, t*); 4 Vs, [æ], [a], [ʌ], and [ɪ] (presented as *a, o, u, i*); and 4 coda Cs, [m], [n], [s], and [ŋ] (presented as *m, n, s, ng*). Phrases were assembled from the syllables with four CVC syllables in each phrase with no repeated consonants or vowels (e.g. *vas pon dum fing*). 10 speakers of American English with no speech impairments viewed 45 distinct phrases presented orthographically, one at a time, on a computer screen. Subjects repeated each phrase six times along with a 200bpm metronome. Productions were audio recorded in a sound-dampened booth and later transcribed phone-by-phone by transcribers with experience in phonetic transcription.

Transcribers were able to replay the data recordings as many times as necessary to ensure accurate transcription, and were instructed to record a “?” if they were unsure which phone was being produced; all “?”s were discarded and not included in the data. Two transcribers each transcribed half of the data and a third transcribed 10% of the data assigned to each of the primary coders. There was a 95% agreement rate between the third transcriber and each of the two primary transcribers.

6.2.2 Analyses

The transcriptions were compared with the list of stimuli (the intended productions), and any phone in a production which did not match the intended phone was marked as an error. The frequency with which subjects produced errors was compared among syllable positions.

6.2.3 Results

Subjects produced disproportionately more errors on consonants than on vowels ($\chi^2(1, N = 1177) = 402.18, p < 0.0001$), and more errors on onsets than on codas ($\chi^2(1, N=1066)=173.45 p<0.0001$). See Figures 6-3 and 6-4

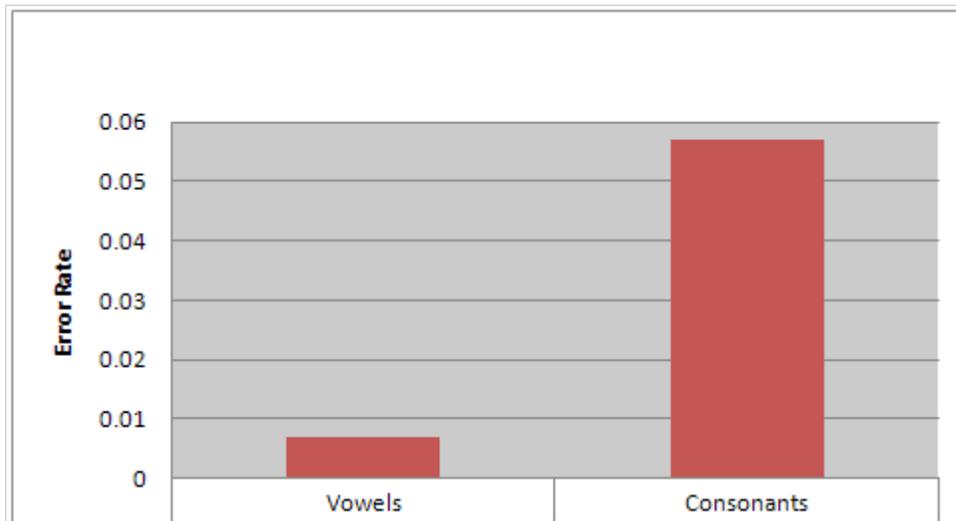


Figure 6-3: Errors by segment type

Subjects also produced a significantly lower rate of errors on nuclei than codas ($\chi^2(1, N=415) = 187.57, p < 0.0001$).

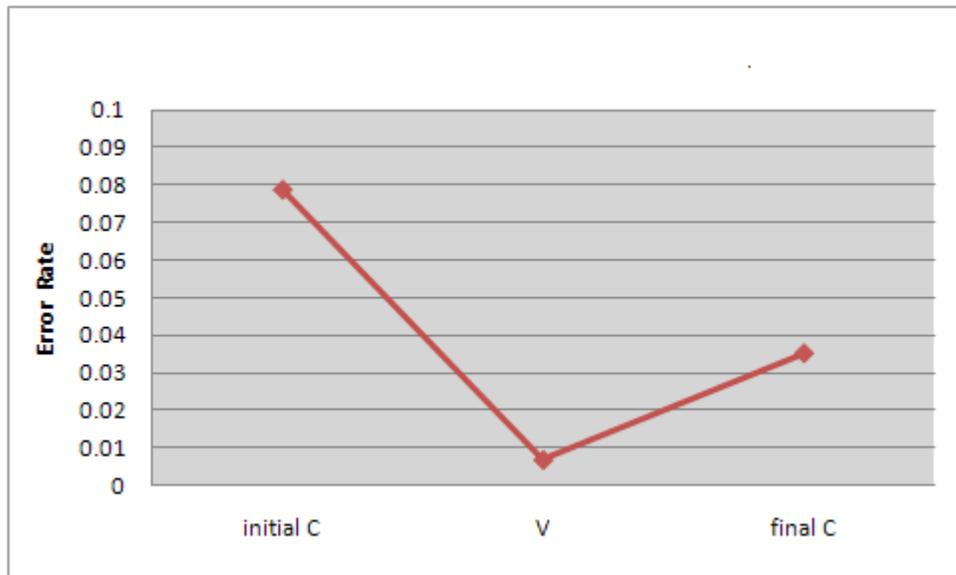


Figure 6-4: Errors by syllable position

As shown in Figure 6-5, errors on more than one contiguous segment did not coincide at the levels that would be expected if the errors were independent of each other and co-occurring at chance rates. Errors on both contiguous VC and CV sequences occurred more often than expected based on the rate of nucleus and coda errors.

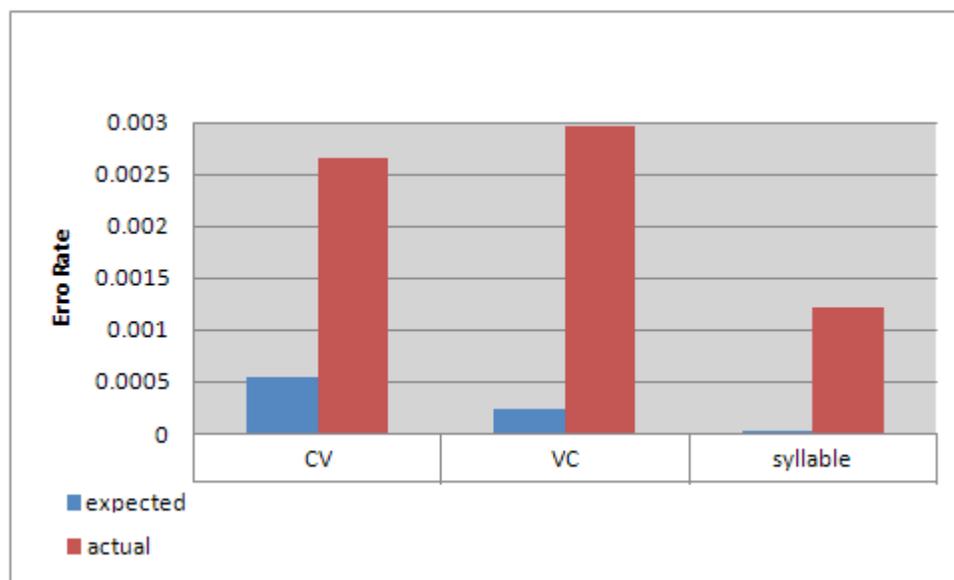


Figure 6-5: Independence of C and V errors

Contiguous errors were not evenly distributed between segment types, either. Errors on Cs frequently occurred as isolated errors, while errors on Vs occurred most often in conjunction with errors on one or both adjacent consonants (see Fig. 6-6).

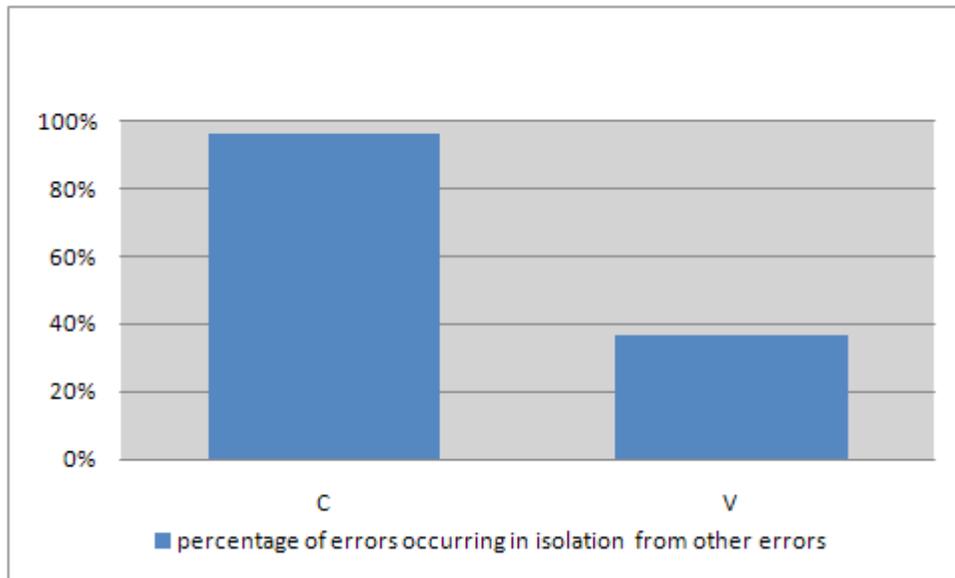


Figure 6-6: Occurrence of isolated errors

6.3 Model

The basic architecture of this model is shown in Figure 6-7. The model consists of four major parts: the oscillators and thresholded node pairs described in previous chapters; a layer of “tuned” nodes; one set of output nodes representing available onset and coda gestures; and a second set of output nodes representing available nuclear gestures. Figures 6-8 through 6-12 highlight these components one at a time with accompanying explanations.

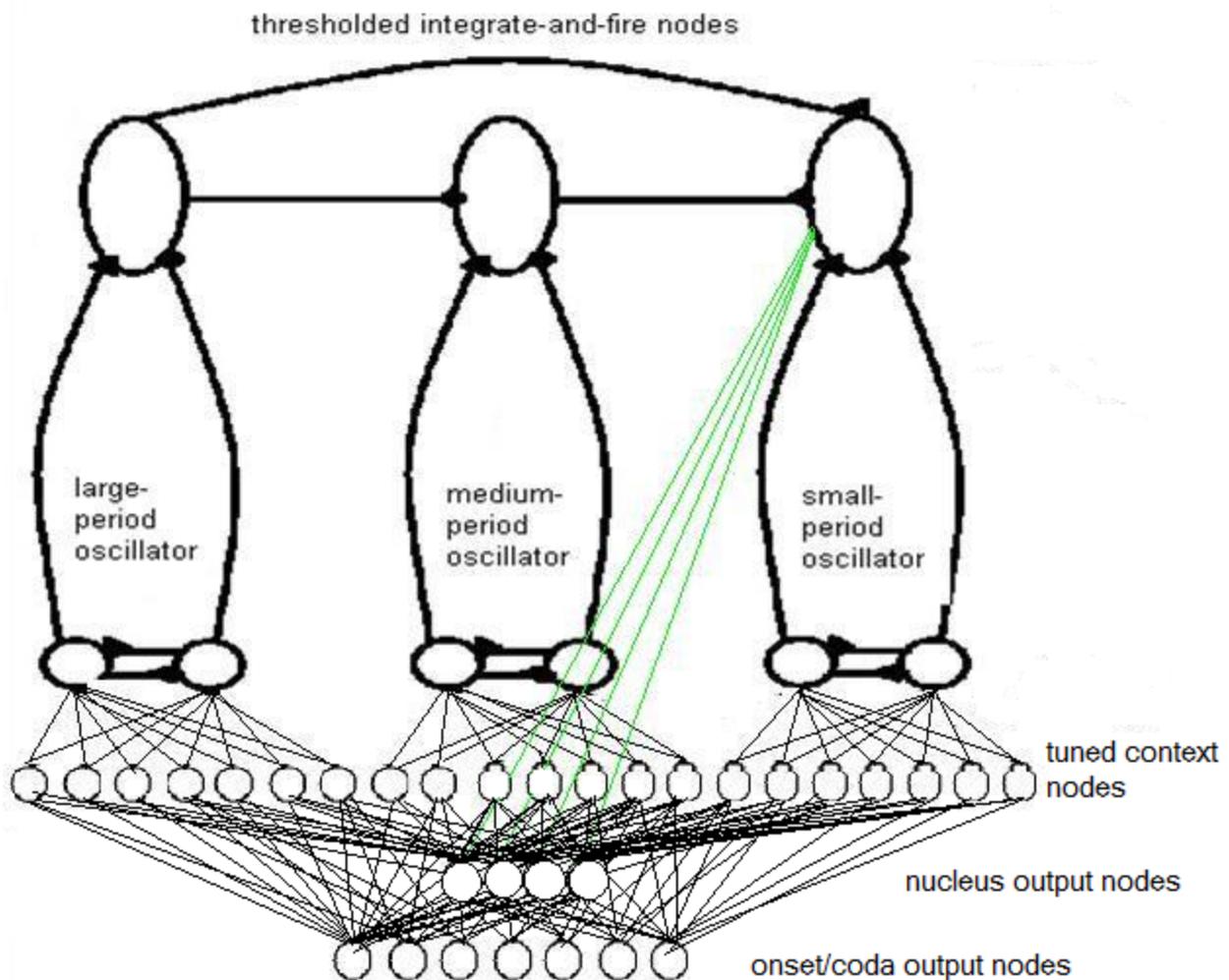


Figure 6-7: Diagram of the architecture of NOMSTR with the gesture string output extension

This model uses the oscillator-driven ANN from the first section of the thesis to provide the phonological context vector which drives its serial recall, taking the place of the oscillators in Vousden, Brown, and Harley's model or the suprasegmental nodes in Dell's model. In the proposed model, each oscillator can be set to represent the progression of specific size of prosodic chunk, such as the syllable, foot, and phrase, so that the phonological context each gesture node is associated with contains the

information of that gesture's appropriate location within its particular phrase, foot, and syllable (for example). Since in this model, unlike in OSCAR, the driving oscillations are produced by simulated half-cells, the oscillators themselves do not directly produce a sinusoidal output, but rather can induce sinusoidal activation levels in nodes to which they output (as described in Chapter 3). In order to activate in the correct sequence, each output node must be associated to a particular state of each of the three oscillations, its "address" in the sequences. However, integrate-and-fire neurons cannot have multiple levels of activation at the same time, so no output node can be associated with a particular level of activation from each of the three oscillators at once. In order to transform the output of the three oscillators into a context signal which can be "learned" by the output nodes, the NOMSTR extension uses a layer of tuned nodes between the oscillators and output nodes, as shown in Figure 6-8. Each node in this layer is tuned to a particular level of activation based on the combined excitation and inhibition from a particular oscillator pair. This means that each node is tuned to activate at a particular part of the cycle of either the fast, medium, or slow oscillator. A set of 7 nodes are tuned to activate at different points along the cycle of the small oscillator, a set of 7 are tuned to the medium oscillator, and a set of 7 are tuned to the large oscillator.³ These sets of nodes transform the three alternating outputs of the neural oscillators to single complex signal which changes over time, so that each time step has a unique pattern of activation and output in the tuned layer.

³ The number of tuned neurons per oscillator is not meant to be theoretically significant; the choice of 7 here is a practical one only, based on the requirements of simulating this data set. More tuned nodes would allow for a more fine-grained representation of the oscillations, which may be necessary for more complex productions than CVC syllable recitation.

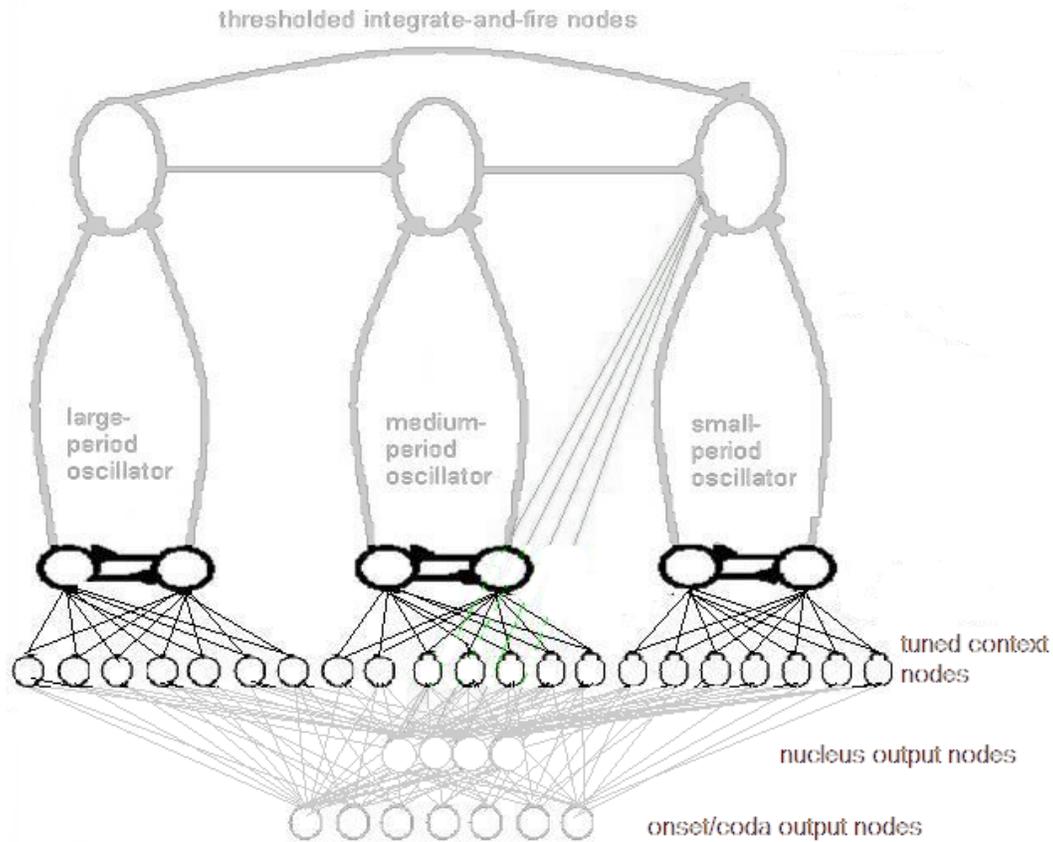


Figure 6-8: Oscillators and tuned neuron layer

Because each third of the tuned layer is tuned to one of the three repeating oscillators, time steps that are in the same locations in the cycle of one of the oscillators (e.g. in the case of the fast oscillator, time steps in the same syllable position of different syllables), have the same activation pattern in one third of the nodes. The changing pattern of activation of the tuned nodes is what Vousden, Brown, and Harley (2000) refer to as the “phonological context signal”. When each oscillator is set to represent a different prosodic level or structure (e.g. syllable, accentual phrase), then at similar locations in the phonological structure of an utterance (e.g. syllable onset, phrase final), the phonological context signal in the model is similar.

Below the layer of tuned nodes are two sets of gesture output nodes, the first for syllable nuclei, as shown in Figure 6-9. Each phone in the intended output sequence is represented by a particular pattern of activation in the nucleus output layer; in the case of the nucleus layer, each node represents a vocalic posture, so in the intended output only one nucleus output node is activated at once. Each pattern of activation in the nucleus output layer is associated via Hebbian learning to a particular activation pattern in the tuned phonological context layer. During the recall phase, as the oscillators proceed through time, they cause the tuned layer to produce a changing activation pattern, which causes the output layer to produce the intended activation patterns in sequence due to the learned associations between the tuned and output layers.

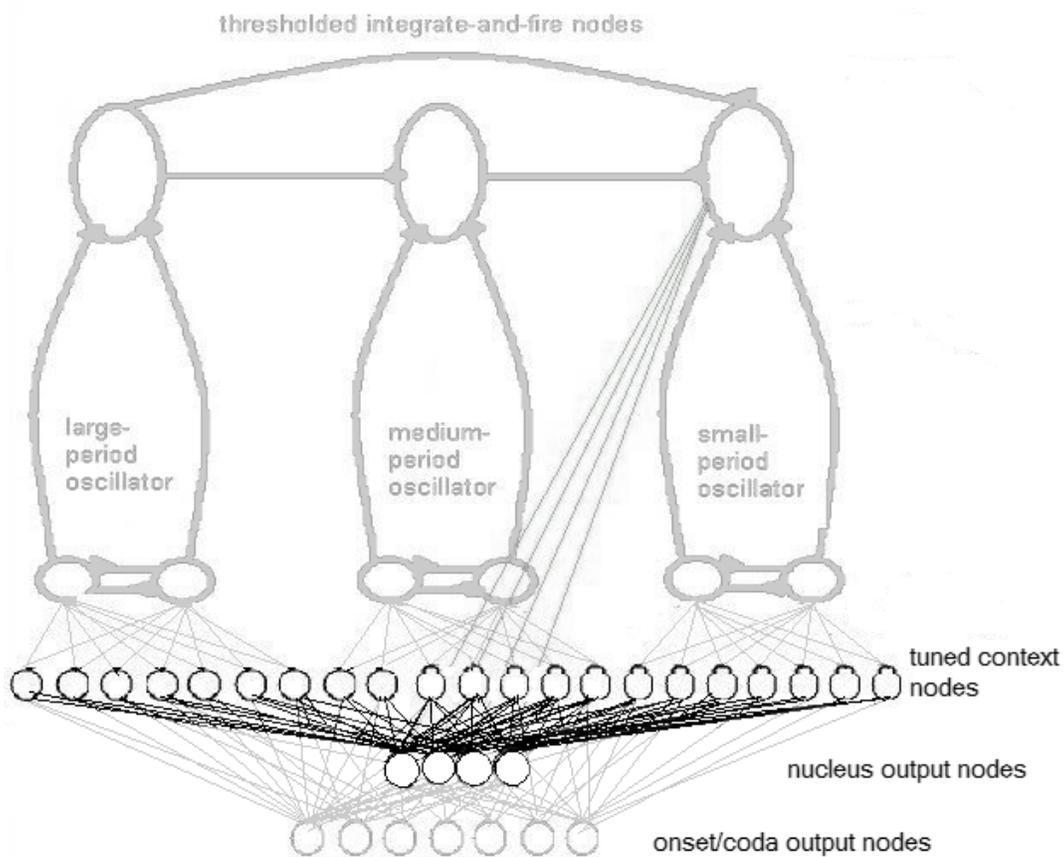


Figure 6-9: Tuned neuron layer and nucleus output layer

The tuned layer is associated with the layer of onset/coda output nodes (i.e. consonant nodes) in the same way that it is associated with the layer of nucleus output nodes (see Fig. 6-10). There is one important difference: a phone in the intended output sequence may be represented by the activation more than one consonant gesture node at once.

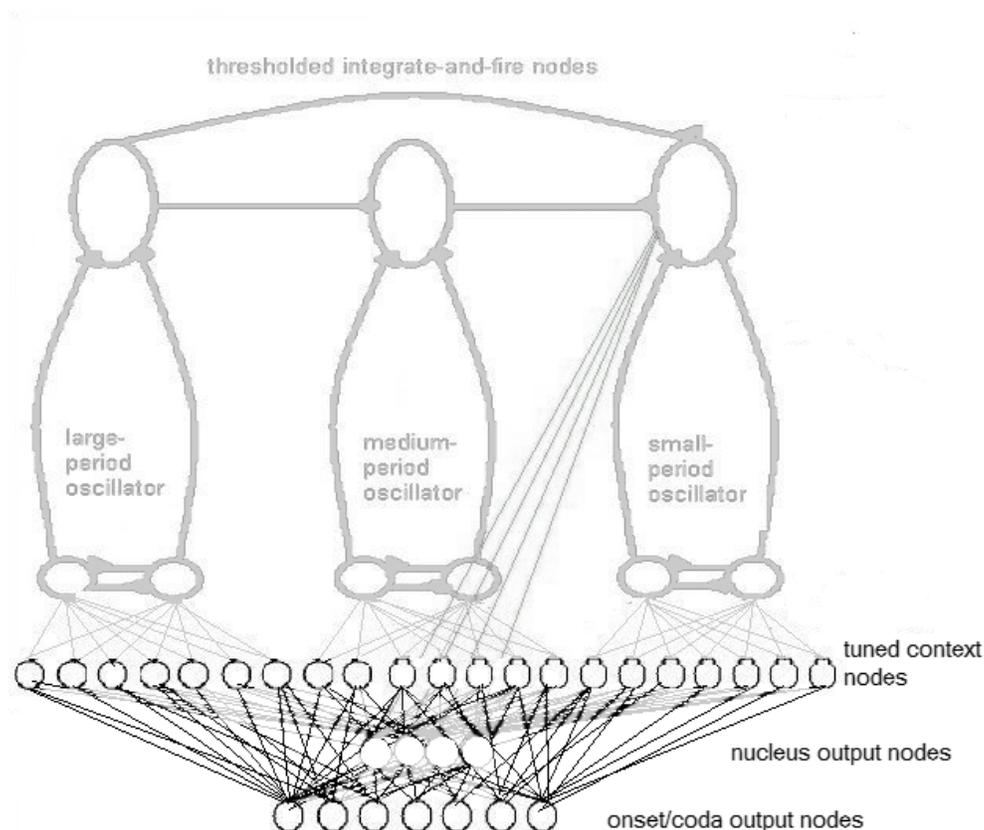


Figure 6-10: Tuned neuron layer and onset/coda output layer

Besides the prosodic context signal output by the tuned nodes, the gesture output nodes have two other sources of input. Nuclear gesture nodes provide excitatory input to the nodes required to produce the onset and coda gestures they share a syllable with, as shown in Figure 6-11. This mechanism is the ANN instantiation of the nucleus-

centered syllable model. It can also be seen as a very basic representation of a type of syllable inventory, which contains information about the gestural content of each syllable.

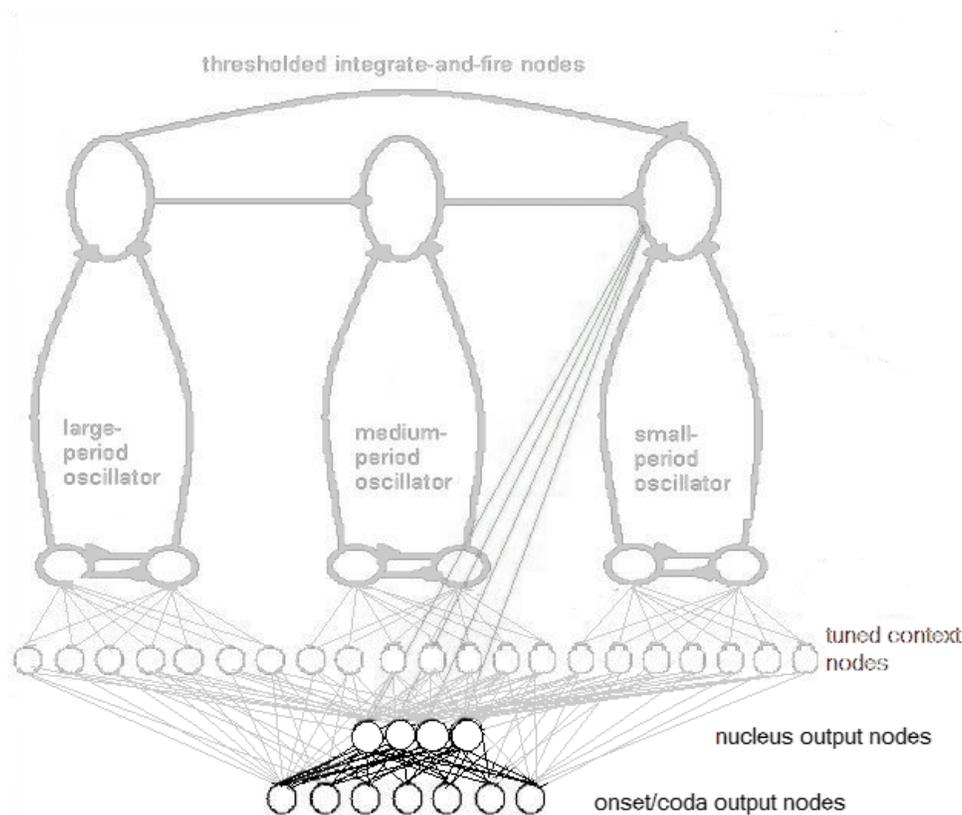


Figure 6-11: Nucleus output layer and onset/coda output layer

Second, nuclear gesture nodes receive excitatory input from NOMSTR’s syllable-level thresholded node, resulting in an activation “kick” at the time of the syllable-level node’s peak (see Fig. 6-12). This ties the timing of the articulation of the gestures in a syllable (particularly the syllable c-center) to the time at which the syllable “occurs” in the phonological output. Although not strictly necessary for speech error simulation, this is the mechanism that would allow the subjects in this study to entrain their production of the syllables to the metronome.

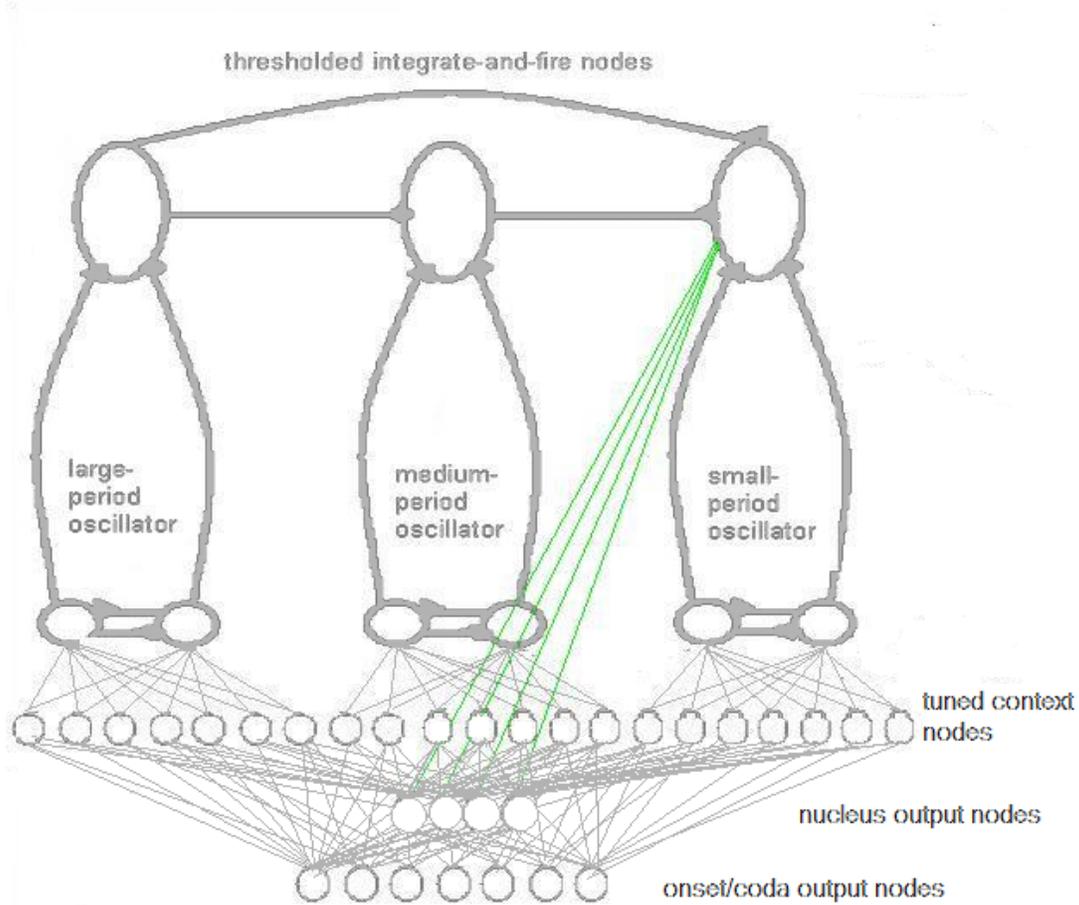


Figure 6-12: Excitatory connections between the syllable thresholded node and nucleus output nodes

This ANN model adopts many of the concepts behind the Articulatory Phonology (AP) models of the syllable and speech production: the atomic elements of the syllable output by the model are gestures, several of which may be activated at once; errors are modeled as arising from intrusive activation of erroneous gestures; and the nuclear vowel gesture is the coordinative center of the syllable. Nuclear gestures begin to activate at the end of the preceding syllable, and maintain activation during and after the activation of the onset gestures; and all of the gestures necessary for an onset or coda consonant are activated at the same timepoint (although the actual articulation of

co-activated gestures' may not be perfectly simultaneous, due to the articulatory dynamics between the gestures themselves). Errors in the simulation are produced when an erroneous gesture output node is at a higher level of activation than at least one of the appropriate output nodes for that timepoint.

In order to induce model to make errors, noise was added to each of the output nodes in the form of a noise variable, N , a pseudorandom value drawn from a normal distribution with a standard deviation of 8. The noise level in the system can be manipulated by adjusting N 's distribution, but all output nodes (both nuclear and onset/coda) use the same noise distribution. Nucleus output nodes used the equation shown in A, while onset/coda nodes used equation B.

A) activation = excitement from phonological context + excitement from syllable-oscillator thresholded node + N

B) activation = excitement from phonological context + excitement from syllable nucleus node + N

6.3.1 Model Output

Figures 6-13 and 6-14 show the output of the nucleus and onset/coda arrays, respectively, for the first half of simulated production of the nonce phrase *fung dis ban tom* ([fʌŋdɪsbæntəm]). These figures show an example of accurate recall, in which the model reproduces the intended output with no errors.⁴

⁴ As a note, the outputs of the two layers do not compete; this model is not designed to simulate errors in which a nuclear gesture is erroneously produced in onset or coda position, or vice versa.

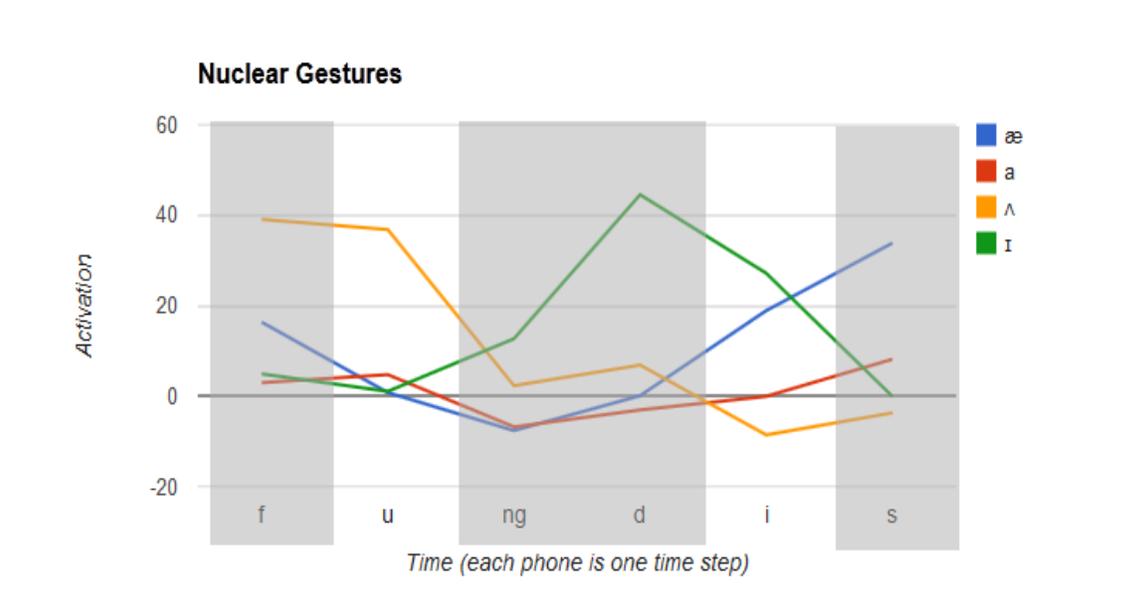


Figure 6-13: Example output of nucleus output nodes

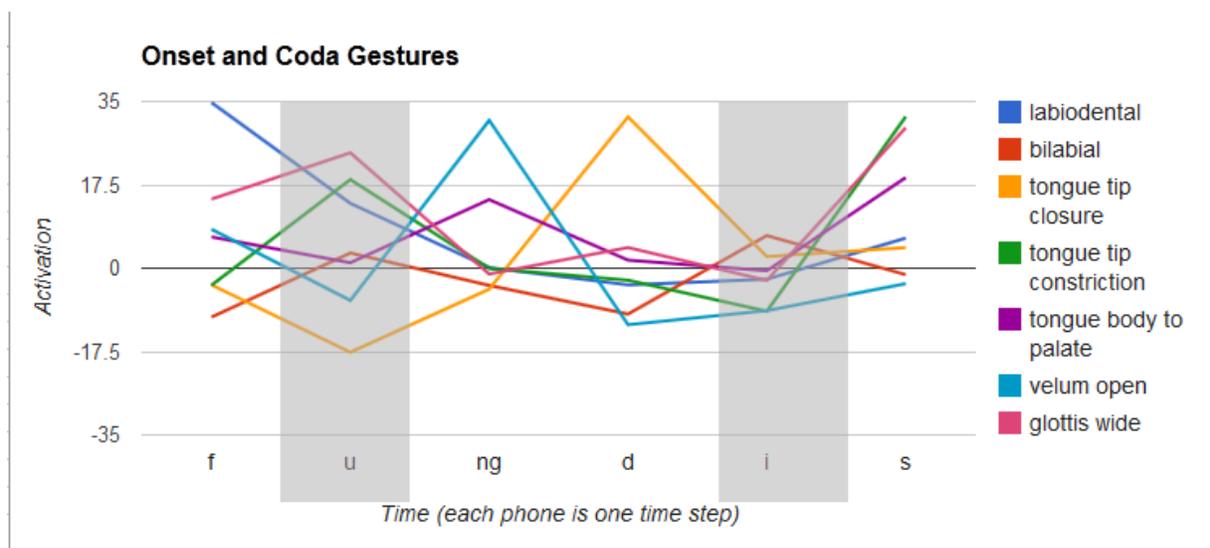


Figure 6-14: Example output of onset/coda output nodes

As mentioned earlier, an error in this simulated output is defined as occurring any time a node representing an erroneous gesture or vowel posture is more highly activated than one of the intended gestures for that time slot.⁵ For example, Figure 6-15 below shows an output of [s] with no error; the gestures encoded to produce an [s] are tongue tip constriction and glottis wide, which are correctly reproduced as the most highly activated gestures at the sixth time slot of the simulated phrase.

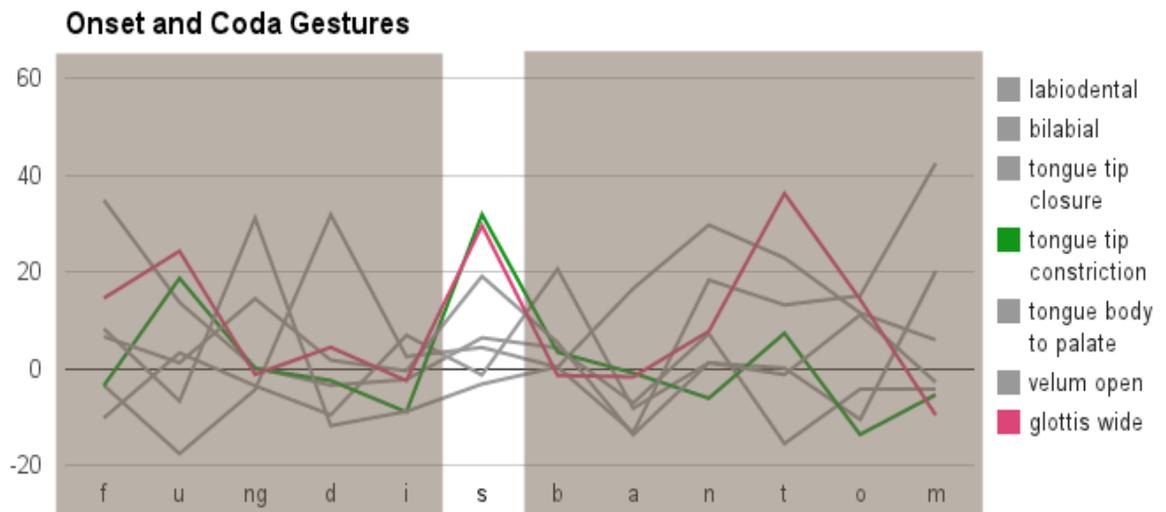


Figure 6-15: A correctly simulated [s]

Figure 6-16 shows an erroneous production of an intended [s] in which the tongue body to palate gesture is more highly activated than the tongue tip constriction gesture. Although this model cannot simulate the articulatory or acoustic effects of intrusive gestures, it is reasonable to assume that a production of [s] with an intrusive tongue body to palate constriction may result in something more like a [k].

⁵ When an erroneous gesture node matches the activation level of an intended gesture (within one point), an error was not recorded, although these “ties” were tallied separately.

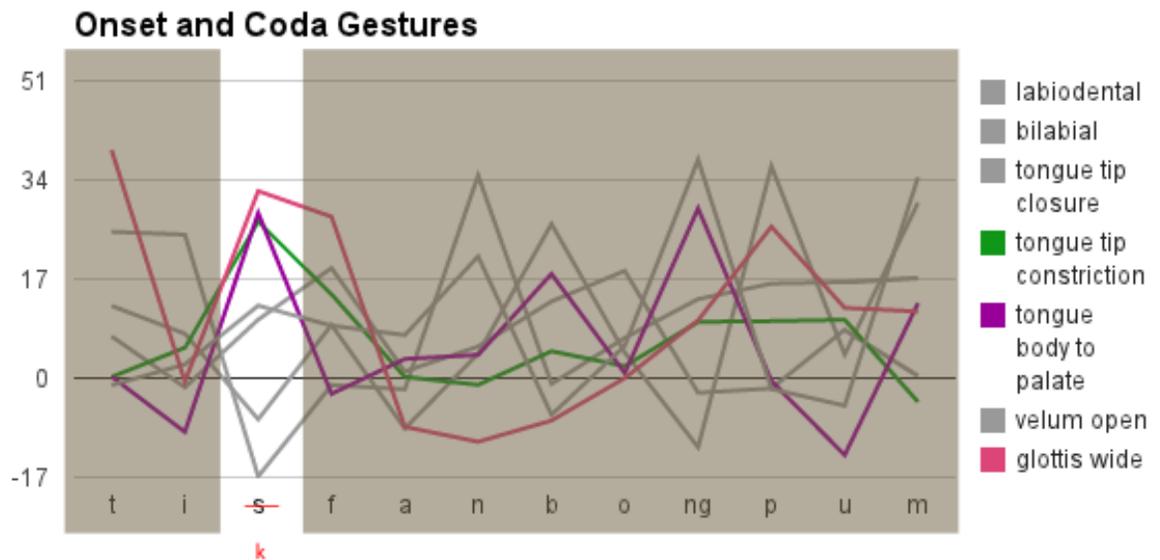


Figure 6-16: An erroneously simulated [s]

6.3.2 Simulation Results

The extended NOMSTR model was used to simulate 270 nonce phrases, or 1080 syllables, from the materials used in the speech error production study. Only phrases used in the production materials were used in the simulation materials, and every phrase occurring at least once in the production materials occurs in the simulation materials. Overall, the model produced 127 errors; as in the production study results the simulated errors were asymmetrically distributed. The simulation reproduced the disproportionately greater rate of errors on consonant (onset/coda) timesteps than on vowel (nuclear) timesteps found in the original data (see Figures 6-17 and 6-18). Errors on nucleus time steps occurred at a rate of .007, and errors on onset and coda timesteps occurred at a combined rate of .055, nearly matching the vowel and consonant error rates from the production study of .007 and .057. It is important to note that although the error rates can be manipulated in a general sense, by adjusting the noise variable, the nucleus and onset/coda output nodes all use the same variable, so the comparative difference between nucleus and onset/coda error is produced by the architecture of the

model itself, not by adjusting the noise variable. The input from nucleus nodes to onset/coda nodes in “their” syllables may account for some of the difference; besides errors caused by endogenous noise, onset/coda errors may be caused by an erroneously activated nucleus node exciting unintended onset/coda nodes. It is also possible that the nucleus output is inherently more robust, having fewer competitors, longer activations, and a less complex intended output (one activation at a time).

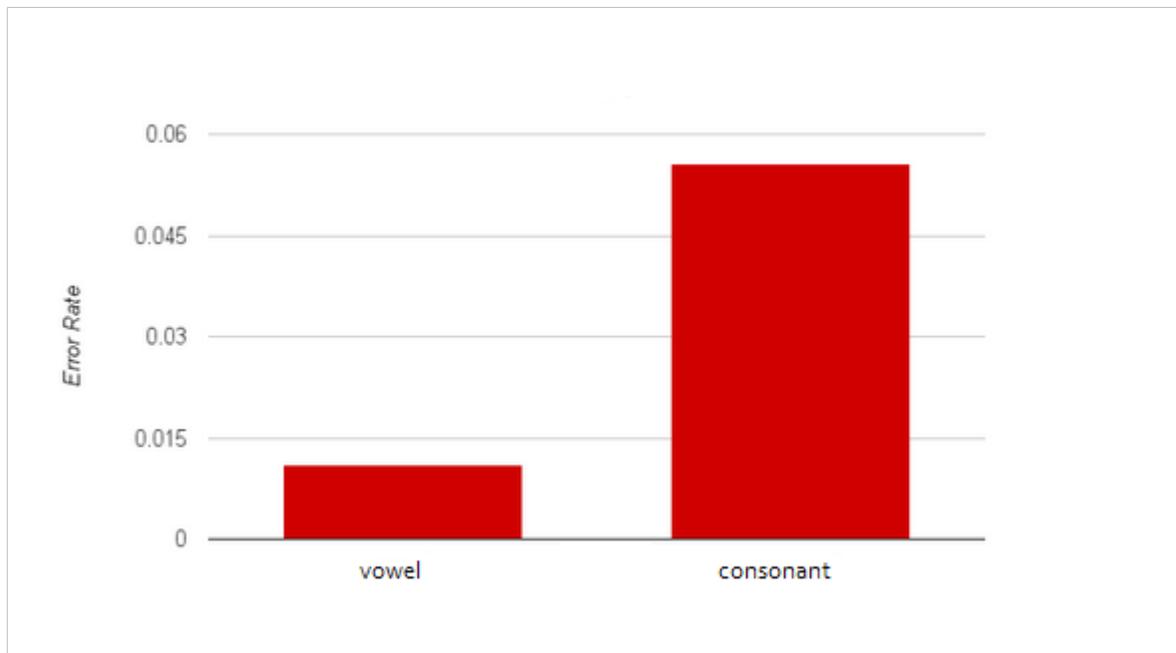


Figure 6-17: Simulated errors by segment type

As Figure 6-17 shows, the simulation was not able to reproduce the asymmetry in error rates between the onset and coda positions. This suggests that the model may be improved with the addition of a mechanism for decreasing error rates as the syllable progresses.

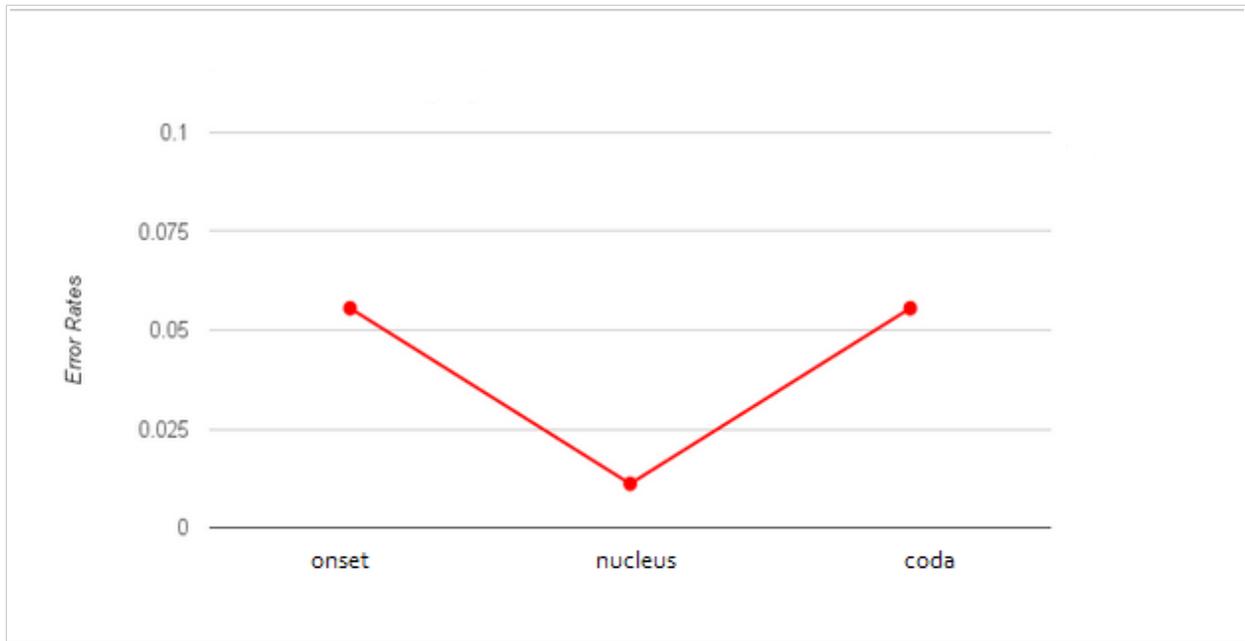


Figure 6-18: Simulated errors by syllable position

As expected due to the influence of syllable nucleus nodes on onset/coda nodes, errors on contiguous VC and CVC sequences occurred more often than by chance (Figure 6-19)⁶. CV errors did not show this influence, however, possibly due to the very small numbers involved: even if contiguous CV errors occurred at double the rate of chance, that only represents an average of .86 CV errors per 1080 syllables (the size of the simulated data set).

⁶ As in the production data, chance of cooccurrence here is calculated by multiplying together the occurrence rates of each type of error.

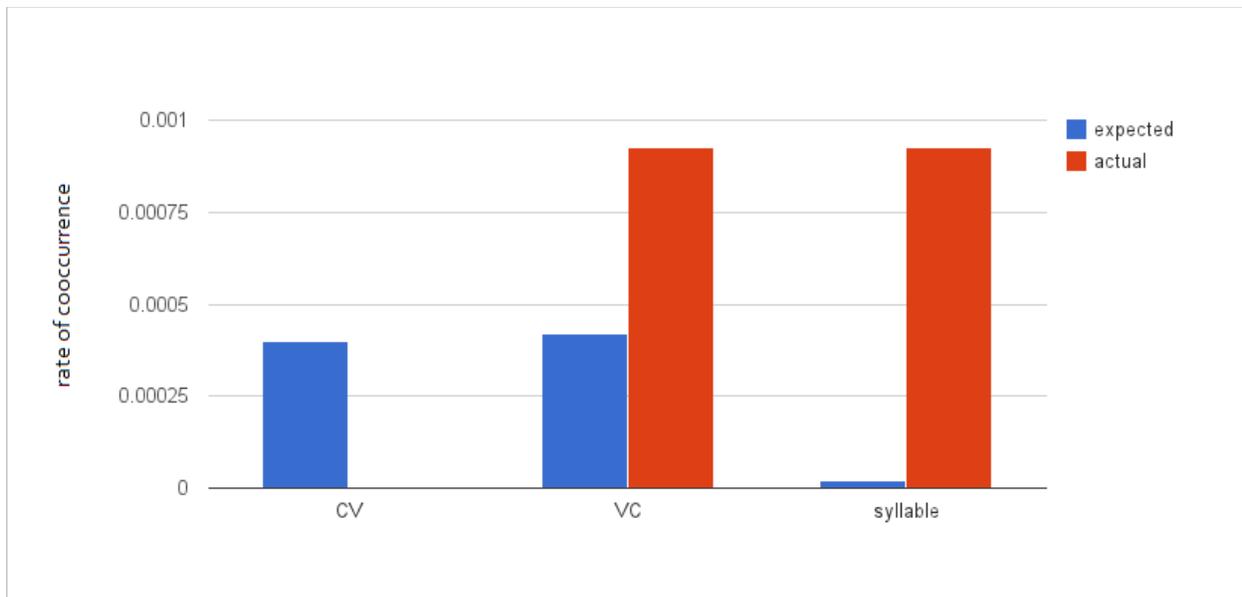


Figure 6-19: Independence of simulated C and V errors

As in the production data, contiguous errors were not evenly distributed by segment type, with errors on onset or coda time steps more likely to occur in isolation (Figure 6-20). However, a higher percentage of vowel errors occurred in isolation in the simulation than in the data, which may suggest heavier influence of the nuclei on the onsets/codas than used in this version of the model.

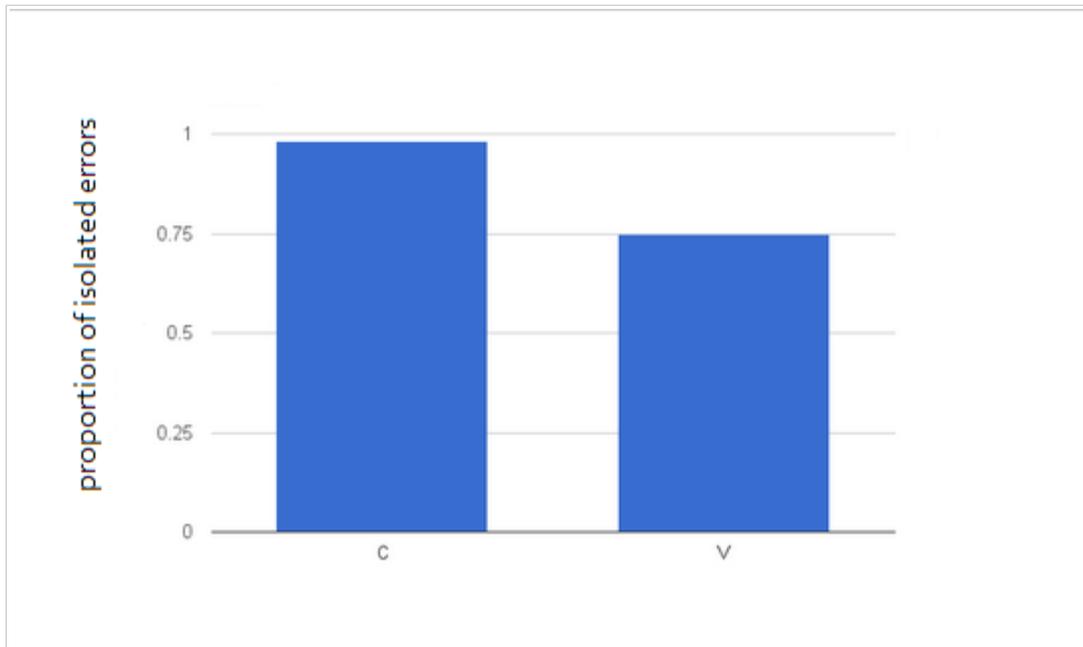


Figure 6-20: Occurrence of isolated simulated errors

6.4 Discussion

Serial access models of speech production have been fairly successful at simulating many aspects of speech error behavior. NOMSTR can capture many of the benefits of other oscillator-based and neural network models of serial recall in speech production using only 3 neural oscillators, many fewer than used by previous models (e.g. Vousden, Brown, & Harley 2000, Harris 2002). One of the benefits of using NOMSTR's system of relatively few oscillators is that the same oscillators can be used to model the timing and progression of suprasegmental elements of speech production, as well as timing the sequential production of the segments in an utterance.

The extension to the core NOMSTR model described in this chapter represents a neural network implementation of the Articulatory Phonology model of syllable organization. That syllable model was selected for implementation because it explained the speech error results from the production study. The modeling work demonstrated in this chapter demonstrates how certain concepts from AP research can be integrated into a model of serial speech production: onsets, codas, and nuclei are composed of gestures

of variable length; gestures are not strictly sequential and may overlap; the nuclear gesture is the organizational center of the syllable upon which timing of the onset and coda gestures rely; and most speech errors are due to the intrusion of erroneous gestures. A serial model of speech production with these properties can be integrated into a model of serial speech production in order to simulate aspects of speech error behavior which other models have been unable to explain, such as C/V error (O-C/N) asymmetry, and error dependence. However, NOMSTR was unable to reproduce asymmetries in the probability of onset vs. coda errors, suggesting that some type of cascading connections as in Dell's model may be necessary for more accurate simulations, although whether chaining across the duration of an utterance or merely within a syllable (or word) is appropriate cannot be determined from these data.

The ability of the NOMSTR model, originally designed to model prosodic structure, to also successfully “power” a model of sequential gesture production, is evidence of the power of the combination of oscillators and a neural network to simulate speech behavior. The nature of neural networks, in particular, allows the NOMSTR core model to interact with models of other aspects of the speech production and planning system, if they are implemented as ANNs.

Chapter 7

Conclusions

This thesis project set out to contribute to understanding of how levels of phonological structure are coordinated and to explore the development of a computational model combining oscillators with a neural network for the purpose of modeling those interactions. In this chapter I will first summarize the findings of the empirical studies and modeling work presented in earlier chapters, and discuss their contributions to our understanding of rhythm, prosody, speech errors, and speech production modeling. I will discuss the implications my findings have for current theories of speech production, as well suggesting directions for future research in this area.

7.1 Summary of Findings

The phonological structure is multi-leveled and complex; it encodes information about several levels of phonological organization (the gesture, syllable, stress or accent group, and phrase) into a single behavioral output, the articulatory string. In this work I have investigated how these different levels of phonological organization are coordinated in speech production by measuring the effects of that coordination on acoustic and behavioral measures of speech production, and by modeling those coordination patterns using a central pattern generator (CPG) model, a specific type of artificial neural network (ANN).

A model based on CPG models of other types of patterned animal behavior was selected for construction and testing for both practical and theoretical reasons. On the practical side, a COG-inspired model combines the simulation power of two types of models which have been successfully used to model different aspects of speech production. Oscillator models have been used to model inter-gestural relationships and the relationships between syllables and stress feet, as well as to simulate sequential

timing of speech segments. ANNs have long been used to model several aspects of speech planning and production, including sequential segment timing. On a theoretical level, a model based on other neural networks which drive other, non-linguistic behavior, was chosen as a way to help incorporate the study of speech production into the wider study of animal motor behaviors, and reach towards more biologically plausible models of linguistic structure as it is stored and produced in the human brain.

This dissertation demonstrates how an artificial neural network which incorporates oscillators can be a powerful tool for modeling the interactions between elements of phonological structure and for stimulating speech timing patterns, and additionally that a common underlying model architecture can be useful for stimulating multiple languages and multiple levels of structure. Chapter 3 introduced the CPG-inspired ANN model NOMSTR which was designed to model three levels of rhythmic structure using oscillators, and interactions between these levels using excitation, inhibition, and thresholding. Chapter 4 looked at how the model could be used to simulate the prosodic timing of French and English utterances. I found that by manipulating the parameters of the model—oscillator frequencies and phases and the connections between nodes of the network—the model architecture described in Chapter 3 can be used to simulate the rhythmic structures of both French and English utterances, specifically the influence of phrasing and accenting on syllable durations, and the location of syllables influence by accents and phrase boundaries. Chapter 5 focused on the relationship between the syllable, stress group, and phrase, in terms of their temporal coordination, and how this relationship differs between French and English, languages with differing prosodic systems. Results of simulations show that while in English, there is evidence that IP boundaries have an enhancing influence on accentual lengthening, in French Initial Accent lengthening was *less* likely to occur closer to an IP boundary, indicating some inhibitory or “push-back” effect. The ANN was shown to be successful in modeling these interactions and simulating their effects on syllable durations. Chapter 6 described an expanded model which was designed to use the original oscillator-based network to power an ANN for serial encoding and recall of speech gestures, and showed how incorporating new ideas about the nature of the

syllable's structure and components from Articulatory Phonology (AP) research could allow the model to simulate asymmetries in speech error probabilities across the syllable which previous models had not accounted for.

7.2 Conclusion

This thesis contributes to our understanding of speech rhythm and timing by presenting a universal model architecture with which to simulate the speech timing of various languages, providing a common means of describing and quantifying the differences between them (like some other oscillator models), and providing a way to understand which aspects of speech timing are universal and shared, and which are flexible or language-specific. In opposition to widely-used models of prosodic structure, it suggests that there may not be a categorical difference between phrasing and accenting, so they can be modeled in the same way, although they are used for different purposes; and also that accents and phrase boundaries can both be conceptualized as having durations as well as dynamic intensity. It corroborates studies on cycled and simultaneous speech which show that speech production has a tendency to be rhythmic, and shows how the idea of temporal regularity can still be useful for modeling spontaneous or natural speech, even while integrating temporal flexibility and discarding the idea of true isochrony. The serial recall extension of NOMSTR demonstrates how the AP models of syllable structure and speech segments/errors on speech segments can be incorporated into a neural network model of serial production/erroneous production to contribute to the success that type of model has had simulating speech errors.

7.3 Future Work

The work presented here describes the initial development and testing of the neural oscillator model and its usefulness in understanding speech timing. Much work is still to be done to make the model more accurate and useful. First, improving the dataset used to build the model by adding new types of data and data from additional languages could improve the model's accuracy and explanatory power. Attempting to simulate articulatory speech production data may help uncover aspects of speech

production planning that the model should take into account which are not obvious from the acoustic data analyzed in this thesis. Similarly, the simulation work on prosodic timing in this dissertation focuses mostly the effect on syllable durations of interactions between prosodic levels in order to understand their interactions; investigating the effects on pitch contours could bring a more complete understanding of prosodic timing, and thus a better model. Simulation work should also be attempted on additional languages, especially languages such as Japanese and Mandarin, which have aspects to their prosodic systems not shared by French or English (i.e. moraic timing and lexical tones). In addition, the further development of this model and evaluation of its abilities to simulate different languages will require the development of some type of benchmark against which model performance can be measured. This evaluation could take the form of some comparison between the model's current and future performance, or some independent measurement of the model's ability to explain the data it simulates, although the latter may be difficult, given that there do not exist any other models which simulate exactly the same data against which NOMSTR could be compared.

Second, the tests of the model which are already complete suggest some practical improvements which could be made more immediately: 1) although modeling three levels of prosodic structure was fairly successful at simulating the data presented here, more than three levels of suprasegmental prosodic structure have been proposed for both English and French, so adding more levels of oscillators to the model could make it more theoretically accurate and possibly produce more accurate simulations; and 2) the plausibility of this model as a model of a neural network could be improved by research into and application of more realistic equations to govern the behavior of the artificial neurons; 3) construction of a more user-friendly interface for the model would allow for faster simulation and experimentation and transform it into a tool which could be made available for other researchers.

Appendix A

Production Studies

A.1 Utterance Repetition Study

A.1.1 Transcripts of Utterances from Corpora

A.1.1.1 English Utterances from the American English Map Task Corpus

Um, Kate, do you see the Canadian paradise?

[ʌm keɪt̩ dʒu si ðə kənɛɪdɪŋ pɛɪrədɑɪs]

And follow that path right above the fenced meadow.

[ænd fəloʊ ðæt pæθ raɪt əbʌv ðə fɛnst mɛdoʊ]

No, but you go to the left of the antelope.

[nɒʊ bʌt ju goʊ tu ðə lɛft əv ði æntələʊp]

A.1.1.2 French Utterances from the Rhapsodie Corpus

Note: IPA converted from original Rhapsodie transcription, which uses the Rhapsodie transcription protocol to replace some IPA characters. Information about the transcription protocol can be found on the Rhapsodie website: [http://www.projet-rhapsodie.fr/tuto/Codage Prosodique.pdf](http://www.projet-rhapsodie.fr/tuto/Codage%20Prosodique.pdf)

Pendant un petit moment vous allez toujours aller tout droit.

[pɑ̃ dɑ̃ ɛ̃p ti mɔ̃ mɑ̃ vu za le tʒu ʁa le tu dʁwa]

Et le, le porche vous pouvez pas louper i le juste en face.

[e lɔ̃ lə pɔʁʃ vu pu ve pa lu pe i le ʒys tʃɑ̃ fas]

Vous allez passe devant la poste qui sera a votre droit.

[vu za le pa se də vɑ̃ la pɔst ki sɛra a vɔt dʁwat]

A.1.2 Subject Instructions

A.1.2.1 Instructions for French Participants

Dans cette partie, vous entendrez des phrases parlées. Cliquez sur le bouton pour entendre une phrase. Après avoir entendu une phrase, attendez cinq secondes, puis répéter les mots et la façon dont la phrase a été dite. Attendez cinq secondes de plus, puis répéter la phrase de nouveau. Attendez cinq secondes de plus, puis répéter la phrase une troisième fois. Alors écoutez la phrase suivante.

A.2 Accent and Boundary Lengthening Study

A.2.1 Materials

A.2.1.1 English

Trisyllabic words with primary stress on the second syllable, reduced first and last syllables

- Please say ‘Jamaica or Dakota and Chicago will play’.
- Please say ‘Manassas or Pacific and Poseidon will play’.
- Please say ‘MacPherson or MacWalden and MacDouglas will stay’.

Trisyllabic words with primary stress on the first syllable, reduced last syllable

- Please say ‘Michigan or Madison and Vatican will play’.
- Please say ‘Mendelson or Morganton and Palmerson will stay’.

Trisyllabic words with primary stress on the first syllable, full last syllable.

- Please say ‘Trinidad or Rotterdam and Lebanon will play’.
- Please say ‘Benedict or Cheddarfield and Bonaparte will play’.
- Please say ‘Thomasburg or Terraceville and Watertown will play’.

Four-syllable words with primary stress on the third syllable

- Please say ‘Mississippi or Mogadishu and Honolulu will play’.
- Please say ‘Nagasaki or Winnebago and Madagascar will play’.
- Please say ‘Carolina or Nicaragua and Manitoba will play’.
- Please say ‘Massachussetts or Chappaquiddick and Athabaskan will play.’

A.2.1.2 French

Three-syllable target words

- Les termitières et les fourmilières napolitaines, en fait, sont souvent méconnues.
- Les bonimenteurs et les baratineurs fabuleux, en fait, sont plutôt répandus.
- Les tranquillisants et les barbituriques salutaires, en fait, sont relativement rares.
- Les débordements et les déferlements fantastiques, en fait, sont assez appréciés.
- Les infamies et les barbaries soldatesques, en fait, sont relativement rares.
- Les termitières et les fourmilières naturelles, en fait, sont souvent méconnues.
- Les désertions et les défections dissidentes, en fait, sont assez appréciées.
- Les bagatelles et les balivernes saugrenues, en fait, sont plutôt répandues.
- Les publications et les filmographies militantes, en fait, sont souvent méconnues.
- Les infamies et les barbaries certifiées, en fait, sont relativement rares.

Four-syllable target words

- Les désertions et les défections disciplinées, en fait, sont assez appréciées.
- Les bagatelles et les balivernes somnambuliques, en fait, sont plutôt répandues
- Les publications et les filmographies mirobolantes, en fait, sont souvent méconnues
- Les bonimenteurs et les baratineurs fabulateurs, en fait, sont plutôt répandus.
- Les débordements et les déferlements fantomatiques, en fait, sont assez appréciés.
- Les tranquillisants et les barbituriques salicyliques, en fait, sont relativement rares.

A.2.2 Instructions and Consent Forms

A.2.2.1 Experimenter Instructions Including Verbal Instructions Given to English Subjects

Instructions for French Prosodic Timing Studies

General procedure of experiments:

- participants see each sentence and then speak it aloud

Before the participant arrives:

- make sure the digital recorder is on the correct settings and that you are beginning a new file
- reset the PowerPoint presentation and ensure speakers/earphones and sound are working

When the participant arrives:

- hand them the informed consent form and verbally go over what it says, then allow them to read and sign it, and give them a copy
- ensure that their cell phone is turned off

Sit them at the computer and tell them before they begin:

- in the second section they are going to read a sentence, then say it aloud
- the parenthesis tell them how to group the words in the sentence
- read the entire sentence first, then say it aloud, naturally

After the experiment is complete:

- pay them and have them sign the receipt
- answer any questions they have about the hypothesis being tested, expected results, etc

A.2.2.2 Instructions for French Participants

Dans cette partie, vous lirez une phrase et puis le direz à haute voix. Les parenthèses vous dire comment regrouper les mots dans la phrase. Par exemple: Si les bas sont lisses, mais les gants ne sont pas lisses: Les gants et les bas lisses, en fait, sont relativement rares. Si les gants et les bas sont lisses: Les gants et les bas lisses, en fait, sont relativement rares. Commencez par lire toute la phrase, puis le dire naturellement.

A.2.2.3 English Subject Consent Form

Informed Consent

Reading aloud task

Department of Linguistics
University of Illinois at Urbana-Champaign
4088 Foreign Languages Building, MC-168
707 South Mathews Avenue
Urbana, IL 61801-3625

Study of native speakers' speech rhythm in English **Directed by Prof. Jennifer S. Cole, graduate student Erin Rusaw**

Please read this consent agreement carefully. You must be 18 years old or older to participate.

Purpose of the research: We are studying how speakers of English produce spoken phrases because we are interested in how speech rhythm and timing differs between languages. The purpose of the study is to advance scientific knowledge about how people use language.

What you will do in this study: In this experiment, you will be presented with written sentences or phrases in Korean/English. You will have to read them aloud. Your speech will be digitally recorded for further analysis. The session is expected to take less than half an hour to complete.

Risks: There are no anticipated risks associated with participating in this study, beyond those of daily life.

Benefits: This research will provide valuable data on language use which will contribute to basic linguistic science.

Compensation: There is no monetary/other compensation for your participation.

Voluntary Withdrawal: Your participation in this study is completely voluntary, and you may withdraw from the study at any time without penalty by informing the research associate that you no longer wish to participate no questions will be asked. Your decision to participate, decline or withdraw participation will have no effect on your status at or relationship with your educational institution if applicable.

Exclusion of Data: You may request to have any of the responses submitted by you deleted without penalty. Those responses would then be excluded from the study.

Confidentiality: Your participation in this study will remain confidential, and your identity will not be stored with your data. Your recording will be associated with a code number and your name or other identifying information will not be stored together with your speech data. All data will be stored in the investigator's password protected computer storage drive. Consent forms will be stored in a locked office or laboratory. Data records will be retained for at least 5 years following publication of the results. Results of this study may be presented at conferences and/or published in books, journals and/or in the popular media.

Further information: If you have questions about this study, please contact Erin Rusaw [erusak2@illinois.edu, 352-514-7602] or Jennifer Cole [jscole@illinois.edu, 217-244-3057]; Department of Linguistics, University of

Illinois, Urbana, IL 61801. If you have any concerns about this study or your experience as a participant, you may contact the Institutional Review Board IRB at UIUC at 217333- 2670 collect calls will be accepted if you state you are a study participant; email: irb@illinois.edu.

Agreement: The purpose and nature of this research have been sufficiently explained and I agree to participate in this study. I give permission to record my voice and use the recordings in preparation and presentation of this and other research reports. I understand that I am free to withdraw at any time without incurring any penalty.

____ initial I give permission for excerpts from my recorded speech to be played at scholarly presentations derived from this research.

Signature: _____ **Date:** _____

Name print: _____

A.2.2.4 French Subject Consent Form

Consentement éclairé

Lecture à haute voix

Department of Linguistics
University of Illinois at Urbana-Champaign
4088 Foreign Languages Building, MC-168
707 South Mathews Avenue
Urbana, IL 61801-3625

Etude du rythme de la parole des locuteurs natifs en français **Réalisé par le professeur Jennifer S. Cole, étudiant au doctorat Erin Rusaw**

S'il vous plaît lisez ce consentement attentivement.

Vous devez avoir 18 ans ou plus pour participer.

But de la recherche: Nous étudions comment les locuteurs du français produisent phrases parlées, car nous sommes intéressés à la manière dont le rythme de la parole diffère entre les langues. Le but de l'étude est de faire progresser les connaissances scientifiques sur comment les gens utilisent la langue.

Qu'est-ce que vous ferez dans cette étude: Dans cette expérience, il vous sera présenté avec des phrases écrites ou des phrases en français. Vous aurez à les lire à haute voix. Votre discours sera enregistrée numériquement pour une analyse ultérieure. La session devrait se tenir à moins de 30 minutes à compléter.

Risques: Il n'ya pas de risques anticipés associés à participer à cette étude, au-delà ceux de la vie quotidienne..

Avantages: Cette recherche fournira des données précieuses sur l'utilisation des langues qui contribueront à la science linguistique fondamentale.

Compensation: Vous serez payé €6 à la fin de la session expérience. Si vous retirez avant l'achèvement de l'étude, vous ne recevrez pas de compensation partielle.

Retrait Volontaire: Votre participation à cette étude est entièrement volontaire. Vous pouvez retirer de l'étude à tout moment. Si vous ne souhaitez plus participer, raconter l'associé de recherche pas de questions seront posées. Votre décision de participer, de refuser ou retirer votre participation ne sera pas d'avoir un effet sur votre état à votre établissement d'enseignement le cas échéant.

Exclusion des données: Vous pouvez demander à avoir vos réponses supprimés sans pénalité. Ces réponses seraient alors exclus de l'étude.

Confidentialité: Votre participation à cette étude resteront confidentielles. Votre identité ne sera pas stocké avec vos données. Votre enregistrement sera associé à un numéro de code. Votre nom ou autres informations d'identification ne seront pas stockées avec vos données de parole. utes les données seront stockées sur protégé par mot d'un disque d'ordinateur. Les formulaires de consentement seront stockées dans un bureau verrouillé ou en laboratoire. Enregistrements de données seront conservées pendant au moins 5 ans suivant la publication des résultats. Les résultats de cette étude pourraient être présentés lors de conférences et / ou publiées dans des livres, des revues et / ou dans les médias populaires.

Plus d'informations: Si vous avez des questions concernant cette étude, s'il vous plaît contactez Erin Rusaw [erusaw2@illinois.edu, 352-514-7602] ou Jennifer Cole [jscole@illinois.edu, 217-244-3057]; Department of Linguistics, University of Illinois, Urbana, IL 61801. Si vous avez des préoccupations au sujet de cette étude ou

votre expérience en tant que participant, vous pouvez contacter le comité d'éthique de la recherche Institutional Review Board, or IRB at UIUC at 1-217-333-2670 appels à frais virés seront acceptés si vous déclarez vous êtes un participant à l'étude; email: irb@illinois.edu.

Accord: Le but et la nature de cette recherche ont été suffisamment expliqué et je m'engage à participer à cette étude. Je donne la permission d'enregistrer ma voix et d'utiliser les enregistrements dans la préparation et la présentation de ce rapport et d'autres recherches. Je comprends que je suis libre de retirer à tout moment sans encourir aucune sanction.

_____ initiales Je donne la permission pour les extraits de mon discours enregistré pour être joué lors des présentations académiques.

Signature: _____ **Date:** _____

Nom: _____

A.3 Speech Error Study

A.3.1 Materials: Nonce Phrases Presented to Subjects for Speeded Repetition

tis fan bong pum
vum pin dang fos
fung dis ban tom
bas pom vun ding
fum don bang tis
pom fin bung tan
ding tam bon tus
don tum vis pang
dis tan pom vung
dum tis pang von
vung pim das fon
vas pon dum fing
bun pos vam ding
vos pin dung fam
ping fam bon tus
vom pung din fas
fim das bong tun
pus fin bang tom
fun ding bam tos
pas fong bun tim
fun dom bang tis
don tis pum vat
fim dang bos tun
pon fus bing tam
vis pan dom tung
bus pin vam dong
pum fis bang ton
vas pong dun fim
dus tong pam vin
bong pin vum das
ping fam bon tus
pong fus bin tam
bin pas vong dum
bum pin vas dong
fum din bas tong
ban pos vung dim
fum dong ban tis
dos tin pung vam
bing pam vos dun
pon fung bis tam
din tas pom vung
dun ting pas vom
vus pim dan fong
pas fong bun tim
bun pom vang dis

A.3.2 Instructions and Consent Forms

A.3.2.1 Experimenter Instructions Including Verbal Instructions Given to Subjects

Instructions for Speech Error Studies

General procedure of experiment:

- participants reads each line once slowly and 3 times fast

Before the participant arrives:

- make sure the digital recorder is on the correct settings and that you are beginning a new file

When the participant arrives:

- hand them the informed consent form and verbally go over what it says, then allow them to read and sign it, and give them a copy
- ensure that their cell phone is turned off

Sit them at the computer and tell them before they begin:

- tell them that they are going to be repeating nonsense words with the vowels from the words "sid, sad, sod, and sud" in them
- tell them that they are going to read a list of words and they are to read each line once slowly and 3 times quickly
- tell them to first click the left button and read the line to the slow metronome, then click the right button and read the line 3 times to the fast metronome
- when a word begins with "g", please pronounce it as a hard g rather than a soft g not as in gem
- then they do the practice lines

After the experiment is complete:

- pay them and have them sign the receipt
- answer any questions they have about the hypothesis being tested, expected results, etc

A.3.2.2 Subject Consent Form

University of Illinois at Urbana-Champaign

Department of Linguistics

4088 Foreign Languages Building, MC-168
707 South Mathews Avenue
Urbana, IL 61801-3625

Speech production experiment directed by Professor Jennifer Cole Informed Consent –text repetition

Please read this consent agreement carefully. You must be 18 years old or older to participate.

Purpose of the research: We are interested in how quickly and accurately people can repeat words, as part of a larger study on how we hear and produce speech. The purpose of the study is to advance scientific knowledge about human language. Such research may help in the development of therapies for speech disorders.

What you will do in this study: In this experiment you will be asked to read and repeat out loud a word or short lists of words which may include pronunciations that are not common or familiar in English. The words will appear on a computer screen or a sheet of paper. Your speech will be tape-recorded. This project involves no evaluations or comparison of individual people, and your responses are not identified by name in formal or informal reports of our findings.

Risks: There are no anticipated risks, beyond those encountered in daily life, associated with participating in this study.

Compensation: This study will take under 50 minutes to complete. For participating in this study you will receive extra credit in Ling 100 or Ling 225, or 1 PSYCH 100 course credit 1 hour credit, depending on the course you are currently taking and in which we have recruited you as a participant. At the end of the study, you will receive an explanation of the study and the hypotheses. We hope that you will learn a little bit about how linguistic research is conducted.

Voluntary Withdrawal: Your participation in this study is completely voluntary, and you may withdraw from the study at any time without penalty, but course credit will be given only for completion of the experiment. You may withdraw by informing the research associate that you no longer wish to participate no questions will be asked. Your decision to participate, decline or withdraw participation will have no effect on your status at or relationship with the University of Illinois.

Confidentiality: Your participation in this study will remain confidential, and your identity will not be stored with your data. Your responses will be assigned a code number that is not linked to your name or other identifying information. All data and consent forms will be stored in a locked room. Results of this study may be presented at conferences and/or published in books, journals and/or in the popular media.

Further information: If you have questions about this study, please contact Jennifer Cole, Department of Linguistics, University of Illinois, Urbana, IL 61801. Email: jscole@illinois.edu; phone: 217244-3057.

Who to contact about your rights in this study: If you have any concerns about this study or your experience as a participant, you may contact the Institutional Review Board IRB at UIUC at 217333-2670 collect calls will be accepted if you state you are a study participant; email: irb@illinois.edu.

Agreement: The purpose and nature of this research have been sufficiently explained and I agree to participate in this study. I understand that I am free to withdraw at any time without incurring any penalty. I understand that I will receive a copy of this form to take with me.

Signature: _____ **Date:** _____

Name print: _____

Initial here if you agree to have your speech audio recorded: _____

A.3.2.3 Participant Information Form

PARTICIPANT INFORMATION FORM

NAME: _____

AGE: _____

YEAR IN SCHOOL: _____

GENDER: Female Male

Please answer the following questions about yourself:

Do you have any known hearing problem? YES NO

Do you speak any languages other than English? YES NO

If YES please list: _____

Was English the first language you learned? YES NO

__ You are not required to answer these questions. Check here if you do not wish to provide some or all of the information below.

Ethnicity

1. Do you consider yourself to be Hispanic or Latino? Select one.

<input type="checkbox"/>	Hispanic or Latino
<input type="checkbox"/>	Not Hispanic or Latino
<input type="checkbox"/>	Unknown

Race

2. What race do you consider yourself to be? Select one or more of the following.

<input type="checkbox"/>	American Indian or Alaska Native
<input type="checkbox"/>	Asian
<input type="checkbox"/>	Black or African American
<input type="checkbox"/>	Native Hawaiian or Other Pacific Islander
<input type="checkbox"/>	White

For Experimenter to fill in: Experiment: _____

Subject #: _____

Appendix B

NOMSTR Scripts for MATLAB/Freemat

B.1 Core NOMSTR Scripts

B.1.1 Sinusoid Generation Script

```
1     function sinewave = basic_sine(amplitude,frequency,time,phase)
2
3     sinewave = amplitude * sin(2*pi*frequency*((time) + phase));
```

B.1.2 Thresholded Nodes Script

```
1    function prosody_network_output = prosody_network(input_sin_matrix, weight_matrix,
2    threshold)
3    output_1 = 0;
4    output_2 = 0;
5    output_3 = 0;
6    activation_1 = 0;
7    activation_2 = 0;
8    activation_3 = 0;
9
10   q=length(input_sin_matrix);
11   for t=2:q;
12   %neuron 1 activation from inputs
13   activation_1(t) = (input_sin_matrix(1,t-1)*weight_matrix(1,4)) +
14   (output_1(t-1)*weight_matrix(1,1))+ (output_2(t-1)*weight_matrix(1,2)) +
15   (output_3(t-1)*weight_matrix(1,3));
16   %output function for neuron 1
17   if activation_1(t)<threshold, output_1(t)=0;
18   else output_1(t)=input_sin_matrix(1,t);
19   end
20   %neuron 2 activation from inputs
21   activation_2(t) = (input_sin_matrix(2,t-1)*weight_matrix(2,4)) +
22   (output_1(t-1)*weight_matrix(2,1))+ (output_2(t-1)*weight_matrix(2,2)) +
23   (output_3(t-1)*weight_matrix(2,3));
24   %output function for neuron 2
25   if activation_2(t)<threshold, output_2(t)=0;
26   else output_2(t)=input_sin_matrix(2,t);
27   end
28   %neuron 3 activation from inputs
29   activation_3(t) = (input_sin_matrix(3,t-1)*weight_matrix(3,4)) +
30   (output_1(t-1)*weight_matrix(3,1))+ (output_2(t-1)*weight_matrix(3,2)) +
31   (output_3(t-1)*weight_matrix(3,3));
32   %output function for neuron 3
33   if activation_3(t)<threshold, output_3(t)=0;
34   else output_3(t)=input_sin_matrix(3,t);
```

```
34     end
35     end
36
37     prosody_network_output(1,:) = output_1;
38     prosody_network_output(2,:) = output_2;
39     prosody_network_output(3,:) = output_3;
```

B.1.3 Modified Thresholded Nodes Script

(used only in conclusions of Chapter 4)

```
1. function prosody_network_output = prosody_network(input_sin_matrix,
    weight_matrix, threshold)
2. output_1 = 0;
3. output_2 = 0;
4. output_3 = 0;
5. activation_1 = 0;
6. activation_2 = 0;
7. activation_3 = 0;
8. q=length(input_sin_matrix);
9. for t=2:q;
10. %neuron 1 activation from inputs
11. %activation_1(t-1) +
12. activation_1(t) = (input_sin_matrix(1,t-1)*weight_matrix(1,4)) + (output_1(t-
    1)*weight_matrix(1,1)) + (output_2(t-1)*weight_matrix(1,2)) + (output_3(t-
    1)*weight_matrix(1,3));
13. %output function for neuron 1
14. if activation_1(t)<threshold, output_1(t)=0;
15. else output_1(t)=(activation_1(t)/10);
16. end
17. %neuron 2 activation from inputs
18. %activation_2(t-1) +
19. activation_2(t) = (input_sin_matrix(2,t-1)*weight_matrix(2,4)) + (output_1(t-
    1)*weight_matrix(2,1)) + (output_2(t-1)*weight_matrix(2,2)) + (output_3(t-
    1)*weight_matrix(2,3));
20. %output function for neuron 2
21. if activation_2(t)<threshold, output_2(t)=0;
22. else output_2(t)=(activation_2(t)/10);
23. end
24. %neuron 3 activation from inputs
25. %activation_3(t-1) +
26. activation_3(t) = (input_sin_matrix(3,t-1)*weight_matrix(3,4)) + (output_1(t-
    1)*weight_matrix(3,1)) + (output_2(t-1)*weight_matrix(3,2)) + (output_3(t-
    1)*weight_matrix(3,3));
27. %output function for neuron 3
28. if activation_3(t)<threshold, output_3(t)=0;
29. else output_3(t)=(activation_3(t)/10);
30. end
31. end
32. prosody_network_output(1,:) = output_1;
33. prosody_network_output(2,:) = output_2;
34. prosody_network_output(3,:) = output_3;
```

B.2 NOMSTR Extension for Serial Gesture Production Scripts

B.2.1 Tuned Nodes Script

```
1. function phonological_context_vector = phonological_context(input_sin_matrix);
2. %declare height of the tuning curves of the phonological context neurons
3. a=1;
4. %declare width (from center) of the tuning curves
5. c=0.2;
6. %declare the center of each of the tuning curves
7. %7 tuning curves are required for the 3 sets of 7 neurons
8. b1=-1.5;
9. b2=-1;
10. b3=-.5;
11. b4=0;
12. b5=.5;
13. b6=1;
14. b7=1.5;
15. for t=1:length(input_sin_matrix);
16. %equations made up of Gaussian tuning curves for each neuron
17. %neurons attached to the first oscillator
18. phonological_context_vector(1,t)=(a*exp(1))-((input_sin_matrix(1,t)-b1)^2)/((2*c)^2);
19. phonological_context_vector(2,t)=(a*exp(1))-((input_sin_matrix(1,t)-b2)^2)/((2*c)^2);
20. phonological_context_vector(3,t)=(a*exp(1))-((input_sin_matrix(1,t)-b3)^2)/((2*c)^2);
21. phonological_context_vector(4,t)=(a*exp(1))-((input_sin_matrix(1,t)-b4)^2)/((2*c)^2);
22. phonological_context_vector(5,t)=(a*exp(1))-((input_sin_matrix(1,t)-b5)^2)/((2*c)^2);
23. phonological_context_vector(6,t)=(a*exp(1))-((input_sin_matrix(1,t)-b6)^2)/((2*c)^2);
24. phonological_context_vector(7,t)=(a*exp(1))-((input_sin_matrix(1,t)-b7)^2)/((2*c)^2);
25. %neurons attached to the second oscillator
26. phonological_context_vector(8,t)=(a*exp(1))-((input_sin_matrix(2,t)-b1)^2)/((2*c)^2);
27. phonological_context_vector(9,t)=(a*exp(1))-((input_sin_matrix(2,t)-b2)^2)/((2*c)^2);
28. phonological_context_vector(10,t)=(a*exp(1))-((input_sin_matrix(2,t)-b3)^2)/((2*c)^2);
29. phonological_context_vector(11,t)=(a*exp(1))-((input_sin_matrix(2,t)-b4)^2)/((2*c)^2);
30. phonological_context_vector(12,t)=(a*exp(1))-((input_sin_matrix(2,t)-b5)^2)/((2*c)^2);
31. phonological_context_vector(13,t)=(a*exp(1))-((input_sin_matrix(2,t)-b6)^2)/((2*c)^2);
32. phonological_context_vector(14,t)=(a*exp(1))-((input_sin_matrix(2,t)-b7)^2)/((2*c)^2);
33. %neurons attached to the third oscillator
34. phonological_context_vector(15,t)=(a*exp(1))-((input_sin_matrix(3,t)-b1)^2)/((2*c)^2);
35. phonological_context_vector(16,t)=(a*exp(1))-((input_sin_matrix(3,t)-b2)^2)/((2*c)^2);
36. phonological_context_vector(17,t)=(a*exp(1))-((input_sin_matrix(3,t)-b3)^2)/((2*c)^2);
37. phonological_context_vector(18,t)=(a*exp(1))-((input_sin_matrix(3,t)-b4)^2)/((2*c)^2);
38. phonological_context_vector(19,t)=(a*exp(1))-((input_sin_matrix(3,t)-b5)^2)/((2*c)^2);
39. phonological_context_vector(20,t)=(a*exp(1))-((input_sin_matrix(3,t)-b6)^2)/((2*c)^2);
40. phonological_context_vector(21,t)=(a*exp(1))-((input_sin_matrix(3,t)-b7)^2)/((2*c)^2);
41. %neurons with negative activation do not output (output=0)
```

```
42. for x = 1:21
43. if phonological_context_vector(x,t)>0, phonological_context_vector(x,t) =
    phonological_context_vector(x,t);
44. else phonological_context_vector(x,t)=0;
45. end
46. end
47. end
```

B.3 Settings of NOMSTR Variables for Utterance Simulations

B.3.1 English Utterances

Note: wavelengths in seconds

Utterance 1

estimated stress oscillator wavelength = .1

estimated syllable oscillator wavelength = .28

estimated phrase oscillator wavelength = 2

threshold: 5

weights: 0 -17 -15 20

 0 0 2 20

 0 0 0 20

Utterance 2

estimated stress oscillator wavelength = .4

estimated syllable oscillator wavelength = .11

estimated phrase oscillator wavelength = 2.08

threshold: 8

weights: 0 -17 -15 20

 0 0 2 20

 0 0 0 20

Utterance 3

estimated stress oscillator wavelength = .24 (phase=.66pi)

estimated syllable oscillator wavelength = .11

estimated phrase oscillator wavelength = .79

threshold: 5

weights: 0 -17 -15 20

 0 0 2 20

 0 0 0 20

B.3.2 French Utterances

Utterance 1

estimated stress oscillator wavelength = .61

estimated syllable oscillator wavelength = .10

estimated phrase oscillator wavelength = 3.33

threshold: 5

weights: 0 -23-14.1 20
 0 0 0 20
 0 0 0 20

Utterance 2

estimated syllable oscillator wavelength = .09

estimated stress oscillator wavelength = .59 (phase= $\pi/3$)

estimated phrase oscillator wavelength = 2.17 (phase= $4\pi/3$)

threshold: 5

weights: 0 -20-14.1 20
 0 0 0 20
 0 0 0 20

Utterance 3

estimated syllable oscillator wavelength = .1

estimated stress oscillator wavelength = .64 (phase= 2π)

estimated phrase oscillator wavelength = 4 (phase= $\pi/2$) 1.1

threshold: 5

weights: 0 -20-14.1 20
 0 0 -15 20
 0 0 0 20

B.4 Scripts for NOMSTR extension used in Chapter 6

B.4.1 Script for Layer of Tuned Neurons for Phonological Context Vector Creation

```
1. function phonological_context_vector = phonological_context(input_sin_matrix);
2.
3. %declare height of the tuning curves of the phonological context neurons
4. a=1;
5.
6. %declare width (from center) of the tuning curves
7. c=0.2;
8.
9. %declare the center of each of the tuning curves
10. %7 tuning curves are required for the 3 sets of 7 neurons
11. b1=-1.5;
12. b2=-1;
13. b3=-.5;
14. b4=0;
15. b5=.5;
16. b6=1;
17. b7=1.5;
18.
19.
20. for t=1:length(input_sin_matrix);
21.
22. %equations made up of Gaussian tuning curves for each neuron
23. %neurons attached to the first oscillator
24. phonological_context_vector(1,t)=(a*exp(1))-((input_sin_matrix(1,t)-b1)^2)/
    ((2*c)^2);
25. phonological_context_vector(2,t)=(a*exp(1))-((input_sin_matrix(1,t)-b2)^2)/
    ((2*c)^2);
26. phonological_context_vector(3,t)=(a*exp(1))-((input_sin_matrix(1,t)-b3)^2)/
    ((2*c)^2);
27. phonological_context_vector(4,t)=(a*exp(1))-((input_sin_matrix(1,t)-b4)^2)/
    ((2*c)^2);
28. phonological_context_vector(5,t)=(a*exp(1))-((input_sin_matrix(1,t)-b5)^2)/
    ((2*c)^2);
29. phonological_context_vector(6,t)=(a*exp(1))-((input_sin_matrix(1,t)-b6)^2)/
    ((2*c)^2);
30. phonological_context_vector(7,t)=(a*exp(1))-((input_sin_matrix(1,t)-b7)^2)/
    ((2*c)^2);
31.
32. %neurons attached to the second oscillator
33. phonological_context_vector(8,t)=(a*exp(1))-((input_sin_matrix(2,t)-b1)^2)/
```

```

((2*c)^2);
34. phonological_context_vector(9,t)=(a*exp(1))-((input_sin_matrix(2,t)-b2)^2)/
((2*c)^2);
35. phonological_context_vector(10,t)=(a*exp(1))-((input_sin_matrix(2,t)-b3)^2)/
((2*c)^2);
36. phonological_context_vector(11,t)=(a*exp(1))-((input_sin_matrix(2,t)-b4)^2)/
((2*c)^2);
37. phonological_context_vector(12,t)=(a*exp(1))-((input_sin_matrix(2,t)-b5)^2)/
((2*c)^2);
38. phonological_context_vector(13,t)=(a*exp(1))-((input_sin_matrix(2,t)-b6)^2)/
((2*c)^2);
39. phonological_context_vector(14,t)=(a*exp(1))-((input_sin_matrix(2,t)-b7)^2)/
((2*c)^2);
40.
41. %neurons attached to the third oscillator
42. phonological_context_vector(15,t)=(a*exp(1))-((input_sin_matrix(3,t)-b1)^2)/
((2*c)^2);
43. phonological_context_vector(16,t)=(a*exp(1))-((input_sin_matrix(3,t)-b2)^2)/
((2*c)^2);
44. phonological_context_vector(17,t)=(a*exp(1))-((input_sin_matrix(3,t)-b3)^2)/
((2*c)^2);
45. phonological_context_vector(18,t)=(a*exp(1))-((input_sin_matrix(3,t)-b4)^2)/
((2*c)^2);
46. phonological_context_vector(19,t)=(a*exp(1))-((input_sin_matrix(3,t)-b5)^2)/
((2*c)^2);
47. phonological_context_vector(20,t)=(a*exp(1))-((input_sin_matrix(3,t)-b6)^2)/
((2*c)^2);
48. phonological_context_vector(21,t)=(a*exp(1))-((input_sin_matrix(3,t)-b7)^2)/
((2*c)^2);
49.
50. %neurons with negative activation do not output (output=0)
51. for x = 1:21
52. if phonological_context_vector(x,t)>0, phonological_context_vector(x,t) =
phonological_context_vector(x,t);
53. else phonological_context_vector(x,t)=0;
54. end
55. end
56. end
57.

```

B.4.2 Scripts to Create the Associations/Weights Between Context and Output Nodes

weight matrix loop

1. a=size(phonological_context_vectot,2);
2. long_weight_matrices={};
3. for t=1:a;
4. long_weight_matrices{1,t}=create_weight_matrix(phonological_context_vectot(:,t),gesture_vector(:,t));
5. end

function weight_matrix=create_weight_matrix(phonological_context_vector,gesture_vector);

1. a=size(phonological_context_vector,2);
2. for t=1:a;
3. weight_matrix=phonological_context_vector*gesture_vector';
4. end
5. end
- 6.

B.4.3 Scripts for Output Layers

B.4.3.1 Nucleus Nodes Activation Script

```
function c_center_timed_vowels = c_center_timing(sequential_vowels_output,  
prosody_network_context_output);  
1. for t = 1:length(sequential_vowels_output);  
2. %activation of each of the vowel nodes at time t is equal to the activation level  
3. %from the sequential vowels output at t plus the (output of the syllable  
4. %node of the prosody network context output at t times the weight of  
5. %the connection between the syllable node and the syllable nucleus  
6. %array)  
7. c_center_timed_vowels(1,t)=sequential_vowels_output(1,t)+  
   (prosody_network_context_output(1,t)*10)+randn*8;  
8. c_center_timed_vowels(2,t)=sequential_vowels_output(2,t)+  
   (prosody_network_context_output(1,t)*10)+randn*8;  
9. c_center_timed_vowels(3,t)=sequential_vowels_output(3,t)+  
   (prosody_network_context_output(1,t)*10)+randn*8;  
10. c_center_timed_vowels(4,t)=sequential_vowels_output(4,t)+  
   (prosody_network_context_output(1,t)*10)+randn*8;  
11. end
```

B.4.3.2 Onset/Coda Nodes Activation Script

```
1. function vowel_excited_consonants =  
   syllables(sequential_consonants_output,sequential_vowels_output);  
2. for t=1:12  
3. %tis  
4. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+  
   (sequential_vowels_output(4,t)/10)+randn*8;  
5. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+  
   (sequential_vowels_output(4,t)/10)+randn*8;  
6. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+  
   (sequential_vowels_output(4,t)/10)+randn*8;  
7. %fan  
8. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+  
   (sequential_vowels_output(1,t)/10)+randn*8;  
9. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+  
   (sequential_vowels_output(1,t)/10)+randn*8;  
10. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+  
   (sequential_vowels_output(1,t)/10)+randn*8;  
11. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+  
   (sequential_vowels_output(1,t)/10)+randn*8;  
12. %bong  
13. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+  
   (sequential_vowels_output(2,t)/10)+randn*8;  
14. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+  
   (sequential_vowels_output(2,t)/10)+randn*8;  
15. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+  
   (sequential_vowels_output(2,t)/10)+randn*8;  
16. %pum  
17. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+  
   (sequential_vowels_output(3,t)/10)+randn*8;  
18. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+  
   (sequential_vowels_output(3,t)/10)+randn*8;  
19. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+  
   (sequential_vowels_output(3,t)/10)+randn*8;  
20. end  
21.  
22. for t=13:224  
23. %vum  
24. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+  
   (sequential_vowels_output(3,t)/10)+randn*8;  
25. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+  
   (sequential_vowels_output(3,t)/10)+randn*8;  
26. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
```

```

    (sequential_vowels_output(3,t)/10)+randn*8;
27. %pin
28. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
29. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
30. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
31. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
32. %dang
33. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
34. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
35. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
36. %fos
37. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
38. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
39. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
40. end
41.
42. for t=25:36
43. %fung
44. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
45. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
46. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
47. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
48. %dis
49. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
50. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
51. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
52. %ban

```

```

53. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
54. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
55. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
56. %tom
57. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
58. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
59. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
60. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
61. end
62.
63. for t=37:48
64. %bas
65. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
66. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
67. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
68. %pom
69. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
70. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
71. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
72. %vun
73. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
74. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
75. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
76. %ding
77. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
78. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;

```

```

79. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
80. end
81.
82. for t=49:60
83. %fum
84. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
85. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
86. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
87. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
88. %don
89. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
90. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
91. %bang
92. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
93. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
94. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
95. %tis
96. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
97. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
98. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
99. end
100. for t=61:72
101. %pom
102. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
103. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
104. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
105. %fin
106. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+

```

```

    (sequential_vowels_output(4,t)/10)+randn*8;
107. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
108. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
109. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
110. %bung
111. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
112. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
113. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
114. %tan
115. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
116. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
117. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
118. end
119.
120. for t=73:84
121. %ding
122. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
123. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
124. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
125. %tam
126. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
127. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
128. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
129. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
130. %bon
131. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
132. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+

```

```

        (sequential_vowels_output(2,t)/10)+randn*8;
133. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
134. %tus
135. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
136. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
137. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
138. end
139.
140. for t=85:96
141. %don
142. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
143. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
144. %tum
145. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
146. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
147. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
148. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
149. %vis
150. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
151. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
152. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
153. %pang
154. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
155. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
156. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
157. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
158. end

```

```

159.
160. for t=97:108
161. %dis
162. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
163. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
164. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
165. %tan
166. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
167. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
168. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
169. %pom
170. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
171. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
172. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
173. %vung
174. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
175. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
176. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
177. end
178.
179. for t=109:120
180. %dum
181. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
182. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
183. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
184. %tis
185. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
186. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+

```

```

    (sequential_vowels_output(4,t)/10)+randn*8;
187. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
188. %pang
189. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
190. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
191. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
192. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
193. %von
194. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
195. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
196. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
197. end
198.
199. for t=121:132
200. %vung
201. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
202. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
203. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
204. %pim
205. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
206. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
207. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
208. %das
209. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
210. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
211. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
212. %fon

```

```

213. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
214. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
215. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
216. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
217. end
218.
219. for t=133:144
220.     %vas
221.     vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
        (sequential_vowels_output(1,t)/10)+randn*8;
222.     vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
        (sequential_vowels_output(1,t)/10)+randn*8;
223.     vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
        (sequential_vowels_output(1,t)/10)+randn*8;
224.     %pon
225.     vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
226.     vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
227.     vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
228.     vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
229.     %dum
230.     vowel_excited_consonants(2,t)=sequential_consonants_output(3,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
231.     vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
        (sequential_vowels_output(3,t)/10)+randn*8;
232.     vowel_excited_consonants(6,t)=sequential_consonants_output(3,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
233.     %fing
234.     vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
235.     vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
236.     vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
237.     vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
238. end

```

```

239.
240.   for t=145:156
241.     %bun
242.     vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
        (sequential_vowels_output(3,t)/10)+randn*8;
243.     vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
        (sequential_vowels_output(3,t)/10)+randn*8;
244.     vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
        (sequential_vowels_output(3,t)/10)+randn*8;
245.     %pos
246.     vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
247.     vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
248.     vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
249.     %vam
250.     vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
        (sequential_vowels_output(1,t)/10)+randn*8;
251.     vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
        (sequential_vowels_output(1,t)/10)+randn*8;
252.     vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
        (sequential_vowels_output(1,t)/10)+randn*8;
253.     %ding
254.     vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
255.     vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
256.     vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
257.   end
258.
259.   for t=157:168
260.     %vos
261.     vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
262.     vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
263.     vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
264.     %pin
265.     vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
266.     vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+

```

```

    (sequential_vowels_output(4,t)/10)+randn*8;
267. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
268. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
269. %dung
270. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
271. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
272. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
273. %fam
274. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
275. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
276. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
277. end
278.
279. for t=169:180
280. %ping
281. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
282. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
283. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
284. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
285. %fam
286. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
287. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
288. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
289. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
290. %bon
291. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
292. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+

```

```

    (sequential_vowels_output(2,t)/10)+randn*8;
293. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
294. %tus
295. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
296. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
297. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
298. end
299.
300. for t=181:192
301. %vom
302. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
303. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
304. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
305. %pung
306. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
307. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
308. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
309. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
310. %din
311. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
312. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
313. %fas
314. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
315. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
316. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
317. end
318.
319. for t=193:204

```

```

320. %fim
321. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
322. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
323. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
324. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
325. %das
326. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
327. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
328. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
329. %bong
330. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t))+randn*8;
331. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(2,t))+randn*8;
332. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t))+randn*8;
333. %tun
334. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
335. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
336. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
337. end
338.
339. for t=205:216
340. %pus
341. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
342. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
343. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
344. %fin
345. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
346. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+

```

```

    (sequential_vowels_output(4,t)/10)+randn*8;
347. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
348. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
349. %bang
350. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
351. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
352. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
353. %tom
354. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
355. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
356. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
357. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
358. end
359.
360. for t=217:228
361. %fun
362. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
363. vowel_excited_consonants(3,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
364. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
365. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
366. %ding
367. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
368. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
369. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
370. %bam
371. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
372. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+

```

```

        (sequential_vowels_output(1,t)/10)+randn*8;
373. %tos
374. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
375. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
376. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
377. end
378.
379. for t=229:240
380. %pas
381. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
382. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
383. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
384. %fong
385. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
386. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
387. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
388. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
389. %bun
390. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
391. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
392. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
393. %tim
394. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
395. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
396. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
397. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
398. end

```

```

399.
400.   for t=241:252
401.     %fun
402.     vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
        (sequential_vowels_output(3,t)/10)+randn*8;
403.     vowel_excited_consonants(3,t)=sequential_consonants_output(5,t)+
        (sequential_vowels_output(3,t)/10)+randn*8;
404.     vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
        (sequential_vowels_output(3,t)/10)+randn*8;
405.     vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
        (sequential_vowels_output(3,t)/10)+randn*8;
406.     %dom
407.     vowel_excited_consonants(2,t)=sequential_consonants_output(3,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
408.     vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
409.     vowel_excited_consonants(6,t)=sequential_consonants_output(3,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
410.     %bang
411.     vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
        (sequential_vowels_output(1,t)/10)+randn*8;
412.     vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
        (sequential_vowels_output(1,t)/10)+randn*8;
413.     vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
        (sequential_vowels_output(1,t)/10)+randn*8;
414.     %tis
415.     vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
416.     vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
417.     vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
418.   end
419.
420.   for t=253:264
421.     %don
422.     vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
423.     vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
424.     %tis
425.     vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
426.     vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+

```

```

    (sequential_vowels_output(4,t)/10)+randn*8;
427. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
428. %pum
429. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
430. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
431. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
432. %vat
433. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
434. vowel_excited_consonants(3,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
435. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
436. end
437.
438. for t=265:276
439. %fim
440. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
441. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
442. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
443. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
444. %dang
445. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
446. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
447. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
448. %bos
449. vowel_excited_consonants(2,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
450. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
451. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
452. %tun

```

```

453. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
454. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
455. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
456. end
457.
458. for t=277:288
459.     %pon
460.     vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
461.     vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
462.     vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
463.     vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
464.     %fus
465.     vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
        (sequential_vowels_output(3,t)/10)+randn*8;
466.     vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
        (sequential_vowels_output(3,t)/10)+randn*8;
467.     vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
        (sequential_vowels_output(3,t)/10)+randn*8;
468.     %bing
469.     vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
470.     vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
471.     vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
472.     %tam
473.     vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
        (sequential_vowels_output(1,t)/10)+randn*8;
474.     vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
        (sequential_vowels_output(1,t)/10)+randn*8;
475.     vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
        (sequential_vowels_output(1,t)/10)+randn*8;
476.     vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
        (sequential_vowels_output(1,t)/10)+randn*8;
477. end
478.
479. for t=289:300

```

```

480. %vis
481. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
482. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
483. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
484. %pan
485. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
486. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
487. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
488. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
489. %dom
490. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
491. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
492. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
493. %tung
494. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
495. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
496. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
497. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
498. end
499.
500. for t=301:312
501. %bus
502. vowel_excited_consonants(2,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
503. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
504. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
505. %pin
506. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+

```

```

    (sequential_vowels_output(4,t)/10)+randn*8;
507. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
508. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
509. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
510. %vam
511. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
512. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
513. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
514. %dong
515. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
516. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
517. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
518. end
519.
520. for t=313:324
521. %pum
522. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
523. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
524. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
525. %fis
526. vowel_excited_consonants(1,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
527. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
528. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
529. %bang
530. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
531. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
532. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+

```

```

        (sequential_vowels_output(1,t)/10)+randn*8;
533. %ton
534. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
535. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
536. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
537. end
538.
539. for t=325:336
540. %vas
541. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
542. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
543. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
544. %pong
545. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
546. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
547. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
548. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
549. %dun
550. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
551. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
552. %fim
553. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
554. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
555. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
556. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
557. end
558.
559. for t=337:348

```

```

560. %dus
561. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
562. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
563. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
564. %tong
565. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
566. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
567. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
568. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
569. %pam
570. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
571. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
572. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
573. %vin
574. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
575. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
576. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
577. end
578.
579. for t=349:360
580. %bong
581. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
582. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
583. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
584. %pin
585. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
586. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+

```

```

        (sequential_vowels_output(4,t)/10)+randn*8;
587. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
588. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
589. %vum
590. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
591. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
592. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
593. %das
594. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
595. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
596. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
597. end
598.
599. for t=361:372
600. %ping
601. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
602. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
603. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
604. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
605. %fam
606. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
607. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
608. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
609. %bon
610. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
611. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
612. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+

```

```

        (sequential_vowels_output(2,t)/10)+randn*8;
613. %tus
614. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
615. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
616. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
617. end
618.
619. for t=373:384
620. %pong
621. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
622. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
623. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
624. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
625. %fus
626. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
627. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
628. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
629. %bin
630. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
631. vowel_excited_consonants(3,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
632. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
633. %tam
634. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
635. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
636. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
637. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
638. end

```

```

639.
640.   for t=385:396
641.     %bin
642.     vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
643.     vowel_excited_consonants(3,t)=sequential_consonants_output(5,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
644.     vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
645.     %pas
646.     vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
        (sequential_vowels_output(1,t)/10)+randn*8;
647.     vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
        (sequential_vowels_output(1,t)/10)+randn*8;
648.     vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
        (sequential_vowels_output(1,t)/10)+randn*8;
649.     %vong
650.     vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
651.     vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
652.     vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
        (sequential_vowels_output(2,t)/10)+randn*8;
653.     %dum
654.     vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
        (sequential_vowels_output(3,t)/10)+randn*8;
655.     vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
        (sequential_vowels_output(3,t)/10)+randn*8;
656.     vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
        (sequential_vowels_output(3,t)/10)+randn*8;
657.   end
658.
659.   for t=397:408
660.     %bum
661.     vowel_excited_consonants(2,t)=sequential_consonants_output(1,t)+
        (sequential_vowels_output(3,t)/10)+randn*8;
662.     vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
        (sequential_vowels_output(3,t)/10)+randn*8;
663.     %pin
664.     vowel_excited_consonants(2,t)=sequential_consonants_output(1,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
665.     vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
        (sequential_vowels_output(4,t)/10)+randn*8;
666.     vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+

```

```

        (sequential_vowels_output(4,t)/10)+randn*8;
667. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
668. %vas
669. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
670. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
671. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
672. %dong
673. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
674. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
675. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
676. end
677.
678. for t=409:420
679. %fum
680. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
681. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
682. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
683. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
684. %din
685. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
686. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
687. %bas
688. vowel_excited_consonants(2,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
689. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
690. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
691. %tong
692. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;

```

```

693. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
694. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
695. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
696. end
697.
698. for t=421:432
699. %ban
700. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
701. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
702. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
703. %pos
704. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
705. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
706. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
707. %vung
708. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
709. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
710. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
711. %dim
712. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
713. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
714. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
715. end
716.
717. for t=433:444
718. %fum
719. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
720. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+

```

```

    (sequential_vowels_output(3,t)/10)+randn*8;
721. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
722. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
723. %dong
724. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
725. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
726. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
727. %ban
728. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
729. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
730. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
731. %tis
732. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
733. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
734. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
735. end
736.
737. for t=445:456
738. %dos
739. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
740. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
741. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
742. %tin
743. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
744. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
745. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
746. %pung

```

```

747. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
748. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
749. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
750. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
751. %vam
752. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
753. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
754. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
755. end
756.
757. for t=457:468
758. %bing
759. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
760. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
761. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
762. %pam
763. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
764. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
765. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
766. %vos
767. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
768. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
769. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
770. %dun
771. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
772. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;

```

```

773. end
774.
775. for t=469:480
776. %pon
777. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
778. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
779. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
780. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
781. %fung
782. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
783. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
784. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
785. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
786. %bis
787. vowel_excited_consonants(2,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
788. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
789. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
790. %tam
791. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
792. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
793. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
794. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
795. end
796.
797. for t=481:492
798. %din
799. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
800. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+

```

```

      (sequential_vowels_output(4,t)/10)+randn*8;
801. %tas
802. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
      (sequential_vowels_output(1,t)/10)+randn*8;
803. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
      (sequential_vowels_output(1,t)/10)+randn*8;
804. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
      (sequential_vowels_output(1,t)/10)+randn*8;
805. %pom
806. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
      (sequential_vowels_output(2,t)/10)+randn*8;
807. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
      (sequential_vowels_output(2,t)/10)+randn*8;
808. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
      (sequential_vowels_output(2,t)/10)+randn*8;
809. %vung
810. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
      (sequential_vowels_output(3,t)/10)+randn*8;
811. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
      (sequential_vowels_output(3,t)/10)+randn*8;
812. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
      (sequential_vowels_output(3,t)/10)+randn*8;
813. end
814.
815. for t=493:504
816. %dun
817. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
      (sequential_vowels_output(3,t)/10)+randn*8;
818. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
      (sequential_vowels_output(3,t)/10)+randn*8;
819. %ting
820. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
      (sequential_vowels_output(4,t)/10)+randn*8;
821. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
      (sequential_vowels_output(4,t)/10)+randn*8;
822. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
      (sequential_vowels_output(4,t)/10)+randn*8;
823. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
      (sequential_vowels_output(4,t)/10)+randn*8;
824. %pas
825. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
      (sequential_vowels_output(1,t)/10)+randn*8;
826. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
      (sequential_vowels_output(1,t)/10)+randn*8;

```

```

827. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
828. %vom
829. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
830. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
831. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
832. end
833.
834. for t=505:516
835. %vus
836. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
837. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
838. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
839. %pim
840. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
841. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
842. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
843. %dan
844. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
845. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
846. %fong
847. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
848. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
849. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
850. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
851. end
852.
853. for t=517:528
854. %pas

```

```

855. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
856. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
857. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
858. %fong
859. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
860. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
861. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
862. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
863. %bun
864. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
865. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
866. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
867. %tim
868. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
869. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
870. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
871. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
872. end
873.
874. for t=529:540
875. %bun
876. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
877. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
878. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(3,t)/10)+randn*8;
879. %pom
880. vowel_excited_consonants(2,t)=sequential_consonants_output(2,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;

```

```
881. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
882. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(2,t)/10)+randn*8;
883. %vang
884. vowel_excited_consonants(1,t)=sequential_consonants_output(1,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
885. vowel_excited_consonants(5,t)=sequential_consonants_output(5,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
886. vowel_excited_consonants(6,t)=sequential_consonants_output(6,t)+
    (sequential_vowels_output(1,t)/10)+randn*8;
887. %dis
888. vowel_excited_consonants(3,t)=sequential_consonants_output(3,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
889. vowel_excited_consonants(4,t)=sequential_consonants_output(4,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
890. vowel_excited_consonants(7,t)=sequential_consonants_output(7,t)+
    (sequential_vowels_output(4,t)/10)+randn*8;
891. end
892. end
```

References

- Anastasio, T.J. 2010. *Tutorial on Neural Systems Modeling*. Sinauer Associates. Sunderland, MA.
- Astésano, C., E.G. Bard & A. Turk. 2007. Structural influences on initial accent placement in French. *Language and Speech*. 50;3. 423-446.
- Barbosa, P.A. 2001. Generating duration from a cognitively plausible model of rhythm production. Seventh European Conference on Speech Communication and Technology. Aalborg, Denmark.
- Barbosa, P.A. 2002. Explaining Brazilian Portuguese resistance to stress shift with a coupled-oscillator model of speech rhythm production. *Cadernos de Estudos Linguísticos*. 43. 71-92.
- Barlow, S.M & M. Estep. 2006. Central pattern generation and the motor infrastructure for suck, respiration, and speech. *Journal of Communication Disorders*. 39;5. 366-380.
- Basu, S. 2011. FreeMat [Computer program] Version 4.1, retrieved February 29, 2012 from <http://freemat.sourceforge.net/>
- Boersma, P. & D. Weenink. 2013. Praat: doing phonetics by computer [Computer program]. Version 5.1.25, retrieved February 02, 2010 from <http://www.praat.org/>
- Bolinger, D. L. 1958. A theory of pitch accent in English. *Word*. 14. 109-149.
- Bradlow, AR, L. Ackerman, L.A. Burchfield, L. Hesterberg, J.S. Luque, K. Mok. 2010. ALLSTAR: Archive of L1 and L2 Scripted and Spontaneous Transcripts And Recordings. http://groups.linguistics.northwestern.edu/speech_comm_group/allstar/index.html. Department of Linguistics, Northwestern University.
- Browman, C.P. & L.M. Goldstein. 1986. Towards an articulatory phonology. *Phonology*. 3. 219-252.
- Browman, C.P. & L.M. Goldstein. 1988. Some notes on syllable structure in articulatory phonology. *Phonetica*. 45. 140-155.
- Browman, C.P. & L.M. Goldstein. 1989. Articulatory gestures as phonological units. *Phonology*. 6. 201-251.

- Browman, C.P. & L.M. Goldstein. 1992. Articulatory phonology, An overview. *Phonetica*. 49. 155-180.
- Browman, C.P. & L.M. Goldstein. 1995. Dynamics and Articulatory Phonology. *Mind as Motion*. Port, R.F. & van Gelder, T. eds. The MIT Press. Cambridge.
- Buckley, E. 2009. Locality in metrical typology. *Phonology*. 26. 389-435.
- Byrd, D. 1996. A phase window framework for articulatory timing. *Phonology*, 13, 139-170.
- Calhoun, S. 2010a. The centrality of metrical structure in signaling information structure: A probabilistic perspective. *Language* 86: 1-42.
- Cole, J. & S. Shattuck-Hufnagel. 2011. The phonology and phonetics of perceived prosody: What do listeners imitate? *Proceedings of Interspeech 2011*. Florence, Italy.
- Cooper, W.E. & S.J. Eady. 1986. Metrical phonology in speech production. *Journal of Memory and Language*. 25; 3. 369-384.
- Crystal, T.H. & A.S. House. 1988. Segmental durations in connected-speech signals: Syllabic stress. *Journal of the Acoustical Society of America*. 83;4. 1574-1585.
- Cummins, F. 2005. Interval timing in spoken lists of words. *Music Perception*. 22;3. 497-508.
- Cummins, F. & R. Port. 1998. Rhythmic constraints on stress timing in English. *Journal of Phonetics*. 26. 145-171.
- Dauer, R. 1983. Stress-timing and syllable-timing reanalyzed. *Journal of Phonetics*. 11. 51-62.
- Delcomyn F. 1980 Neural basis of rhythmic behavior in animals. *Science*. 210. 492-498.
- Dell, G. 1988. The retrieval of phonological forms in production: Tests of predictions from a connectionist model. *Journal of Memory and Language*. 27;2. 124-142.
- Dell, G., L.K. Burger & W.R. Svec. 1997. Language production and serial order: A functional analysis and a model. *Psychological Review*. 104;1. 123-147.
- Dell, G.; F. Chang & Z.M. Griffin. 1999. Connectionist models of language production: lexical access and grammatical encoding. *Cognitive Science*. 23;4. 517-542.

- Deng, G.R. & D. Sun. 1997. Production models as a structural basis for automatic speech recognition. *Speech Communication*. 33. 93–111.
- Di Cristo, A.. 1998. Intonation in French. In: Hirst, D. J. & Di Cristo, A. eds. *Intonational Systems*, 195-218. Cambridge: CUP.
- Di Cristo, A., P. Di Cristo & J. Véronis. 1997. A metrical model of rhythm and intonation for French text-to-speech synthesis. In: Botinis, A. ed.. *Intonation: Theory, Models and Applications ESCA*. 83–86.
- Edwards, J, & M.E. Beckman. 1988. Articulatory timing and the prosodic interpretation of syllable duration, *Phonetica* 45. 156-174.
- Edwards, J., M.E. Beckman & J. Fletcher. 1991. The articulatory kinematics of final lengthening. *JASA*. 89:1, 369-382.
- Euler, C. von. 1983. On the central pattern generator for the basic breathing rhythmicity. *Journal of Applied Physiology* 55; 6. 1647-1659.
- Grabe, E. & E.L. Low. 2002. Durational variability in speech and the rhythm class hypothesis. *Laboratory Phonology 7*. Gussenhoven, C. & Warner, N. eds. Mouton de Gruyter. Berlin.
- Gracco, V.L. & J.H. Abbs. 1988. Central patterning of speech movements. *Experimental Brain Research*. 71; 3. 515-526.
- Grillner, S. 1985. Neurobiological bases of rhythmic motor acts in vertebrates. *Science*. 228. 143-149.
- Goldman, J.-Ph. 2011. *EasyAlign: an automatic phonetic alignment tool under Praat*. InterSpeech 2011. September, Firenze, Italy
- Goldstein, L., H. Nam, E. Saltzman & I. Chitoran. 2008. Coupled oscillator planning model of speech planning and syllable structure. Proceedings of the 8th phonetics conference of China and the international symposium on phonetic frontiers. Beijing.
- Goldstein, L., M. Pouplier, L. Chen, E. Saltzman & D. Byrd. 2007. Dynamic action units slip in speech production errors. *Cognition*. 103;3. 386-412.
- Harris, H.D. 2002. Holographic Reduced Representations for Oscillator Recall: A Model of Phonological Production. In W. D. Gray & C. D. Schunn Eds., Proceedings of the 24th annual meeting of the Cognitive Science Society. Hillsdale, NJ: Lawrence Erlbaum Associates Inc.

- Hayes, B. 1982. Extrametricality and English stress. *Linguistic Inquiry*. 13;2. 227-276.
- Hayes, B. 1995. *Metrical Stress Theory: Principles and Case Studies*. The University of Chicago Press, Chicago.
- Hirst, D. 1998. Intonation in British English. In: Hirst, D. & Di Cristo, A. eds.. *Intonation Systems: A Survey of Twenty Languages*. 56-77. Cambridge University Press, Cambridge.
- Ijspeert, A.J. 2001. A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander. *Biological Cybernetics*. 84;5. 331-348.
- Livescu, K. & J. Glass. 2004. Feature-based pronunciation modeling with trainable asynchrony probabilities. ICSLP.
- Lund, J.P. & A. Kolta. 2006. Brainstem circuits that control mastication: Do they have anything to say during speech?. *Journal of Communication Disorders*. 39; 5. 381-390.
- MATLAB version 7.10.0. Natick, Massachusetts: The MathWorks Inc., 2010.
- Morton, J., S. Marcus & C. Frankish. 1976. Perceptual centers P-centers. *Psychological Review*. 83;5. 405-408.
- Nam, H. & E. Saltzman. 2003. A competitive, coupled oscillator model for syllable structure. *Proceedings of the 15th International Congress of Phonetic Science*. Barcelona.
- Nespor, M. & I. Vogel. 2007. *Prosodic Phonology: With a new forward*. Mouton de Gruyter. Berlin.
- Noske, R. 1982. Syllabification and syllable changing rules in French. in: van der Hulst, H. & Smith, N. eds.. *The Structure of Phonological Representations*. Foris, Dordrecht.
- O'Dell, M., M. Lennes & T. Nieminen. 2008. Hierarchical levels of rhythm in conversational speech. *Proceedings of Speech Prosody 2008*. Campinas, Brazil.
- O'Dell, M., & T. Nieminen. 1999. Coupled oscillator model of speech rhythm. *Proceedings of the XIVth International Congress of the Phonetic Sciences*. New York, NY.
- Pike, K. L. 1945. *The Intonation of American English*, Ann Arbor: University of Michigan Press.

- Port, R.F. 2003. Meter and speech. *Journal of Phonetics*. 31; 3-4. 599-611.
- Port, R.F. & K. Tajima. 1999. Speech and rhythmic behavior. *The Non-Linear Analyses of Developmental Processes*. Savelsbergh, G.J.P.; van der Maas, H. & van Geert, P.C.L. eds. Royal Dutch Academy of Arts and Sciences. Amsterdam.
- Price, P.J., C.W. Wightman, M. Ostendorf, & J. Bear. 1990. The use of relative duration in syntactic disambiguation. In *ICSLP-1990*, 13-16.
- Rhapsodie 2010. Corpus prosodique de référence en français parlé.
<http://rhapsodie.risc.cnrs.fr/en/archives.html>
- Roelofs, A. 2000. WEAVER++ and other computational models of lemma retrieval and word- form encoding. *Aspects of language production*. Wheeldon, Linda ed. Psychology Press Hove, UK.
- Saltzman, E. & D. Byrd. 2000. Task-dynamics of gestural timing: Phase windows and multifrequency rhythms. *Human Movement Science*. 19;4. 499-526.
- Saltzman, E.; H. Nam, J. Krivokapic & L.M. Goldstein. 2008. A task-dynamic toolkit for modeling the effects of prosodic structure on articulation. *Proceedings of Speech Prosody 2008*. Campinas, Brazil.
- Scott, D.R. 1982. Duration as a cue to the perception of a phrase boundary. *Journal of the Acoustical Society of America*. 71; 4. 996-1007.
- Shattuck-Hufnagel, S., L. Dille, N. Veilleux, A. Brugos & R. Speer. 2004. F0 peaks and valleys aligned with non-prominent syllables can influence perceived prominence in adjacent syllables. In *Proceedings of Speech Prosody 2004*, 705-708, Nara, Japan
- Shattuck-Hufnagel, S. & A.E. Turk. 1998. The domain of phrase-final lengthening in English. *Proceedings of the 16th International Congress on Acoustics*. Seattle, USA.
- Silverman, K., M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert & J. Hirschberg. 1992. TOBI: A standard for labeling English prosody. *Proceedings of the Second International Conference on Spoken Language Processing*. Banff, Canada.
- Smith, A. 1992. The control of orofacial movements in speech. *Critical Reviews in Oral Biology & Medicine*. 3;3. 233-267.
- SOFA Statistics version 1.2.2. Auckland, New Zealand: Paton-Simpson & Associates Ltd

- Taga, G. 1995. A model of the neuro-musculo-skeletal system for human locomotion. *Biological Cybernetics*. 73;2. 97-111.
- Tajima, K. 1998. Speech rhythm in English and Japanese: Experiments in speech cycling.
- Tajima, K. & R. Port. 2003. Speech rhythm in English and Japanese. *Phonetic interpretation: Papers in laboratory phonology VI*. Local, J.; Ogden, R. & Temple, R. eds. Cambridge University Press, Cambridge, UK. 317–334.
- Taylor, B. E., & K. Lukowiak. 2000. The respiratory central pattern generator of *Lymnaea*: a model, measured and malleable. *Respiration physiology*, 122(2), 197-207.
- Turk, A. & S. Shattuck-Hufnagel. 2007. Multiple targets of phrase-final lengthening in American English. *Journal of Phonetics*. 35; 4. 445-472.
- Vousden, J.I., G.D. Brown & T.A. Harley. 2000. Serial control of phonology in speech production: A hierarchical model. *Cognitive Psychology*. 41;2. 101-175.
- Wells, J.C. 2006. *English Intonation: An Introduction*. Cambridge University Press, Cambridge.
- Wildgruber, D, H. Ackermann, U. Klose, B. Kardatzki & W. Grodd. 1996. Functional lateralization of speech production at primary motor cortex: a fMRI study. *NeuroReport*. 7:15-17. 2791-5.
- Wilson, D.M. 1961. The central nervous control of flight in a locust. *J. of Experimental Bio*. 38. 471-490.
- Zawaydeh, B.A., K. Tajima & M. Kitahara. 2002. Discovering Arabic rhythm through a speech-cycling task. *Amsterdam Studies in the Theory and History of Linguistic Science: Series 4*. Benjamins Publishing Company. Amsterdam.